
DMSVIVA
2018
Proceedings of the 24th
International Distributed
Multimedia Systems
Conference on
Visualization and
Visual Languages
San Francisco Bay
June 29-30, 2018

PROCEEDINGS

DMSVIVA2018

The 24th International DMS Conference on Visualization and Visual Languages

Sponsored by

KSI Research Inc. and Knowledge Systems Institute, USA

Technical Program

June 29 to 30, 2018

Hotel Pullman, Redwood City, California, USA

Organized by

KSI Research Inc. and Knowledge Systems Institute, USA

ISBN: 1-891706-45-4

ISSN: 2326-3261 (print)

2326-3318 (online)

DOI: 10.18293/DMSVIVA2018

Additional copies can be ordered from:

KSI Research Inc.

156 Park Square Lane

Pittsburgh, PA 15238 USA

Tel: +1-412-606-5022

Fax: +1-847-679-3166

Email: dms@ksiresearch.org

Web: http://www.ksi.edu/ seke/dmsvlss18.html

Proceedings preparation, editing and printing are sponsored by KSI Research Inc. and Knowledge Systems Institute, USA.

Printed by KSI Research Inc. and Knowledge Systems Institute, USA.

FOREWORD

On behalf of the Program Committee of the *24th International DMS Conference on Visualization and Visual Languages (DMSVIVA2018)*, we would like to welcome you to San Francisco, USA. This year's main theme was Soft Computing in Visualization and Visual Languages. The conference aimed at bringing together experts in visualization, visual languages and distributed multimedia computing and providing a forum for productive discussions about these topics.

It is our pleasure to announce that by the deadline of 23 March 2018, the conference and the special sessions of Big Data Analytics and Smart Cities received 25 submissions. All the papers were rigorously reviewed by three to five members of the international Program Committee. Based on the review results, 11 papers have been accepted as regular papers with an acceptance rate of 44% and 5 accepted as short papers with an acceptance rate of 20%. We would like to thank all the authors for their contributions.

This year, we had a rich collection of activities in the technical program, including one plenary talk by Professor Gem Stapleton of University of Brighton, UK, an invited talk by Professor Zhi Li of Guangxi Normal University, China, and four technical sessions:

- Supporting People: Education and Perception
- Smart Cities and Systems
- Data and Systems
- Graphs, Logic and Validation

The invited talks and technical sessions covered a wide range of topics in multimedia, education, data sciences, and visual languages.

We are very grateful to the two invited speakers for their inspiring talks, and to the members of the Program Committee for their publicity effort and timely reviews of the submitted papers. We are particularly grateful to Andrew Fish, Lidia Stanganelli, and Max North for their additional and much heavier load of paper reviews. Technical Session Chairs Jennifer Leopold, Zhi Li, Walter Balzano are also very much appreciated for their organizational assistance during the conference.

Finally, we would like to thank the Steering Committee Chair Professor Shi-Kuo Chang for his guidance and leadership throughout organization of this conference. The assistance of the staff at KSI Research and Knowledge Systems Institute is also greatly appreciated. Special thanks go to Wilson Chen for his effective and efficient assistance in working with the paper submission and review system, which made the review process smooth and timely.

Kang Zhang, The University of Texas at Dallas, USA
Gem Stapleton, University of Brighton, UK
Program Co-Chairs

DMSVIVA2018

The 24th International DMS Conference on Visualization and Visual Languages

June 29 to 30, 2018

Hotel Pullman, Redwood City, California, USA

Conference Organization

DMSVIVA2018 Conference Chair and Co-Chairs

Jennifer Leopold, Missouri University of Science & Technology, USA; Conference Chair
F. Colace, University of Salerno, Italy; Conference Co-Chair
Weibin Liu, Beijing JiaoTung Univ., China; Conference Co-Chair
Chaman Sabharwal, Missouri University of Science & Technology, USA; Conference Co-Chair

DMSVIVA2018 Steering Committee Chair

Shi-Kuo Chang, University of Pittsburgh, USA; Steering Committee Chair

DMSVIVA2018 Steering Committee

Paolo Nesi, University of Florence, Italy; Steering Committee Member
Kia Ng, University of Leeds, UK; Steering Committee Member

DMSVIVA2018 Program Chair and Co-Chair

Kang Zhang, University of Texas at Dallas, USA; Program Chair
Gem Stapleton, University of Brighton, UK; Program Co-Chair

DMSVIVA2018 Program Committee

Subcommittee on Visualization and Visual Languages
Bilal Alsallakh, Vienna University of Technology, Austria
Danilo Avola, University of Rome, Italy
Paolo Bottoni, Universita Sapienza, Italy
Paolo Buono, University of Bari, Italy
Peter Chapman, University of Brighton, UK
Kendra Cooper, University of Texas at Dallas, USA
Gennaro Costagliola, University of Salerno, Italy

Sergiu Dascalu, University of Nevada, USA
Aidan Delaney, University of Brighton, UK
Vincenzo Deufemia, University of Salerno, Italy
Filomena Ferrucci, University of Salerno, Italy
Andrew Fish, University of Brighton, UK
Manuel J. Fonseca, University of Lisbon, Portugal
Jun Kong, North Dokota State University, USA
Robert Laurini, University of Lyon, France
Jennifer Leopold, Missouri University of Science & Technology, USA
Luana Micallef, Helsinki Institute for Information Technology, Finland
Joseph J. Pfeiffer, Jr., New Mexico State University, USA
Peter Rodgers, University of Kent, UK
Giuseppe Santucci, University Di Roma, Italy
Gem Stapleton, University of Brighton, UK
Giuliana Vitiello, University of Salerno, Italy

Subcommittee on Sentient and Distributed Multimedia Systems
Arvind K. Bansal, Kent State University, USA
Andrew Blake, University of Brighton, UK
Loredana Caruccio, University of Salerno, Italy
William Cheng-Chung Chu, Tunghai University, Taiwan
Gennaro Costagliola, Univ of Salerno, Italy
Tiansi Dong, Bonn-Aachen International Center for Information Technology, Germany
Martin Erwig, Oregon State University, USA
Daniela Fogli, Universita degli Studi di Brescia, Italy
Kaori Fujinami, Tokyo University of Agriculture and Technology, Japan
David Fuschi, Brunel University, UK
Angela Guercio, Kent State University, USA
Carlos A. Iglesias, Intelligent Systems Group, Spain
Yau-Hwang Kuo, National Cheng Kung University, Taiwan
Fuhua Lin, Athabasca University, Canada
Alan Liu, National Chung Cheng Univeristy, Taiwan
Max North, Southern Polytechnic State University, USA
Antonio Piccinno, Univ. of Bari, Italy
Giuseppe Polese, University of Salerno, Italy
Genny Tortora, University of Salerno, Italy
Atsuo Yoshitaka, JAIST, Japan
Ing Tomas Zeman, Czech Technical University, Czech Republic
Kang Zhang, University of Texas at Dallas, USA

Subcommittee on Distance Education Technologies
Maiga Chang, Athabasca University, Canada
Mauro Coccoli, University of Genova, Italy
Rita Francese, University of Salerno, Italy
Angelo Gargantini, University of Bergamo, Italy
Hong Lin, University of Houston-Downtown, USA
Paolo Maresca, University Federico II, Napoli, Italy

Andrea Molinari, University of Trento, Trento, Italy
Elvinia Riccobene, University of Milano, Italy
Michele Risi, University of Salerno, Italy
Teresa Roselli, University of Bari, Italy
Lidia Stanganelli, Freelance Software Engineer, Italy

Publicity Chair and Co-Chair

Eloe Nathan, Northwest Missouri State University, USA; Publicity Chair
Lidia Stanganelli, Freelance Software Engineer, Italy; Publicity Co-Chair

Plenary Talk

Reasoning with Diagrams: Observation, Inference and Overspecificity

Gem Stapleton
University of Brighton
Brighton, United Kingdom

Abstract

The ability of diagrams to convey information effectively comes, in part, from their ability to make facts explicit that would otherwise need to be inferred. This type of advantage has often been referred to as a free ride and was deemed to occur only when a diagram was obtained by translating a symbolic representation of information. Recent work generalised free rides, introducing the idea of an observational advantage, where the existence of such a translation is not required. In this talk, I will provide an overview of the theory of observation. Using observability, a formal characterisation of observational advantages can be explored. The talk will proceed to demonstrate the theory of observation and observational advantages by applying the concepts to set theory and Euler diagrams without existential import. It has been shown that Euler diagrams without existential import have significant observational advantages over set theory: they are observationally complete. The talk will then explore to what extent Euler diagrams with existential import are observationally complete with respect to set-theoretic sentences. In particular, it will be shown that existential import significantly limits the cases when observational completeness arises, due to the potential for overspecificity. These two results formally support Larkin and Simon's claim that “a diagram is (sometimes) worth ten thousand words”. (Note: This is joint research with Atsushi Shimojima and Mateja Jamnik).

About the Speaker

Dr Gem Stapleton is a Reader in Computer Science at the University of Brighton. She has over 120 publications and received five Best Paper Awards at international conferences. A major focus of her research has been on developing diagrammatic logics that support accessible reasoning. As part of this effort, she has devised layout algorithms for diagrams that incorporate geometric and topological constraints which are important for usability. As well as having a strong mathematical element, her research also encompasses empirical HCI to ensure the usability of her theoretical work. She has been PI and Co-I on major grants from the UK’s EPSRC and the Leverhulme Trust. Gem has also organised many international conferences, was Chair of the Diagrams Steering Committee (until 2016) and is a member of the VL/HCC Steering Committee.

Invited Talk

PURE: Problem-oriented Urban Requirements Engineering for Big Data Analytics

Zhi Li
College of Computer Science and Information Technology
Guangxi Normal University, China

Abstract

Modern cities, especially metropolitans in China are facing challenging problems in terms of deterioration of environment quality, heavy traffic congestion, over-consumption of energy, and other socio-technical issues. There are multiple dimensions to this complex and taxing problem, which requires real-time analytics based on large volumes of multiple-source data sets for real-time predictions and decision-making. The research work on "urban computing" have provided solutions based on data mining techniques, which can help meet some of these challenges in urban planning and management. However, the software engineering of such analytics systems in a real world setting remains an under-explored research area. From a Requirements Engineering (RE) perspective, understanding the stakeholders and their problems is of primary importance, in particular identifying their requirements value chains and further defining a data value chain for big data is more promising than trying to obtain the full data set. In this talk, I will present our PURE process framework which spans three sub-processes, namely the Requirements Engineering (RE), Problem-Orientation (PO) and Urban Computing processes.

[1] Zheng, Y., L. Capra, O. Wolfson and H. Yang (2014). Urban Computing: Concepts, Methodologies, and Applications. ACM Trans. Intell. Syst. Technol. 5(3): 1-55.

[2] Fricker, S (2010). Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems, in Requirements Engineering: Foundation for Software Quality: 16th International Working Conference, REFSQ 2010, Essen, Germany, June 30-July 2, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 60-66.

[3] H. G. Miller and P. Mork (2013). From Data to Decisions: A Value Chain for Big Data. in IT Professional, vol. 15, no. 1, pp. 57-59, Jan.-Feb. 2013.doi: 10.1109/MITP.2013.11

About the Speaker

Dr. Zhi Li is a Full Professor at Guangxi Normal University (Vice-Dean of College of Computer Science and Information Technology), China. He is a Distinguished Member of China Computer Federation (CCF), and a member of CCF's Technical Council on Software Engineering (TCSE) and Task Force on Formal Methods. Prof. Li's research interests are Problem-oriented Requirements Engineering (PURE) for big data analytics, modelling and verification of cyber-physical systems, software testing and human-computer interaction. His research has been sponsored by grants from the National Natural Science Foundation of China, and the Provincial Natural Science Foundation of Guangxi. He has published over 20 research papers and obtained 8 software copyrights in China. Before joining GXNU, he was a post-doctoral researcher and honorary research fellow at Keele University (Software Engineering Research Group, 2007-2010). Prof. Li completed his PhD in computer science at the Open University (Software Engineering and Design, 2003-2007), after obtaining an MSc from the Department of Computer Science at the University of York.

Table of Contents

Plenary Talk

Reasoning with Diagrams: Observation, Inference and Overspecificity (Plenary) 1
Gem Stapleton

Session I Supporting People: Education and Perception

A Sentiment Analysis Approach for supporting Blended Learning Process 8
Francesco Colace, Fabio Clarizia, Marco Lombardi and Francesco Pascale

Implications of learning environments on the Information Systems of educational institutions 15
Andrea Molinari and Paolo Maresca

Where do People Look while Identifying Colors in Images 23
Soraia M. Alarcão, Ruben Pavão and Manuel J. Fonseca

Session II Smart Cities and Systems

A multi-level approach for forecasting critical events in Smart Cities (S) 31
Francesco Colace, Marco Lombardi, Francesco Pascale and Domenico Santaniello

Supporting Living Lab with Life Cycle and Tools for Smart City Environments (S) 36
Paolo Nesi and Mchela Paolucci

Smart City Control Room Dashboards Exploiting Big Data Infrastructure (S) 44
Paolo Nesi, Pierfrancesco Bellini, Mino Marazzini, Nicola Mitolo, Michela Paolucci and Daniele Cenni

RADS: a smart Road Anomalies Detection System using Vehicle-2-Vehicle network and cluster features (S) 51
Walter Balzano and Fabio Vitale

Session III Data and Systems

Enriching IAPS and GAPED Image Datasets with Unrestrained Emotional Data 57
Soraia M. Alarcão and Manuel J. Fonseca

A Mobile Dietary and Emotional Diary System for Eating Disorder Care on the Smart Phone 65
Shikuo Chang, Jung Kim and Hanzhong Zheng

Event-Based Data Input, Modeling and Analysis for Meditation Tracking using TDR System 71
Shikuo Chang, Cuiling Chen, Wei Guo and Nannan Wen

Generating Synthetic Memory References via Hierarchical Markov Models Effectively and Efficiently 83
Alfredo Cuzzocrea

Session IV Graphs, Logic and Validation

Spider Diagrams with Absence 91
Gem Stapleton, Lopamudra Choudhury and Mihir Chakraborty

An Edge-based Graph Grammar Formalism and its Support System 101
Xiaoqin Zeng, Yufeng Liu, Zhan Shi, Yingfeng Wang, Yang Zou, Jun Kong and Kang Zhang

BugHint: A Visual Debugger Based on Graph Mining 109
Jennifer Leopold, Nathan Eloe and Patrick Taylor

A Logic Range-free Algorithm for Localization in Wireless Sensor Networks 119
Walter Balzano and Silvia Stranieri

On the Problem-Oriented Validation of Cyber-Physical Systems Using System-Level Test Sequences (S) 125
Changlan Fu, Xiao Zhang, Zhi Li, Ziyan Zhao, Chao Wang and Yuekun Yu

Note: (S) indicates a short paper

Reasoning with Diagrams: Observation, Inference and Overspecificity

Gem Stapleton
Centre for Secure, Intelligent and Usable Systems,
University of Brighton, UK
g.e.stapleton@brighton.ac.uk

Abstract

The ability of diagrams to convey information effectively comes, in part, from their ability to make facts explicit that would otherwise need to be inferred. This type of advantage has often been referred to as a free ride and was deemed to occur only when a diagram was obtained by translating a symbolic representation of information. Recent work generalised free rides, introducing the idea of an observational advantage, where the existence of such a translation is not required. In this paper, I will provide an overview of the theory of observation. It has been shown that Euler diagrams without existential import have significant observational advantages over set theory: they are observationally complete. I will then explore to what extent Euler diagrams with existential import are observationally complete with respect to set-theoretic sentences. In particular, has been shown that existential import significantly limits the cases when observational completeness arises, due to the potential for overspecificity. These two results formally support Larkin and Simon's claim that "a diagram is (sometimes) worth ten thousand words". The work in this invited paper is derived from previously published results as cited in the text.

1. Introduction

The choice of notation in which to represent information is an important consideration if the desire is for effective communication. But, even when a choice has been made, one must still select from the semantically equivalent representations of the information to be conveyed. Understanding the impact of such choices from the perspective of human cognition is important. This paper presents an overview of selected state-of-the-art work on these choices from a theoretical perspective, summarising results previously published in [12, 13].

DOI reference number: 10.18293/DMSVIVA2018-026.

There are many ways in which visual representations can be manipulated to impact their effectiveness. For instance, graphical features, such as colour or size, can be manipulated to enhance or diminish the effectiveness of a representation [2]. The particular focus of this paper is the recent work on *observational advantages* [12], which generalises prior work on *free rides* [9]. If we can understand when and how one representation of information has observational advantages over another then it allows us to make an informed choice of representation: we should choose a representation with many observational advantages.

An example can be seen in Figure 1. Here, five textual statements convey information about countries visited by people. They are translated into an Euler diagram which represents exactly the same information. For instance, the first textual statement is visualized by the curve labelled Germany being drawn inside the curve labelled Italy. The fifth statement is shown by the fact that the curves labelled Qatar and Sudan do not overlap. From the diagram, one can simply read off that everyone who visited Germany visited Qatar but this fact needs to be *inferred* from the text. As such, 'everyone who visited Germany visited Qatar' is a free ride from the diagram given the text.

Euler diagrams are not the only notation in which free rides arise. Figure 2 shows the same textual statements but this time translated into a semantically equivalent linear diagram. Here, lines are used to represent sets and their positions relative to each other along the x-axis provides information about subset and disjointness relationships. So, the information that no one visited both Qatar and Sudan is shown by the fact that the two corresponding lines do not overlap. This diagram has exactly the same free rides, given the associated text, as the Euler diagram in Figure 1. For instance, since the lines for Germany and Sudan do not overlap, we can read off that no one visited both Germany and Sudan, which again must be inferred from the text.

Free rides, in general, are defined to arise only when one representation of information is derived by systematically translating another, given, representation of information (see [12] for details). By contrast, observational advan-

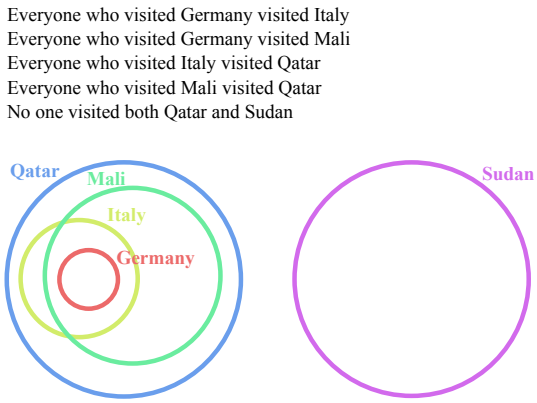


Figure 1. Illustrating free rides in Euler diagrams.

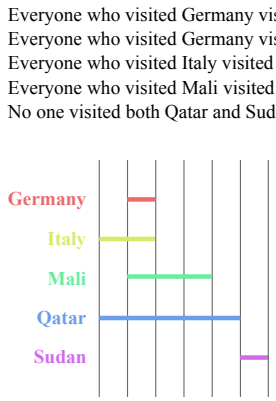


Figure 2. Illustrating free rides in linear diagrams.

tages, of which free rides are examples, only require that the two representations are semantically equivalent, removing the need for a translation. Both of them capture the idea that information which can simply be read off from one representation but must be inferred from another can be considered a (potential) advantage of the former over the latter.

In section 2 we discuss, in more detail, the role of meaning-carriers and observation, where we look at observational advantages. Section 3 demonstrates, by example, how Euler diagrams without existential import (like the examples we have seen so far) are what is known to be *observationally complete* with respect to set-theoretic statements. When the existential import assumption is introduced, observational completeness is no longer guaranteed to hold, demonstrated in section 4. We then conclude in section 5, where we highlight the need for future work, specifically focusing on cognition and usability. The main results in this invited paper are derived from previously published results, primarily [11, 13].

2. Observation

A formal account of the idea of an observational advantage and, therefore, of free rides, requires a formal definition of the syntax and semantics of the notations being considered. Of course, we do not have that for natural language in general, although the statements given in Figures 1 and 2 are clearly in a controlled form. To ease our exposition, therefore, we consider versions of these statements in set-theoretic notation: $Germany \subseteq Italy$, $Germany \subseteq Mali$, $Italy \subseteq Qatar$, $Mali \subseteq Qatar$, and $Qatar \cap Sudan = \emptyset$. Each of these five statements[1] express the desired information. What is important here is that each statement has a single *meaning-carrying relationship*.

This idea of a meaning-carrying relationship is at the heart of how free rides and observational advantages are defined. So what is a meaning-carrying relationship? Well, it is taken to be a relation on the syntax of a representation that evaluates to either 'true' or 'false' when the syntax is given meaning. Therefore, a meaning-carrier is similar to a 'representing fact' in Shin's work [10]. Why, then, do each of the five set-theoretic statements have a single meaning-carrying relationship? Consider $Germany \subseteq Italy$, which is either true or false, depending on the sets represented by $Germany$ and $Italy$. The meaning-carrying relationship is that $Germany$ is written to the left of $\subseteq$ and $Italy$ to the right. Similarly, the fifth statement's meaning carrier is that $Qatar \cap Sudan$ is written to the left of $=$ and $\emptyset$ is on the right.

What, then, are the meaning-carrying relationships in the Euler and linear diagrams? Each diagram is a single statement but they have *many* meaning-carrying relationships. The Euler diagram in Figure 1 uses curve containment to express the four subset statements. For the two curves arising from $Germany \subseteq Italy$, we see that the one is inside the other and the assertion made by the diagram is true whenever the set represented by $Germany$ is a subset of that represented by $Italy$, otherwise it is false. So, one example of a meaning-carrying relationship is the containment of one curve by another. Likewise, the Euler diagram uses non-overlapping curves to express the disjointness of sets.

In addition to the five meaning-carrying relationships corresponding to the five textual statements, we have further meaning-carrying relationships, such as the non-overlapping of the $Mali$ and $Sudan$ curves which expresses that no one visited both $Mali$ and $Sudan$. Thus, from the

[1] A statement is a syntactic entity (in any representation system) that represents some information. For example, a set-theoretic sentence is a single statement, and so is an Euler diagram.

Euler diagram we can *observe* that $Mali \cap Sudan = \emptyset$ but we must infer this from the set-theoretic statements. In the linear diagram, Figure 2, the statements that can be observed are the same as those that can be observed from the Euler diagram, even though their syntax is different and their meaning-carriers are therefore identified in a different way.

The presence of multiple meaning-carrying relationships suggests that, as compared to representation systems with single-meaning carrying relationship, facts can sometimes be observed to be true rather than inferred to be true. Representations of information that allow statements to be observed as true, without the need for inference, can be considered advantageous. Here we must be clear, though, that this difference between single and multiple meaning carriers should not suggest a dichotomy of sentential and diagrammatic notations. Merely, the examples we have presented contrast symbolic and diagrammatic representations where the former has single meaning-carriers and the latter has multiple meaning-carriers.

So, we have seen that meaning-carriers can lead to information being observed as true from representations of information. The concept of observation has been considered in proof systems, where it was formalized as an inference rule [1, 14]. To give the idea, an inference rule based on observation allows one to identify pieces of information expressed in a statement and re-express them in another statement. As we have already seen, many sentences support precisely one meaning-carrying relationship. In such cases, if we were to define and apply an observation inference rule, it would merely restate the information in an identical, or semantically equivalent sentence. By contrast, in systems where single statements have many meaning-carriers, such as some diagrammatic representations, using meaning-carriers to apply an observation inference rule can yield many different statements. This suggests that the role of observation is important when considering inference problems.

One example where observation has been used is Barwise and Etchemendy's *Hyperproof* system [1]. Moreover, Swoboda and Allwein [14] included both an observation rule and other inference rules in their work. They called for the distinctive treatment of the observation rule, which consists of visual perception and the restatement of the information thus obtained. This draws on Dretske's classification of various cases that are commonly described as "somebody's seeing that something is the case" [5].

It is necessary to define meaning-carriers and the notion of observation in the context of the syntax and semantics of notations; what it means to be a meaning-carrier is clearly determined by the syntax and semantics. However, if we assume such definitions are given, we can proceed define what it means to be an observational advantage. Firstly, we require that if we observe a statement, σ_o, from another statement, σ then the following properties must hold:

1. some of the meaning-carrying relationships holding in σ also hold in σ_o, and

2. σ_o supports just enough relationships to express the meanings carried by the selected relationships in σ and nothing stronger [12].

These properties ensure that σ_o is semantically entailed by σ.

Suppose now that we have the more general case of a set of statements, Σ, from which we wish to observe information: the *only* meaning-carrying relationships in Σ are derived from the statements in Σ. So, the only statements observable from Σ must be observable from one of the statements in Σ. This leads us to be able to define the notion of observation from a set of statements.

Definition 1 *Let Σ be a finite set of statements and σ_o be a single statement. Then σ_o is **observable** from Σ iff σ_o is observable from some statement, σ, in Σ. The set of statements that are observable from Σ is denoted $\mathcal{O}(\Sigma)$ [12].*

Now we have understood what it means to be observable, we can define what it means to be an observational advantage. Intuitively, an observational advantage occurs when we have two semantically equivalent representations of information, say two sets of statements, Σ and $\hat{\Sigma}$. If we can observe a statement from $\hat{\Sigma}$ but not from Σ then it is an advantage of $\hat{\Sigma}$ over Σ. This is captured by definition 2.

Definition 2 *Let Σ and $\hat{\Sigma}$ be finite, semantically equivalent sets of statements. Let σ be a statement. If*

1. *σ is not observable from Σ, and*

2. *σ is observable from $\hat{\Sigma}$*

*then σ is an **observational advantage** of $\hat{\Sigma}$ given Σ. The set of all observational advantages of $\hat{\Sigma}$ given Σ is denoted $\mathcal{OA}(\hat{\Sigma}, \Sigma)$ [12].*

To finish this section, we consider two extreme cases where observational advantages can arise, or even fail to do so. Firstly, suppose given a set of statements, Σ, there is some information, captured by a set, $\Sigma_{\models}$, of statements, whose truth we want to establish. If we can simply read-off (observe) *all* of the respective statements in $\Sigma_{\models}$ then Σ can be considered *observationally complete*:

Definition 3 *Let Σ and $\Sigma_{\models}$ be finite sets of statements. Then Σ is **observationally complete** with respect to $\Sigma_{\models}$ if*

$$\Sigma_{\models} \subseteq \mathcal{O}(\Sigma) \text{ [12].}$$

Lastly, we have the other extreme case: if we cannot simply read-off (observe) *any* of the respective statements, then Σ can be considered *observationally devoid*:

Definition 4 *Let Σ and $\Sigma_{\models}$ be finite sets of statements. Then Σ is **observationally devoid** with respect to $\Sigma_{\models}$ if*

$$\Sigma_{\models} \cap \mathcal{O}(\Sigma) = \emptyset \text{ [12].}$$

These two extreme cases allow us to formally establish when one representation has numerous advantages over another. This occurs when, say, Σ is observationally complete, yet $\hat{\Sigma}$ is observationally devoid with respect to a given $\Sigma_{\models}$.

3. Observational Advantages of Euler Diagrams without Existential Import

Euler diagrams have substantial observational advantages over set-theoretic *statements*. For our purposes, we consider set-theoretic *expressions* of the form $s_1 \cap s_2$, $s_1 \cup s_2$, $s_1 \backslash s_2$ and $\overline{s_1}$ as well as the special symbols U (the universal set) and $\emptyset$ (the empty set). We can then form set-theoretic statements using these expressions. The following are the set-theoretic statements that we consider: $s_1 \subseteq s_2$ and $s_1 = s_2$. This allows a rich variety of information about sets to be expressed. What is important here is that given any finite collection of set-theoretic statements, there exists a semantically equivalent Euler diagram (without existential import).

As an example, consider the following set-theoretic statements:

1. no one visited both Sudan and Vietnam:

$$Sudan \cap Vietnam = \emptyset$$

2. no one visited both Denmark and Vietnam:

$$Denmark \cap Vietnam = \emptyset$$

3. everyone who visited Denmark visited France:

$$Denmark \subseteq France$$

4. everyone who visited Libya visited Sudan:

$$Libya \subseteq Sudan$$

5. no one visited both France and Libya:

$$France \cap Libya = \emptyset.$$

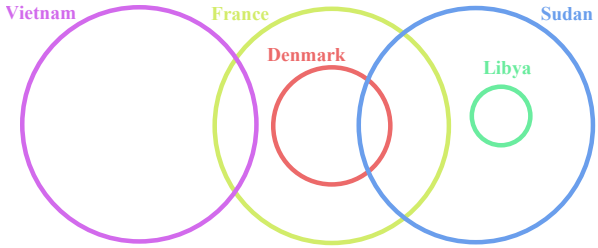


Figure 3. Observational completeness.

A semantically equivalent Euler diagram (without existential import) can be seen in Figure 3.

We have already seen some examples of set-theoretic statements that are observable from Euler diagrams. In this particular case, one observable statement is

$$Vietnam \cap Libya = \emptyset$$

as well as the equivalent statement $Libya \cap Vietnam = \emptyset$. More complex set-theoretic statements are also observable, such as

$$(Denmark \cup Libya) \cap Vietnam = \emptyset.$$

Why can this more complex statement be observed? Well, here we must consider the regions formed by the curves. The region, comprising three *zones*, formed by the interiors of the curves for Denmark and Libya is disjoint from (shares no points with) the region inside the Vietnam curve. Thus, the *disjointness* of two regions is a meaning-carrier in Euler diagrams. The presence of this meaning-carrier shows that we can observe $(Denmark \cup Libya) \cap Vietnam = \emptyset$ from the diagram.

Consider, then, the types of set-theoretic expressions to which we have access, in the context of our example:

1. Each 'basic' set-expression (i.e. one that does not involve intersection, union, difference or complement) corresponds to a region inside a curve. In our example, there are five basic set-expressions: $Denmark$, $France$, $Libya$, $Sudan$, and $Vietnam$. We say, informally, that these basic set-expressions *correspond* to regions in the diagram.

2. Given any two set–expressions, s_1 and s_2, that correspond to regions in the diagram, the set-expressions $s_1 \cap s_2$, $s_1 \cup s_2$, $s_1 \backslash s_2$, and $\overline{s_1}$ also correspond to regions in the diagram (noting that 'empty' regions contain no points). For example, $Denmark \cup France$, $Denmark \cap France$, $Sudan \backslash France$ and $\overline{Libya}$ all correspond to regions.

Using this insight, it is easy to see that any set-expression that can be formed from the basic ones corresponds to some

region in the diagram. Therefore, from the diagram, given any set-theoretic statement of the form $s_1 \subseteq s_2$, there are corresponding regions for s_1 and s_2. We can the *observe* $s_1 \subseteq s_2$ from the diagram precisely when the region, r_1 for s_1 is a subset (contained by) the region for s_2. Likewise, if $r_1 = r_2$ then we can observe $s_1 = s_2$. This insight leads us to the following theorem, which is derived from one of the main results in [12]:

Theorem 1 *Let $\mathcal{S} = \{s_1, ..., s_n\}$ be a finite set of set-theoretic statements. Then there exists an Euler diagram, d, where:*

1. *d semantically equivalent to S, and*
2. *given any set-theoretic statement, s, that is formed from set-expressions whose basic sets are all used in S and which is semantically entailed by S can be observed from d.*

In other words, d is *observationally complete* with respect to the set, $S_{\models}$, of all statements that we can infer from S. But, on the other hand, what can be observed from S? Well, since the set-theoretic statements in S only have one meaning carrier, S is observationally devoid, given $S_{\models} \backslash S$. This means that the Euler diagram has maximal observational advantage over S and these result tells us that Euler diagrams are powerful representations of information compared to set-theoretic statements.

4. Observational Advantages of Euler Diagrams with Existential Import

Euler diagrams that do not enforce existential import, which have been the focus up to this point, allow regions in the diagram to represent empty sets. For instance, in Figure 3, there is no information that anyone at all visited Vietnam (or the other countries). There are occasions, of course, when we want to enforce the non-emptiness of sets (or, more generally, provide cardinality information, but that is beyond the scope of this discussion). Euler diagrams can be extended in various was to achieve this.

For instance, Peirce denotes the non-emptiness of a set with $\otimes$-sequences [8], also used by Shin [10] and further developed by Choudhury and Chakraborty [4]. By contrast, Euler diagrams with existential import [3] do not require additional syntax to assert the non-emptiness of a set: all zones are taken to represent non-empty sets (so, any region formed by the curves represents a non-empty set) [6]. The extension of the semantics to require that zones represent non-empty sets leads to an obvious question: are Euler diagrams with existential import still observationally complete?

To answer this question, we need to consider a wider variety of set-theoretic statements, even though the use of intersection, union, difference and complement for forming set-expressions is still appropriate and sufficient. This is, obviously, because using just $\subseteq$ and $=$ does not lead to assertions about non-emptiness. Therefore, we also allow the use of $\not\subseteq$ and $\neq$. Using these two additional operators, we can form statements like $Vietnam \neq \emptyset$ and $Denmark \not\subseteq Sudan$ (so implying that $Denmark \backslash Sudan \neq \emptyset$).

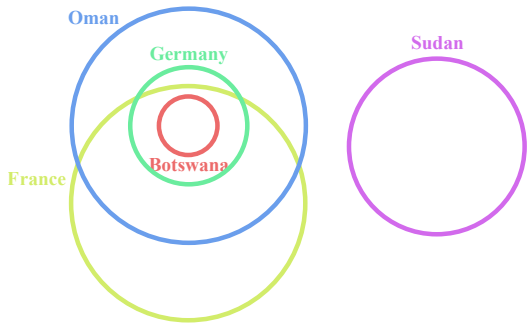


Figure 4. An Euler diagram interpreted under existential import semantics.

Consider, now, the Euler diagram in Figure 4 under the existential import assumption. Can we find a set of set-theoretic sentences that capture the meaning conveyed by this diagram? There are four statements corresponding to the set-theoretic relationships expressible using $\subseteq$ and $=$ (noting that other statements could be written down too, but they would not convey extra subset or equality information):

1. everyone who visited Botswana visited France and Germany:
$$Botwana \subseteq France \cap Germany,$$
2. everyone who Germany visited Oman:
$$Germany \subseteq Oman,$$
3. no one visited both Oman and Sudan:
$$Oman \cap Sudan = \emptyset,$$
and
4. no one visited both France and Sudan:
$$France \cap Sudan = \emptyset.$$

However, these statements alone are not semantically equivalent to the Euler diagram under the existential import assumption, since each zone represents a non-empty set. We need a further seven statements to achieve semantic equivalence (again, different choices of statements exist - there are non-unique sets of set-theoretic statements that are semantically equivalent to the diagram):

1. at least one person visited Oman only:

   $Oman \cap \overline{Botswana} \cap \overline{France} \cap \overline{Germany} \cap \overline{Sudan} \neq \emptyset$,

2. at least one person visited France only:

   $France \cap \overline{Botswana} \cap \overline{Germany} \cap \overline{Oman} \cap \overline{Sudan} \neq \emptyset$,

3. at least one person visited Sudan only:

   $Sudan \cap \overline{Botswana} \cap \overline{France} \cap \overline{Germany} \cap \overline{Oman} \neq \emptyset$,

4. at least one person visited both Germany and Oman, but no other country:

   $Germany \cap Oman \cap \overline{Botswana} \cap \overline{France} \cap \overline{Sudan} \neq \emptyset$,

5. at least one person visited both France and Oman, but no other country:

   $France \cap Oman \cap \overline{Botswana} \cap \overline{Germany} \cap \overline{Sudan} \neq \emptyset$,

6. at least one person visited all of France, Germany and Oman, but no other country:

   $France \cap Germany \cap Oman \cap \overline{Botswana} \cap \overline{Sudan} \neq \emptyset$,

   and

7. at least one person visited all of Botswana, France, Germany and Oman:

   $$Botswana \cap France \cap Germany \cap Oman \neq \emptyset,$$

Notice that the last statement does not involve all five sets. This is sufficient because we know that the set specified by the four-way intersection is disjoint from Sudan (which follows from the earlier statements). We could, instead, have chosen the set-theoretic sentence

$$Botswana \cap France \cap Germany \cap Oman \cap \overline{Sudan} \neq \emptyset.$$

This illustrates how choice of statements can arise, when seeking semantic equivalence.

As with Euler diagrams without existential import, we can observe information from the diagram that needs to be inferred from the set-theoretic statements. For instance, any region represents a non-empty set, so we can observe that $Oman \neq \emptyset$ and $France \cap Oman \neq \emptyset$. We can further observe that $Oman \not\subseteq France$. Many other statements can be observed too. In fact, this diagram is observationally complete with respect to the set of set-theoretic sentences that are given in the two lists above.

What should be evident from this example is that given an arbitrary set of set-theoretic statements there need not exist a semantically equivalent Euler diagram. For instance, the first five statements, focusing on subset and equality, cannot be translated into a semantically equivalent Euler diagram with existential import. If we were to attempt to produce such diagram it would be that in Figure 4, but we know that this expresses more information than those five statements alone.

The problem here arises because of overspecificity. Unfortunately, due to overspecificity, there are numerous sets of set-theoretic sentences where no semantically equivalent Euler diagram with existential import exists. This indicates a problematic situation: diagrams are typically believed to excel as representations of information due to their ability to make facts explicit that would otherwise need to be inferred. But, as a positive, what we can take away from this discussion is that, given a finite set of set-theoretic sentences, if there exists a semantically equivalent Euler diagram with existential import then that diagram is observationally complete [13][2].

5. Conclusion

It has been seen, though a consideration of Euler diagrams under varying semantic conventions, that sometimes they are capable of representing information in an observationally complete way. The incorporation of existential import brings with it increased expressiveness but at a price: overspecificity means that often information cannot be expressed appropriately. The results support Larkin and Simon's claim that "a diagram is (sometimes) worth ten thousand words" [7].

The theoretical characterisation of what it means to be observable, via meaning-carriers, and the subsequent consideration of observational completeness, is driven by a desire to understand what makes diagrams cognitively more effective than symbolic or textual representations. Now that such theory has been developed, it is important to conduct empirical studies to ascertain the relationship between observational advantages and cognitive advantages. Are observational advantages really helpful? That is, do people extract information more effectively from a representation with an observational advantage over another, or does the process of inference lead to more effective task performance? Assuming that observational advantages do bring cognitive benefit, is there a way of modelling the relative cognitive benefit of some observational advantages over others? This latter question is inspired by the fact that some set-theoretic statements are likely to be more readily observable, by people, from a diagram than others.

[2]The conditions under which this happens are non-trivial and so omitted.

Acknowledgement This research was funded by a Leverhulme Trust Research Project Grant (RPG- 2016-082) for the project entitled Accessible Reasoning with Diagrams. I am also grateful to Atsushi Shimojima and Mateja Jamnik with whom I collaborated on the results summarised in this paper and to Andrew Blake for providing the images in Figures 1 and 2.

References

[1] J. Barwise and J. Etchemendy. *Hyperproof*. CSLI Publications, Stanford, CA, USA, 1994.

[2] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[3] S. Chatti and F. Schang. The cube, the square and the problem of existential import. *History and Philosophy of Logic*, 34(2):101–132, 2013.

[4] L. Choudhury and M. K. Chakraborty. On extending Venn diagrams by augmenting names of individuals. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 142–146. Springer-Verlag, March 2004.

[5] F. Dretske. *Seeing and Knowing*. Routledge & Kegan Paul, London, UK, 1969.

[6] E. Hammer and S. J. Shin. Euler's visual logic. *History and Philosophy of Logic*, pages 1–29, 1998.

[7] J. Larkin and H. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11:65–99, 1987.

[8] C. Peirce. *Collected Papers*, volume 4. Harvard University Press, Cambridge, MA, USA, 1933.

[9] A. Shimojima. *Semantic Properties of Diagrams and Their Cognitive Potentials*. CSLI Publications, Stanford, CA, USA, 2015.

[10] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, Cambridge, UK, 1994.

[11] G. Stapleton, J. Howse, J. Taylor, and S. Thompson. What can spider diagrams say? In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams*, volume 2980 of *LNAI*, pages 112–127, Cambridge, UK, 2004. Springer.

[12] G. Stapleton, A. Shimojima, and M. Jamnik. What makes an effective representation of information: A formal account of observational advantages. *Journal of Logic, Language and Information*, 26(2):143177, 2017.

[13] G. Stapleton, A. Shimojima, and M. Jamnik. The observational advantages of Euler diagrams with existential import. In *Diagrams*, LNCS. Springer, 2018.

[14] N. Swoboda and G. Allwein. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. *Journal on Software and System Modeling*, 3(2):136–149, 2004.

A Sentiment Analysis Approach for supporting Blended Learning Process

F.Clarizia, F. Colace, M. Lombardi, F. Pascale
DIIn
University of Salerno
Fisciano (SA), Italy
{fclarizia, fcolace, malombardi, fpascale}@unisa.it

Abstract— E-Learning is one of the most widely used training approaches in recent years. Numerous universities and training institutions adopt this approach to deliver courses or support the students in their training process. In particular, the blended E-Learning is a useful approach for supporting students and better understanding their learning issues. The possibility of using collaborative tools and interacting with other students allows the student to share doubts on certain topics. The teacher often remains outside of this dynamic and does not understand the learning problems that characterize the class. A possible solution, which ensures the privacy of communication between students, is the Sentiment Analysis. The computational study of opinions, feelings and emotions expressed in a text often relates to the identification of agreement or disagreement with statements, contained in comments that convey positive or negative feelings. In this paper, we investigate the adoption of a probabilistic approach based on the Latent Dirichlet Allocation (LDA) as Sentiment Grabber. Through this approach, for a set of documents belonging to a same knowledge domain, a graph, the Mixed Graph of Terms, can be automatically extracted. The paper shows how this graph contains a set of weighted word pairs, which are discriminative for sentiment classification. In this way, the system can detect the feeling of students on some topics and teacher can better tune his/her teaching approach. In fact, the proposed method has been tested in real cases with effective and satisfactory results.

Keywords-Sentiment Analysis; e-Learning; Collaborative Learning Approach; Blended Approach.

I. INTRODUCTION

Our society is living a transformation, maybe the most important of the latest years, which, through the strong diffusion of the new information technologies, is radically modifying the nature of the relationships among countries, markets, people and cultures. This technological revolution has clearly facilitated the process of globalization, Internet well represents the concept of global village, and the information exchange [28][29].

Information can be considered as an economic good whose value is tightly linked to the amount of knowledge that can give to its users. Gaining new knowledge, competences or skills has determined the need for a continuous update by the actors of the supply chain of the new economy. In fact, in this context, a fundamental service is the life-long learning, or permanent training, which continues all along life and aims at promoting people's fulfilment both at personal and social level. In the learning society, keeping continuously up-to-date is the essential condition to live in it and follow the changes of our times. In this scenario, the information technologies, the languages, the business management are among the sectors that depend more and more on the on-line training services [36] [37] [38].

For about twenty years, the 'E-Learning' phenomenon has largely spread itself in the distance-learning panorama. This reality reverses the paradigm of the old distance education experiences representing the evolution through the technological platforms. These use the Internet and/or the web and the user's monitoring and tracking procedures perfectly integrating the pedagogical and technological aspect for a dynamic learning [30] [33] [34].

Employing the new tools offered by the Web 2.0, the E-Learning gives innovative services that make possible the realization of typical aspects of the 'collaborative learning' and allow the users to have an efficient on-line 'conversation'. The students can leave the old role of users who received information with a top-down approach, to assume a new position of talkers, of people who interact among them creating and exchanging culture [31][32].

Recent studies showed that emotions can affect the E-Learning experience. [1] What are emotions? A general definition for emotions could be the following: emotions are complex psychophysical processes that evoke positive or negative psychological responses (or both) and physical expressions, often involuntary. Emotions are often related to feelings, perceptions or beliefs about elements, objects or relations between them, in reality or in the imagination. They typically arise spontaneously, rather than through conscious effort. An emotion (reaction or state) is often differentiated from a feeling (sensation or impression), although the word "feeling" is used as a synonym for "emotion" in some contexts. Obviously, the topic of emotions goes far beyond this simple definition and it is especially hard to detect in an e-learning environment [35]. In a face-to-face class, instructors can detect facial expressions of students but, in an online environment, students need to establish an online presence and the instructors need to be able to notice this [2]. In this scenario, a promising approach is the sentiment analysis: the computational study of opinions, sentiments and emotions expressed in a text [3]. Its main aim is the identification of agreement or disagreement statements to capture positive or negative feelings in comments or reviews. Many scholars are investigating the adoption of Sentiment Analysis in E-Learning field [6]. An introduction to an opinion mining framework that can be manipulated to work in an e-learning system was presented by [7]. A promising

DOI reference number: 10.18293/DMSVIVA2018-001

approach uses Conditional Random Fields for identifying and extracting the opinions; it considers the negative sentences and degree adverbs in sentiment processing [8]. The experiment has proved that it is with high analysis precision and accuracy on opinions' extraction and sentiment analysis are helpful to the e-learning system. Another interesting approach is in [9] where a HMM and SVM-based hybrid learning sentiment classification algorithm has been introduced to classify the learner opinion regarding the e-learning system service to improve its performance. In [10] different possibilities aimed at automatically extracting emotions from texts have been explored: twelve essays written by a fresher student along her first semester in college are analyzed and investigated. The results support the idea of using non-intrusive emotion detection for providing feedback to students. In this paper, an approach for detecting the emotions of students in an e-learning environment by the use of the sentiment analysis is proposed. In particular, we investigate the adoption of an approach to sentiment analysis based on the Latent Dirichlet Allocation (LDA). In LDA, each document may be viewed as composed by a mixture of various topics. This is similar to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior. By the use of the LDA approach on a set of documents belonging to a same knowledge domain, a Mixed Graph of Terms can be automatically extracted [11][12]. Such a graph contains a set of weighted word pairs, which we demonstrate to be discriminative for sentiment classification. The proposed approach has been applied to a real case: the blended course of Software Technologies for the Web held in the University of Salerno's Computer Science School. The organization of this paper is the following: in section 2 related works on sentiment analysis are discussed; section 3 discusses briefly the extraction of a Mixed Graphs of Terms from a document corpus and main features. Section 4 introduces the proposed approach while section 5 discusses experimental results.

II. Related Works

In literature, there are many approaches related to the sentiment analysis [4][5][25][24][23]. In particular, some approaches attempt to classify the sentiment at a document level. In [22] authors introduce an approach based on the algebraic sum of the orientation terms (positive or negative) for document classification. Starting from this approach other techniques have been developed [21]. Baroni [20] proposed to rank a large list of adjectives according to a subjectivity score by employing a small set of manually selected adjectives and computing the mutual information of pairs of them using frequency and co-occurrence frequency counts on the web. Starting from this approach many researchers developed "sentiment" lexicon. The work of Turney [19] proposes an approach to measure the semantic orientation of a given word based on the strength of its association with a set of context insensitive positive words minus the strength of its association with a set of negative words. By this approach sentiment lexicon can be built and a sentiment polarity score can be assigned to each word [18][17]. Artificial intelligence and probabilistic approaches have been adopted for the sentiment mining. In [16] three machine learning approaches (Naive Bayes, Maximum Entropy and Support Vector Machines) have been adopted to label the polarity of movie reviews. A promising approach has been developed in [15] where a novel methodology has been obtained by the combination of rule-based classification, supervised learning and machine learning. Another interesting approach is in [14] where a probabilistic model, the Sentiment Probabilistic Latent Semantic Analysis (S-PLSA), has been adopted [13]. The S-PLSA is an extension of the PLSA where it is assumed that there are a set of hidden semantic factors or aspects in the documents related to each other according to a probabilistic framework. In this paper the adopted approach is the one introduced in [4]. In the next paragraph the proposed approach will be described.

III. What is Mixed Graph of Terms?

In this section we explain how a Mixed Graph of Terms can be extracted from a corpus of documents. The Feature Extraction module (FE) is represented in Fig. 1. The input of the system is a set of documents.

After the pre-processing phase, which involves tokenization, stop words filtering and stemming, a Term-Document Matrix is built to feed the Latent Dirichlet Allocation (LDA) [27] module. The LDA algorithm, assuming that each document is a mixture of a small number of latent topics and each word's creation is attributable to one of the document's topics, provides as output two matrices - Θ and Φ - which express probabilistic relations between topic-document and word-topic respectively. Under particular assumptions [26], LDA module's results can be used to determine: the probability for each word v_i to occur in the corpus (W_A) ; the conditional probability between word pairs (W_C); the joint probability between word pairs (W_J). Details on LDA and probability computation can be found on [26]. Defining *Aggregate roots* (AR) as the words whose occurrence is most implied by the occurrence of other words of the corpus, a set of H aggregate root $r=(r_1,\ldots,r_H)$ can be determined in the following way:

$$r_i = \text{argmax}_{v_i} \prod_{j \neq i} P(v_i|v_j) \quad (1)$$

This phase is referred as Root Selection (RS) in Fig. 1. A weight ψ_{ij} can be defined as a degree of probabilistic correlation between AR pairs: $\psi_{ij} = P(r_i, r_j)$. We define an *aggregate* as word v_s having a high probabilistic dependency with an aggregate root r_i. Such a dependency can be expressed through the probabilistic weight $\rho_{is} = P(r_i|v_s)$. Therefore, for each aggregate root, a set of aggregates can be selected according to the highest weight values. As result of the Root-Word level selection (RWL), an initial mGT structure, composed by H aggregate roots R_l linked to all possible aggregates W_l, is obtained. An optimization phase allows neglecting weakly related pairs according to fitness function [26]. In particular, the proposed algorithm, given the number of aggregate roots H and the desired max number of pairs as constraints, chooses the best parameter settings τ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_H)$ defined as follows:

- τ : the threshold that establishes the number of aggregate root/aggregate root pairs. A relationship

between the aggregate root r_i and aggregate root r_j is relevant if $\psi_{ij} \geq \tau$.

- μ_i: the threshold that establishes, for each aggregate root v_i, the number of aggregate root/word pairs. A relationship between the word v_s and the aggregate root r_i is relevant if $\rho_{is} \geq \mu_i$

A mixed graph of terms is then built from several clusters, each containing a set of words v_s (aggregates) related to an (aggregate root) r_i, the centroid of the cluster. Some aggregate roots are also linked together building a centroids subgraph

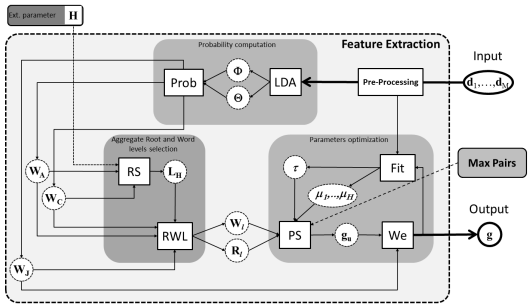


Figure 1. *mGT Feature Extraction Module*

IV. Searching the Sentiment by the use of the Mixed Graph of Terms

As described in the previous section, a Mixed Graph of Terms gives a compact representation of a set of documents related to a well-defined knowledge domain. In this way the obtained graph can be considered as a filter to be employed in document classification problems. The main aim of this paper is to show how mGT can be effectively applied for sentiment mining from texts: the proposed method can be used to build a sentiment detector able to label a document according its sentiment. Our system is composed by the following modules:

- **Mixed Graph of Terms building module**: this module builds a mixed graph of terms starting from a set of documents belonging to a well-defined knowledge domain and previously labeled according the sentiment expressed in them. In this way the obtained mixed graph of terms contains information about the words and their co-occurrences so representing a certain sentiment in a well-defined knowledge domain. As described in section 3 thanks to the LDA approach such a graph can be obtained by the use of a set of few documents. In figure 2 the module architecture and its main functional steps are depicted. The output of this module is a mixed graph of terms representing the documents and their sentiment. By feeding this module with positive or negative training sets, it will be possible to build mixed graphs of terms for documents that express positive or negative sentiment in a well-defined domain.
- **Sentiment Mining Module**: this module extracts the sentiment from a document thanks to the use of the Mixed Graph of Term as a sentiment filter. The input of this module is a generic document, the mixed graph of terms representing positive and negative sentiment in a knowledge domain and the output is the sentiment detected in the input document.

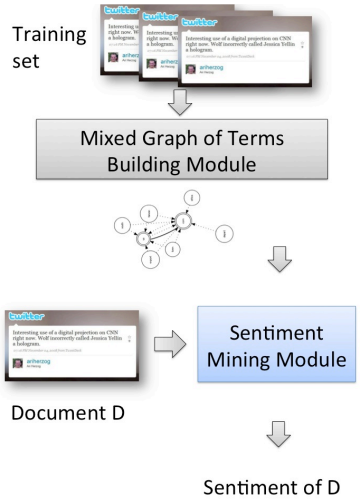


Figure 2. *Sentiment Analysis System Architecture mGT*

The sentiment extraction is obtained by a comparison between document and the mixed graph of terms according to the following algorithm:

Sentiment_Mining_Algorithm

Input: W = [w_1, w_2, …, w_N] the words that are in a Document D belonging a knowledge domain K; the mixed graph of terms mGT_+ and mGT_- obtained analyzing documents related to the knowledge domain K expressing positive and negative sentiment; RW_+ = [rw_1, rw_2, …, rw_t] the aggregator words that are in mGT_+; AW_+ = [aw_1, aw_2, …, rw_m] the aggregated words that are in mGT_+; RW_- = [rw_1, rw_2, …, rw_n] the aggregator words that are in mGT_-; AW_- = [aw_1, aw_2, …, rw_p] the aggregated words that are in mGT, L an annotated lexicon

Output: $Sentiment_D$ = {Positive, Negative, Neutral} the sentiment expressed in the document D

Algorithm Description

f_p=0;
f_n=0;

Determining the synonyms for each word belonging to the vector W

for i=0 -> Length[W]
WS= WS + Synset[L, W[i]];

```
end for
W=W+WS
```

Mining the sentiment from the document

```
for i=0 -> Length[W]
        for k=0 -> Length[RW+]
                if(RW+ [k] == W[i])
                        fp = fp + 2;
                end if
        end for
        for k=0 -> Length[RW-]
                if(RW- [k] == W[i])
                        vfn = fn + 2;
                end if
        end for
        for k=0 -> Length[AW-]
                if(AW+ [k] == W[i])
                        fp = fp + 1;
                end if
        end for
        for k=0 -> Length[AW-]
                if(AW- [k] == W[i])
                        fn = fn + 1;
                end if
        end for
end for
```

Determining the Sentiment

```
if fp / fn > 1.5
        SentimentD = Positive
else
        if fn / fp > 1.5
                SentimentD = Negative
        else
                SentimentD = Neutral
        end if
end if
```

The proposed algorithm requires the use of an annotated lexicon, as for example WordNet or ItalWordNet, for the retrieval of synonyms of the words contained in the document D and not included in the reference mGT. The retrieved synonyms are added to the vector W and analyzed according to the classification strategy. The proposed approach is effective in an asynchronous sentiment classification, but can work also in a synchronous way. In figure 3 the synchronous sentiment real time classificatory architecture is depicted. For real time working two new modules have been introduced:

- **Document Grabber**. This module aims to collect documents from web sources (social networks, blogs and so on). These documents can be collected both for updating the training set and for their classification according to the sentiment. The training set update is an important feature of the proposed approach. In this way, in fact, the various mGTs can be continuously updated and improve their discriminating power introducing new words and relations and deleting inconsistent ones.
- **Document Sentiment Classification**. The new documents inserted into the training set have to be classified by the support of an expert. The aim of this module is to provide a user friendly environment for the classification, according to their sentiment, of the retrieved documents.

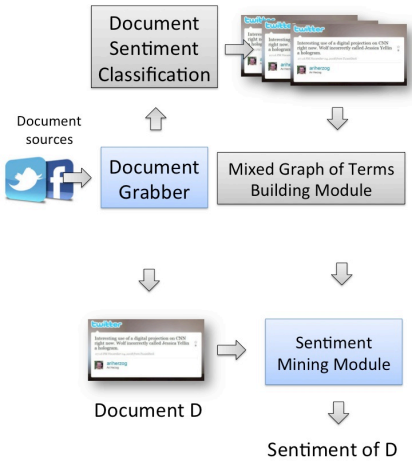


Figure 3. *System Architecture for Synchronous Classification*

V. Experimental Results

The evaluation of the proposed method has been conducted through two steps. Firstly the proposed approach has been applied on a standard dataset: the Movie Reviews Dataset [16]. The main aim of this experimentation was to evaluate method's performance and make a comparison with the other approaches well known in literature. The experimentation has been conducted considering the 25% of the dataset as training set and the remaining 75% as test set. The obtained results and their comparison with other approaches are depicted in table 1. From the table 1 it can be observed that the proposed approach shows the best results from the point of view of accuracy.

Table 1. The accuracy obtained by the various methods on the considered standard dataset.

Referenc	Methodology	Accuracy
[14]	Support Vector Machines	82,90%
		81,50%
	Naïve Bayes	81,00%
mGT Approach	Maximum Entropy	88,50%
	LDA	

The second experimental phase has been carried out using three different datasets. The experimental scenario involved the analysis of posts collected from the popular e-learning platform Moodle. In particular, the courses of Web Software Technologies (Period: September – December 2016, Number of Students: 63), Computer Networks (Period: March – July

2017, Number of Students: 27) and Introduction to Computer Science (Period: September – December 2016, Number of Students: 126) has been held by the use of a blended approach.

The courses has been organized in the following topics:

Web Software Technologies

- Apache Technologies
- XML Language
- HTML Language
- Servlet and JSP
- PHP Language
- Ajax

Computer Networks

- Introduction to Computer Networks
- Application Layer
- Transport Layer
- Network Layer
- Data-Link Layer
- Cyber Security

Introduction to Computer Science

- Introduction to Computer Architecture
- Operating Systems Architecture
- Algorithm

C Language

For each topic a final test has been submitted to the students. The traditional lectures have been supported by the use of additional learning contents distributed by the use of Moodle. Chat and forum enhanced the collaborative approach of the course. Some twitter hashtags supported the courses. The contents exchanged by the use of forum and chat have been set not visible for the teacher and this policy was known by students. A Sentiment Analysis Module has been used for grabbing the mood of each student during the various lectures related to the various topics. The real time analysis of the comments furnished a sort of thermometer of the mood of classroom regarding to the various topics. In table 2 the number of the posts collected from the chat and the forum for each topic has been reported. Also the relative retrieved sentiment has been reported in terms of positive and negative percentage. The observation period expresses the length of the course's section dedicated to a certain topic.

It is interesting to notice the increasing trend of the positive sentiment during the observation time. The reason of this trend is almost clear: at the beginning of each topic students showed a natural disorientation that is greater for topics related to new technologies or concepts. After these first phases teacher updated his teaching style according to the sentiment of the students giving them more contents or introducing more examples or exercises. For example, in the case of PHP language teacher introduced a series of solved exercises and this kind of support had a positive effect on the students. The same approach worked well in the case of C Language. The evaluation of sentiment, furthermore, can highlight the topics which are more complex for the students. In this way, at the end of the course teacher can improve the teaching approach. In general, teachers appreciated the Sentiment Grabber tool above all for the opportunity to manage the mood of the class without the filter of the relationship teacher – student.

VI. CONCLUSIONS

This paper proposes the use of the mixed graph of terms, obtained by the use of Latent Dirichlet Allocation approach, as tool for the sentiment classification of documents. The method relies on building the reference mGTs from documents labeled according their sentiment. The classification of a document can be conducted by using the reference mGTs. The proposed method has been applied in the e-learning field for measuring the mood of a classroom towards some topics. Further development of this approach will include the introduction of annotated lexicon, as SentiWordnet, for a better sentiment evaluation of the words and the sentence structures.

REFERENCES

[1] Haji H. BINALI, Chen WU, Vidyasagar POTDARA, "New Significant Area: Emotion Detection in E-learning Using Opinion Mining Techniques", 3rd IEEE International Conference on Digital Ecosystems and Technologies, pp. 259-264, 2009

[2] N. Hara and R. Kling, "Students' distress with a web-based distance education course", Information, Communication & Society, vol. 3, pp.557-559, 2000.

[3] Bing Liu. Sentiment analysis and subjectivity. In Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca, 2010.

[4] Francesco Colace, Massimo De Santo, Luca Greco, "A Probabilistic Approach to Tweets' Sentiment Classification", ACII 2013: 37-42

[5] Daniel McDuff, Rana El Kaliouby, Evan Kodra, Rosalind W. Picard, "Measuring Voter's Candidate Preference Based on Affective Responses to Election Debates", ACII 2013, 369-374

[6] Dan Song, Hongfei Lin, Zhihao Yang, "Opinion Mining in e-Learning System", 2007 IFIP International Conference on Network and Parallel Computing – Workshops, 2007

[7] H. Binali, V. Potdar, and C. Wu, "A State Of The Art Opinion Mining And Its Application Domains," in Proceedings of ICIT09, 2009

[8] D. Song; H. Lin; Z. Yang, "Opinion Mining in e-Learning System" 2007 IFIP International Conference on Network and Parallel Computing – Workshops. pp: 788 – 792.

[9] Mohamed Ben Ammar, Mahmoud Neji, Adel M. Alimi, Guy Gouardères, "The Affective Tutoring System", Expert Syst. Appl. 37(4): 3013-3023 (2010)

[10] Pilar Rodríguez, Alvaro Ortigosa, Rosa M. Carro: Extracting Emotions from Texts in E-Learning Environments. CISIS 2012: 887-892

[11] Francesco Colace, Massimo De Santo, Luca Greco, Paolo Napoletano: Text classification using a few labeled examples. Computers in Human Behavior 30: 689-697 (2014)

[12] Shi-Kuo Chang, Francesco Colace, Lei Zhao, Yao Sun: rocessing Continuous Queries on Sensor-Based Multimedia Data Streams by Multimedia Dependency Analysis and Ontological Filtering. International Journal of Software Engineering and Knowledge Engineering 21(8): 1169-1208 (2011)

[13] Hofmann, T. (1999), Probabilistic Latent Semantic Analysis, *in* 'In Proc. of Uncertainty in Artificial Intelligence, UAI'99', pp. 289--296.

[14] Yu, X.; Liu, Y.; Huang, X. & An, A. (2012), 'Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain', *IEEE Trans. on Knowl. and Data Eng.* **24**(4), 720--734.

[15] Prabowo, R. & Thelwall, M. (2009), 'Sentiment analysis: A combined approach', *Journal of Informetrics* **3**, 143--157.

[16] Pang, B.; Lee, L. & Vaithyanathan, S. (2002), Thumbs Up? Sentiment Classification Using Machine Learning Techniques, *in* 'Proceedings of EMNLP', pp. 79--86.

[17] Gamon, M. & Aue, A. (2005), Automatic Identification of Sentiment Vocabulary: Exploiting Low Association with Known Sentiment Terms, *in* 'Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing', Association for Computational Linguistics, Ann Arbor, Michigan, pp. 57--64.

[18] Neviarouskaya, A.; Prendinger, H. & Ishizuka, M. (2011), 'SentiFul: A Lexicon for Sentiment Analysis', *Affective Computing, IEEE Transactions on* **2**(1), 22 -36.

[19] Turney, P. & Littman, M. (2002), 'Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus', Technical report, Institute for Information Technology, National Research Council Canada.

[20] Baroni, M. & Vegnaduzzo, S. (2004), Identifying Subjective Adjectives through Web-based Mutual Information, *in* 'In Proceedings of the 7th Konferenz zur Verarbeitung Nat¸rlicher Sprache (German Conference on Natural Language Processing ñ KONVENSí04', pp. 613--619.

[21] Wilson, T.; Wiebe, J. & Hwa, R. (2004), Just how mad are you? finding strong and weak opinion clauses, *in* 'Proceedings of the 19th national conference on Artifical intelligence', AAAI Press, , pp. 761--767.

[22] Colbaugh, R. & Glass, K. (2010), Estimating sentiment orientation in social media for intelligence monitoring and analysis, *in* 'Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on', pp. 135 -137.

[23] Cambria, E.; Song, Y.; Wang, H. & Howard, N. (2013), 'Semantic Multi-Dimensional Scaling for Open-Domain Sentiment Analysis', *Intelligent Systems, IEEE* **PP**(99), 1-1.

[24] Wang, H.; Can, D.; Kazemzadeh, A.; Bar, F. & Narayanan, S. (2012), A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle, *in* 'Proceedings of the ACL 2012 System Demonstrations', Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 115--120.

[25] Pang, B. & Lee, L. (2008), 'Opinion Mining and Sentiment Analysis', *Found. Trends Inf. Retr.* **2**(1-2), 1--135.

[26] Colace, F.; Santo, M. D.; Greco, L. & Napoletano, P. (2013), A Query Expansion Method based on a Weighted Word Pairs Approach, *in* 'Proceedings of the 3nd Italian Information Retrieval (IIR)', CEUR-WS.org

[27] Blei, D. M.; Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993--1022.

[28] Reisman, S., The Future of Online Instruction, Part 1, Computer, Year: 2014, Volume: 47, Issue: 4

[29] Reisman, S., The Future of Online Instruction, Part 2, Computer, Year: 2014, Volume: 47, Issue: 6

[30] Dodero, J.M.; Garcia-Penalvo, F.-J.; Gonzalez, C.; Moreno-Ger, P.; Redondo, M.-A.; Sarasa-Cabezuelo, A.; Sierra, J.-L., Development of e-Learning Solutions: Different Approaches, a Common Mission, Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, Year: 2014, Volume: 9, Issue: 2

[31] Garrido, A.; Morales, L., E-Learning and Intelligent Planning: Improving Content Personalization, Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, Year: 2014, Volume: 9, Issue: 1

[32] Zemirline, N.; Bourda, Y.; Reynaud, C., Expressing Adaptation Strategies Using Adaptation Patterns, Learning Technologies, IEEE Transactions on Year: 2012, Volume: 5, Issue: 1.

[33] Colace, F., Santo, M. D., Lemma, S., Lombardi, M., Rossi, A., Santoriello, A., Terribile A., Vigorito, M. (2016). "How to describe cultural heritage resources in the web 2.0 era?" Paper presented at the Proceedings - 11th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2015, 809-815.

[34] Clarizia, F., Colace, F., De Santo, M., Lombardi, M., Pascale, F., & Pietrosanto, A. (2018). "E-learning and sentiment analysis: A case study." Paper presented at the ACM International Conference Proceeding Series, 111-118.

[35] Colace, F., De Santo, M., Pascale, F., Lemma, S., & Lombardi, M. (2017). "BotWheels: A petri net based chatbot for recommending tires. " Paper presented at the DATA 2017 - Proceedings of the 6th International Conference on Data Science, Technology and Applications, 350-358.

[36] Colace, F., Lemma, S., Lombardi, M., & Pascale, F. (2017). "A context aware approach for promoting tourism events: The case of artist's lights in salerno." Paper presented at the *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, , 2* 752-759.

[37] Clarizia, F., Lemma, S., Lombardi, M. & Pascale, F. 2017, An ontological digital storytelling to enrich tourist destinations and attractions with a mobile tailored story. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 10232 LNCS, 2017, Pages 567-581.

[38] Clarizia, F., Lemma, S., Lombardi, M. & Pascale, F. 2017, A mobile context-aware information system to support tourism events. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 10232 LNCS, 2017, Pages 553-566.

Web Software Technologies						
Topic	Beginning		Middle		End	
	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood
Apache Technologies	65	21	75	13	81	10
XML Language	30	42	38	46	54	34
HTML Language	60	20	71	9	84	6
Servlet and JSP	20	30	31	35	51	38
PHP Language	31	37	54	19	72	11
Ajax	43	21	67	32	82	11

Introduction to Computer Networks						
Topic	Beginning		Middle		End	
	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood
Introduction to Computer Networks	68	27	74	21	86	10
Application Layer	74	22	82	12	87	7
Transport Layer	22	67	34	54	52	32
Network Layer	26	45	45	32	59	21
Data-Link Layer	43	29	56	21	81	12
Cyber Security	67	21	79	10	91	5

Introduction to Computer Science						
Topic	Beginning		Middle		End	
	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood	% Positive Mood	% Negative Mood
Introduction to Computer Architecture	49	27	65	19	81	14
Operating Systems Architecture	39	34	56	27	79	18
Algorithm	34	51	47	39	61	21
C Language	16	76	33	56	59	39

Figure 4. *Obtained Results. The Sentiment has been measured three times: at the beginning of a new topic, at the middle and at the end. The system measured Positive, Negative and Neutral mood. In the figure there are the percentage of Positive and Negative comments.*

Implications of learning environments on the Information Systems of educational institutions

Paolo Maresca[1], Andrea Molinari[2,3]

[1]Department of Ingegneria Elettrica e Tecnologie delle Informazioni (DIETI) – Università Federico II di Napoli (ITALY)

[2]Laboratory of Maieutics – Dept. of Industrial Engineering - University of Trento - Italy
[3]School of Industrial Engineering and Management, Lappeenranta University of Technology (FINLAND)
andrea.molinari@unitn.it

***Abstract*— Since the early years where they started to enter the market, Learning Management Systems (LMS) have reached a very high level of maturity, providing professional solutions to mostly any educational need referring to distance learning. In this paper, an analysis of how LMSs should evolve in the future is presented, according to authors' experience, in terms of functionalities and services provided to users. Behind these new functionalities and services, we foresee research fields that could provide interesting and fruitful stimulus to the market and to these platforms. The foreseen direction is the one that goes towards an expansion of the collaboration services, where virtual learning environments should be mixed with typical Computer Supported Collaborative Work (CSCW) tools and approaches that put collaboration at the heart of the system. Nevertheless, also traditional e-learning services should be improved with additions coming exactly from this integration with cooperative / collaborative services. The reference point is a virtual community platform created and developed along the years, used in the authors' institutions and in several public and private organizations. The platform is oriented towards the support of collaborative processes, where of course e-learning is one of the most important, but not the only one, and where new services supporting collaboration in different ways are constantly added.**

Keywords-component; Learning Management System, Information System, customization, open source software

I. INTRODUCTION

Computers today play a central role in many sectors of our life, by the presence of hardware and software tools covering most of tasks human beings perform. Education is not excluded from this list, both for content providing and for supporting educational tasks with Learning Management Systems (LMSs), Virtual Learning Environments (VLEs) or other labels that refer to software platforms and services devoted to education.

The e-learning sector is well guarded by different groups of LMS platforms. The first group of LMSs is based on open-source solutions, open, free or both, readily available, at no cost of acquisition (if the configuration needed by the educational institution is simple), with source code available, requiring an in-depth knowledge for management and customization. These aspects have created an interesting market for consultancy and services devoted to the customization of the platform for specific needs, like the integration with other services of the information system. Here very well-known platforms are available, being Moodle™ the most famous one.

A second group include the so-called "closed" solutions, in the past linked to major players, now mostly developed on the basis of specific requirements expressed by major customers. In this category many variegated examples can be found, with solutions created from scratch, customizations of open source LMSs, or customization of other software platforms created for other purposes "forced" to become technology-enhanced learning environments. The most frequent case is the customization of Content Management Systems (CMS), like Joomla™, Drupal™, WordPress™ etc. to some educational needs.

Recently, a third group of software solutions for education can be identified that take advantage of the many positive aspects of cloud computing. Normally these platforms are the porting of one of the previous categories, or native platforms only available via cloud services.

This paper is based on almost 30 years of experience of authors in the field of creation of software solutions for education, specifically the creation of Virtual Learning Environments for different public and private institutions. The paper will discuss the pros and cons of one of the aforementioned groups, i.e., the custom solution. It is our strong convincement that, in many situations, a customized LMS provides better services from the educational perspective, but most of all it provides services to other sectors of the information system of the institution that normally are not labelled as "educational services", but that could be found inside LMS. Moreover, equipping a LMS with services not only devoted to the pure educational context, but related to the support of collaborative tasks, could provide a lot of advantages for the institution and for the administrators of the information system itself. The paper is organized as follows: section 2 will present the testbed for our argumentation, a custom virtual learning environment created to support digital training processes. Section 3 will be a frank analysis of the implications of open-source learning environments on the information systems of educational institutions, with respective pros and cons. Section 4 will discuss the relation between LMS and more general information systems of educational institutions, while section 5 will present an example of integration related with tools for decision support systems.

DOI reference number: 10.18293/DMSVIVA2018-010

II. THE TESTBED: A CUSTOM VIRTUAL LEARNING ENVIRONMENT

This paper presents an analysis of the opportunities related with the use of e-learning services in different contexts respect to pure training, and the integration of these tools with the rest of the information system of the organization. Normally, e-learning is perceived as a separate world respect to the information system. However, when the size and variety of e-learning needs grow, turning the platform from a simple repository of material to a tool devoted to integration, collaboration and cooperation between virtual communities, at that point the management of e-learning services becomes much more complicated. It is exactly in these contexts where both open-source platforms and closed-source mainly fail, and in our opinion this is due to their conceptual foundations. What has been experienced is that distance education is nothing but a tool for collaboration between teacher and participants, but extending these tools to other contexts significantly expands the application fields. In what follows, however, an adaptation of the platforms, their customization, or assembly of different tools in "patchwork" often reveals to be inefficient and unusable. In general, this means for organizations to heavily intervene through customizations on platforms created by others, often distorting and then losing or compromising compatibility with future releases, or devoting considerable efforts to keep this compatibility. The growing phenomenon of MOOCs, for example, sees a proliferation of platforms created to handle these complex contexts of massive training, thus forcing educational institutions to adapt their educational model, services, processes etc. to what the LMS provides.

In our opinion, the flow should work in reverse: software platform should be customized on the educational processes that the institution decides to apply, and this creates a competitive advantage for educational institutions. Secondly, the integration with the rest of the information system is crucial to the success of the institution, or at least of the educational initiative. Nowadays software educational platforms should provide services that include the administrative components of educational services, like enrolment, taxes, exam records, students' secretary, single sign-on, certifications, online payments etc., being these services typically provided by the main information system through the organization website.

To validate our argumentation, the authors will present their experience in the creation of a custom platform constantly developing since 1998. Some cases and situations of partners that adopted our platform and collaborated with the team to implement the integration with the respective information systems will be presented,. The system development started at University of Trento for blended teaching 20 years ago. The development started in 1998, largely before the advent of Moodle™ or similar platforms: at the time, there was a market of web-based Learning Management Systems, and the dominant player was BlackBoard/WebCT™. After having finished the first version, in the academic year 1999-2000 the Faculty of Economics of the University of Trento decided to adopt our software system in order to enhance its traditional educational activities. This platform should have absorbed the many different personal initiatives taken by several teachers who had activated autonomous web pages to support their courses. Three options were presented to the Faculty: purchasing commercial software, using free software or building from scratch a new platform: a very similar situation compared to today's alternatives. The decision to build its own platform was a consequence of various reasons [1], which can be summarized as follows. At the time, the use of commercial software appeared to be impossible due to very high costs, considering the total cost of ownership of such solutions: acquisition, maintenance, management, training, software insurance, hardware required, personnel etc. On the other hand, at that time free software was rather rudimentary (if not in a prototype stage), and was limited to very few examples mainly created by single research groups / Universities / freelance consultants.

After a first 5-year of extensive usage, our team focused on carrying out a platform based on the idea of virtual communities. Facebook was probably still in the creators' mind, so the idea of virtual community, according to our interpretation, was not the result of a process of social networking. In fact, it was (and it is in the current implementation) a virtual space shared by people with a common goal, following approximately the original definition of Rheingold back in 1993 [8]. A community's virtual space can be simple or complex; for example, it can contain further virtual communities, thus establishing a hierarchical "parent-child" relationship. The (virtual) community can be an open space accessible to anyone, or can be a restricted space, the access to which is reserved only for some people authorized by the community administrator. The users can have different roles with rights and duties, which vary in the use of space and collaboration services activated in a virtual community. The system maintains the consistency of the completely social environment of the virtual communities, which are active at a given time, in that it provides users of a community with a range of on-demand services that can be activated and used in accordance with the permissions granted and the roles assigned.

Respect to the change of paradigm from a LMS based on the traditional metaphor of a "class" to the metaphor of a "virtual community", here some observations are summarized:

- Models of teaching / learning (such as learning by problems, learning by projects, cooperative learning and their combinations) can hardly be connected to the idea of a class, especially when the software directly represents the metaphor of traditional courses;
- The needs for cooperation within the academic environments can be extended to all the activities that constitute the context in which didactics takes place, not only to the simple activity of teaching. The organization of a research group, for example, is surely a (virtual) community that requires many of the services used in a (virtual) classroom: file repository, videoconferencing, forum, FAQ, blog etc., but surely should not be organized as a (virtual) class: different roles of participants and different services needed;
- The organizational scenario is changing under the effects of new regulations or exogenous decisions, and these changes will inevitably reflect on the LMS functionalities. It is important to note that these changes are usually the result of a debate process in which both

elements of cooperation and negotiation interact, and very often are on a national scale if not regional if not of the single University. Expecting that a world-wide software platform (like Moodle) will add features (sometimes very impacting) for such specific context is rather unlikely, while respectable local attempts to create special plugins can clash with Universities that adopt internationalized versions of the platform;

- The educational processes of a University are not built only as a set of lectures and exams, but these activities are inevitably intertwined with the university's organization and its information system;
- In academic contexts, not everything concerns teaching: for example, the entire Faculty is more than a container of degree courses, and a degree course is more than a container of lessons. So the hierarchy of the organization is relevant for any software platforms, LMS included, for example for the propagation and sharing of documents at different levels of the hierarchy. A general communication of the Dean to all the communities of the Faculty could be propagated without replicating the file in any classroom by simply implementing an inheritance mechanism among communities. The hierarchy mechanisms and the connected propagation effects are normally not implemented in mainstream LMSs, while our platform has these mechanism built-in by design.

To answer these (and other) needs, it was necessary to find another founding metaphor respect to what LMSs have implemented implicitly or explicitly in their code, which had at least three basic characteristics: a) to be general to support any collaboration process, not only learning processes; b) to be capable of modelling adequately the organizational aspects of an educational institution c) to be flexible to provide services to the rest of the information system. This metaphor was found in the concept of virtual community. The system that arose, called "Online Communities", started to offer its services in 2003 and runs uninterruptedly since then. It is still the platform at the Faculty of Economics and at other Faculties of our university, and since then has been adopted by large public bodies and private organizations.

Fig.1 The internationalized home page of "Online Communities"

The complexity of managing virtual communities is objectively quite different from managing a course [2]. It requires a different approach also in the management of roles and permissions. In the logic of integrating systems, there is an ever increasing need to provide a single point of aggregation of the various services in order to enable subjects and systems with different interests (if they are not divergent) to access the same object, acting according to their own competences.

The architecture of Online Communities is based on five fundamental entities: Person, Community, Role and Permission, and the combination of the roles and permissions that gives the Profile for each user. The central entity of the platform is the "virtual community". The main characteristics of a community could be summed up as follows:

- each Community encapsulates a certain number of services.
- The services are general applications that enable users to publish contents, to communicate in synchronous and asynchronous way, to exchange files, to coordinate events, to manage their personal learning environments etc.
- Services for each community are activated by an administrator of the community according to the community members' needs, and the users of a community can use them with different permissions that are specific for each service. The role of the administrator of the community is clearly crucial, not complex in technological terms but in an organizational sense.
- The communities can be aggregated into larger communities with hierarchic mechanisms and infinite nesting levels. The communities can also be aggregated in any arbitrary way into larger communities disregarding the possible position in a hierarchical structure.
- There is no anonymous access to the platform: being the user's profile the base for every operation, all users are profiled in the platform at least with one role and one community of belonging.

Over the last few years the system has evolved into a platform for professional training oriented to life-long learning outside academia, being preferred to mainstream LMSs because of three main reasons:

- the complete knowledge of the University development team on every single part of the platform, due to the complete in-house, from-scratch development;
- the metaphor of the virtual community that particularly fits with many organizational needs and educational methodologies used, more oriented towards a peer-to-peer, equal relationship within the participants of a community;
- The predisposition to be integrated with other components of the information systems, and the provision of services to be encapsulated in other components of the hosting information system.

The new implementation of the system (fig.1) has retained certain basic features of earlier versions, while also extending its functions in order to allow the application of business logics to

the training processes. Such evolution has been required, for example, when the Massive Open Online Courses (MOOCs) idea came into the market. This brought the need to develop previously neglected aspects, especially with the aim of controlling the students' activities more extensively, and the accounting issue of invoicing participants precisely the amount of usage of the platform for their training processes.

Fig.2 The User's home page, with the communities of interest

The connection with the Enterprise Resource Planning (ERP) software of the hosting organization has been another good example of our argumentation, being this the need of evolution of LMSs from a general-purpose platform of a generic educational institution, similar if not equal all over the world. On the contrary, what should be highlighted is that such an effective technological tool should embrace the (social and technological) context where teaching and learning processes take place, including other processes of the information systems. Going back to the origin of ERP, the problem was exactly the same: different silos of information systems treating the same data, but separated and not interconnected, with the consequent mess of customization and integration that created so many issues in the management of information systems. The solution and the time, still valid today, was to have a centralized system with a unique database where one single copy of information was managed. What proposed is an update of this idea to modern Restful web services, cloud computing, distributed databases etc. where best-of-breed services are provided to users by whatever platform inside the information system has been elected as the most suitable platform for that service.

III. Implications of open-source learning environments on the Information Systems of educational institutions

First, the authors want to clarify that the technical and organization value of platforms like Moodle is not under discussion. Moodle and the like changed the world of education because they supplied an easy and quick way to address the request of providing educational services through the web. Our argumentations start from a different perspective, i.e., the need of a mature institution that wants to apply a unique, customized, "personal" set of educational practices, being convinced that customized educational practices instead of standardization imposed by a software platform could be a competitive advantage. (differentiation from the other educational institution)

Public and private educational institutions adopted mainly open source solutions for various (quite obvious) reasons, substantially choosing the no-cost (or this was what they believed), easy way. Respect to this, authors experienced different issue. As a first element, the need of a development team that knows the platform, but being the platform developed by others (many others, in the case of Moodle), substantially the development is confined to a very limited customization, with the general motto "don't touch what you have not coded". So unless the institution has the technical background to fully manage the LMS, from hardware to software to network, having the source code of the LMS (like in Moodle) has a very limited value, and in the end leads to hire external consultants for the installation, the maintenance, the personalization etc., thus vanishing the expected benefits of "free" in the sense of zero-cost. To complete the matter, many Moodle owners know very well the famous "security patch" hassle, and the costs associated with mistakes on this side. There is clearly nothing new respect to any other software platform, but the lack of awareness of many (especially small) educational institutions created a very bad reputation to LMSs, thus hiding the enormous benefits they could bring.

Even if it is a technical issue, scalability is clearly something that in large organizations became a sort of buzzword, while Moodle was mainly created as a single-server box, one for each customer. If the institution is experimenting, for example, peaks of usage during the early days of an academic year, scalability becomes a serious issue. If we use the platform, for example, for a social event where all participants (students) will get a gadget, and in the meantime there are some online exam sessions, then scalability will become a serious issue. Theoretically, no problems exist in putting Moodle in the cloud, but then a) you'll need to spend extra money and resources to deliver this, but especially for public institutions b) not every organization is happy to publish online material dealing with internal topics (for example on security training policies and processes) on a cloud-based platform.

There is also an over-emphasis on the capabilities of customization of free/open source LMSs. If Moodle is taken as a reference, at the moment of writing the core components of this platform are around 800k of PHP lines of code, and close to 100k for Javascript. This excludes all the external libraries, modules etc. It is clearly a huge software effort, and whoever wrote a single line of code knows perfectly the possibility for an external person to safely and consciously put their hands inside this mass of code. Therefore, most of the time, when people claim "we have customized Moodle", they refer to some CSS style changes in visual aspects, labels, some logos, menus and very few other things. Real customization means, for example, to change the structure of a database table in order to add information coming from another component of the organization's information system, in order to connect the two systems, and to create this connection bi-directionally. The closest way to this request is to install a Moodle plug-in, but here other problems rise, related with the enormous amount of plugins from different sources of different quality, their reliability and stability in case of version change, the overlapping of functionalities among different plugins, the availability of more plugins for the same function etc. Even the

simple change of the layout of a page, or of some pages of a certain service, or modifying some dashboards becomes complex, available only to seasoned developers with core competences in Moodle and with a deep knowledge of what will happen if that feature will be changed. Again, it is very well known, and also comprehensible from the perspective of Moodle's maintenance team, that in order to avoid instability and incompatibility situations, there are many roadblocks to (even) modest customization, forcing you substantially to consider the forking of the entire platform as an alternative. Forking is the very last resort for any institution, and the main reason why our "Online Communities" platform found some believers is mainly in this point.

Respect to this limitation in customization, another element could be considered positive in some contexts, but negative for other contexts. This could be label as "boring uniformity", in the sense that most of the institutions that adopt Moodle are stuck with the same layout Moodle provides in the default installation, and any deeper customization of the layout finds the same roadblocks seen before. This leads to a "boring" uniformity of most of the Moodle installation: the authors' never found a person that said "I change my University because I found the same Moodle".

There are many other issues that could be found, like in all software platforms, but this discussion does not want to appear like a demolition of one of the milestones of Technology-enhanced learning (TEL) like Moodle. The argumentation deals with the empty spaces left by the approach carried out by Moodle (and similar platforms) respect to a significant part of the TEL market, where there is a need of new services customized to implement new educational processes and approaches, and on the other side to connect the LMS with the rest of the Information system.

IV. LEARNING ENVIRONMENTS AND INFORMATION SYSTEM

After having presented the issues related with some approaches to LMS and their implementation, the central aspect of this paper deals with the relation between LMS and more general information systems of educational institutions. As previously stated, e-Learning platforms seem to be built to act in a restricted circle made up of teachers, tutors and students. On the contrary, in our system the community is a container ready for didactic processes, but not only: research teams, recreation groups, friends, secretariats, board of directors, sport teams, colleagues, anything that could be an aggregation of people around a scope using virtual spaces on the web.

At present, considering only the instance of the system used by the University of Trento, there are more than 7.500 active communities, 16.000 active users and in 2017 almost 500.000 unique accesses has been achieved (see Figure 2).

The evolution that Online Communities is going through implies increased implementation complexities respect to "simple" LMS settings, considering that the differences between the two approaches refer at least to four dimensions.

The first dimension is a temporal dimension. The concept is amplified on larger spectrum, that is to say, the life of the subject, not necessarily dependent on standard educational path (high school or University). At the moment, on of the largest implementation of "Online Communities" manages all the educational tasks of the largest public body in our region, i.e., the Autonomous Province of Trento, with approx.. 20.000 employees, and thousands of online courses delivered every years. The interest of the Province is clearly a long-term interest, in the perspective of managing an "educational portfolio" of the employee, thus implementing a life-long learning platform.

Figure 3. Online Communities accesses (14th march 2018)

A second dimension is the social dimension. The platform could be used in social contexts of totally diverse life-long learning settings, even in conflict with each other. Let us take as example subjects who, while interested in continuous learning, change the country of their residence, company where they work, training needs, etc. Not necessarily all the information contained in their educational portfolio are relevant for other stakeholders, or vice versa, they are very interesting for them but not the owner of the portfolio.

A third element is the spatial dimension. The place where the learner is conditions the modality of delivery of the educational contents. Let us think, for instance, at the various situated learning needs of a person responsible for maintenance, or a medical doctor when facing an emergency case, or a tourist in front of a work of art in a museum.

A final dimension, more complicated to analyse in this paper, is the anthropological one. The subject uses the platform in completely different life periods; starting with pre-school age until the end of working activity and, not to be excluded, even beyond. The problems linked to these aspects represent something extremely stimulating and yet unexplored, as it is clear (and first evidences are emerging) that our social and even mental behaviors are affected by technologies in general, and social media in particular.

The platform provides, as a set or core services, the "traditional" services provided but full-fledged Learning Management Systems: asynchronous services (like forum, agenda, upload & download of learning objects, newsgroup, notice-board, classroom and users' management, forums, blogs, wikis, FAQ etc.) and synchronous services (chat, streaming audio/video). Other than these, some customized services, closer to the

aspects of life-long learning and "training on the job" (tutorship, training on demand, research tools with problem contextualization, ticketing tools etc.) have been developed for specific partners, like the Autonomous Province of Trento.
A second set of services relates with specific integration needs with external information systems (for example, the Personnel information system of the organization) and with the acquisition of forms for external enrolment of students to university's programs. These services have been developed for institutions that have a selection process of candidates, mainly for master degrees, doctoral schools of private business courses. The Chamber of Commerce of Trento, for example, through its associated training Academy, provides many courses to affiliated companies and institutions, and heavily uses these kind of services to process enrolment, subscription to courses and even payment of fees.
A third set of services provided by the platform regard the fruition of "off-line" courses, i.e., courses already held and recorded, digitalized and made available to controlled communities of users (with the possibility to synchronize the video with slides, podcast, webcast, SCORM modules, etc.). These services are more typical of Learning Management Systems, but the issues related to the integration with a SCORM player provided us the stimulus to develop our own "meta-SCORM" engine, a service call "educational path" where many issues related to size of SCORM packages and rigidity of SCORM standards have been overcome.
As a fourth group of services, services for the creation of evaluation tests, exams, self-evaluation tests, quizzes, polls etc. could be mentioned. Together with this set, personalized reports with statistics about the users' behavior have been developed, using an internal data warehouse enriched by activity logs that overcome some problems of traditional LMSs in extracting detailed information about user performances. These specialized, business intelligence-oriented services have been developed avoiding the creation of sophisticated charting tools (already available on the market), but focusing on providing detailed information about every action that the user is performing while interacting with all the services of the platform. This allows us to follow some requirements for internal certifications.
An important category of services has been added for managing the interactions among members of the community, like project management services, agenda organization, time management, tenders and respective application forms, etc. These are the services that continuously see additions, improvements, new requests etc.
Finally, a set of mobile services to support mobile learners are provided. There are some innovative services which meet the mobility needs of the subject who wants to learn "on the move", performing learning/collaboration activities directly through his/her mobile device (mobile phone, tablet PC, smartphones, phablets, etc.).

Figure 4. A partial list of services provided by "Online Communities

The platform is constantly extended with new services, coming from research projects, users requests and the results of our almost 20-years' experience in designing, developing, implementing and using e-Learning system (LMS), with a specific approach in mind. This approach is, in some sense, "against the current" of standardization and "normalization" of LMSs, in our opinion too flattened over these pre-defined, pre-designed software platforms. From our experimentation, it is clear that an e-learning platform is not an external system respect to the rest of the information systems, but it is a crucial component for any organization. When such a platform enters into an organization, its effects are immediately visible:

- needs for integration with sub-systems existing in the organization: just to mention the simplest ones, integration with the single-sign-on system implemented in the company;
- overlapping of some functionalities of LMS/Virtual communities' platform with pre-existing functionalities in the information system of the organization. Examples: document repository, mailing distribution, virtual room management, forum, etc.;
- Competition with possible new systems entering in the organization, mainly due to the web 2.0 functionalities that nowadays most of the companies intend to implement, and that normally any (serious) LMS is able to supply:
- partially overlapping and competition with some functionalities already present, somewhere in some software.

These are the most insidious aspects, because none of the systems (LMS and other information systems) are able to satisfy the specific needs, but all of them are able in some way to supply part of the functionalities needed. The typical example found in the authors' experience is the support to document sharing for groups of people without having to mount some network disk for file sharing, normally not appreciated by system administrators, and most of the time not accessible via web. In this case, virtual communities are better candidates, as the on-the-fly creation of a virtual community with a set of services

available for the members is a perfect solution for many of these situations, not necessarily related with educational activities.

The last example is what mainly led us, in 1998, to build a new system with virtual communities as the center of our approach. At the time, Moodle™ or similar LMSs did not exist or were not accessible to most of the people, and other solutions were particularly expensive, proprietary or not available. In our vision, a virtual community is a (virtual) space of aggregation for participants, thus supporting cooperative activities among users instead of just learning activities.

As previously stated, our platform has been created to be adapted and connected to the information system of the organization. Considering e-learning and collaboration platforms as external bodies, relegated to secondary roles inside the information system, is in our opinion losing an excellent opportunity to improve collaboration and open innovation inside an organization.

Integrating eLearning systems with existing information systems is not an easy task, mainly due to some resistance and ostracism against learning applications that are seen as not relevant for the organization by the ICT departments. Other difficulties come from the technical side, due to the diversity in these systems.

Universities are using LMS mainly for issuing educational services, but many other services could be provided, expanding the role of LMS more towards information systems and collaborative platforms. It becomes essential to have advanced tools to support activities that often are not limited to training, but that widen the horizon in different contexts in which the availability of a web-based software platform is not only a big help, but an essential element to reduce space and time barriers and enable collaboration "anytime - anywhere" so much desired by the digitalized institution.

In these contexts, limiting LMSs to educational services, limitations of the conceptual and engineering nature of training processes will have to be faced. What is the authors' experience is the need of new tools and services for the educational tasks that expand the idea of training activities to the more general collaborative activities: A non-exhaustive list of these activities found very profitable if integrated in a LMS follows:

- time management at different levels of implementation: calendars, event planning, meeting management etc.
- project management, where projects can be managed with their tasks, durations, critical path, constraints, resources etc. This is profitably integrated with core services provided by the platform, like the file repository (for attaching documents to tasks and resources), forums (to discuss topics about a task with resources assigned to it), or the decision support system, for example for supporting the qualification of a duration through the interaction among experts using a multi-criteria, multi-expert fuzzy algorithm;
- processes related to support decisions in different educational contexts (exam, vote, polls, questionnaires, community participation, group democracy etc.)
- enrolment services in different situations, from enrolling in a course to a single lab session, from organizing a walk with classmate to enrol in a serious game session

Similarly, with the increase of complexity of educational activities, tools for collaboration are becoming increasingly central, like sharing and distributed decision support systems within learning communities. This is the first example described as a significant moment of integration between LMS and other technologies that normally are not available in mainstream platforms.

V. DECISION SUPPORT SYSTEMS AND LMSS

In this section, a module of the platform that provides functionalities added in order to provide support to one of a partner, the local Developing Agency (Trentino Sviluppo S.p.A.) is presented. In e-learning settings, the evaluation of different alternatives regarding learning paths' proposal is nowadays crucial, due to the great attention devoted to the construction of learning objects (LO) available through Learning Management Systems (LMS). Learning processes are normally implemented through the interaction of the learner with a LMS and, in some cases, through the usage of learning, or e-learning, paths. A learning path, as referred inside a LMS, is represented by a set of LO mixed with other tools and services available in the LMS, like questionnaires, forums, wikis, FAQ etc. This combination of information chunks and services is devoted to obtain the educational objectives defined by an instructional designer.

While testing large scale implementation of virtual community systems, authors noticed that SCORM objects and pre-defined learning paths, are more and more important in educational settings today. The market is responding to this request, thanks to adequate technologies for the design, realization and delivery of these pre-constructed educational tools. SCORM packages themselves, if well designed, could be self-consistent learning paths. According to this scenario, educational institutions and specifically the industry rather than academy, are very often in front of the process of evaluating different possible learning paths, composed by different learning objects, composing different contents and representing different approaches and responses to the educational needs stated by the educational stakeholders. The criteria for choosing which alternative better fits with these needs are most of the time based on simple considerations (mainly cost of the learning objects), taken by people with no complete view of different aspects of the learning paths, not taking into consideration all the aspects that should be needed for such an important step.

E-learning has many advantages, but for sure the best application field of its pros is in presence of large numbers of users, where a wrong choice about the learning path to be offered could have serious consequences. In order to support the decision making process aiming at selecting the most suitable e-learning path(s), a multi-attribute, multi-expert model has been introduced, where several attributes are used for evaluating different e-learning paths, according to the rankings expressed by a group of experts. Then, a consensus modelling

mechanism is introduced to find an agreement among the individual rankings. The multi-attribute evaluation is based on fuzzy TOPSIS while the consensual ranking is obtained through a constrained optimization model. Fuzzy logic in e-learning has been used according to different perspectives. Some fuzzy approaches to e-learning have been presented in [3], where fuzzy logic has been applied to the identification of e-learning design requirements and to select the most suitable e-learning service provider. Other approaches [4] use fuzzy inference to analyze students' way of working and group's behavior, while in other research areas fuzzy logic has been used to improve search capabilities of Learning Management Systems (LMSs) [5]. In the field of evaluation, under different perspectives the application of fuzzy logic to the evaluation of students' performances according to their profile [6], or to an evaluation teaching systems' quality [7] has been applied.

The same mechanism and the same attributes, or variations of them, can be applied to a different granularity of objects inside our platform. For example, very frequently in e-learning settings a teacher can use collaborative tools like forums or wikis to discuss over a topic. The comments of the users are often summarized or even pointed as "the best", the most representative response to the original post even coming from participants in form of a question. The provided model could be applied also inside these contexts, where a panel of experts (teachers, students or a mix of them) could evaluate the different alternatives (the different answers to a question) expressing linguistic values in correspondence of pre-defined appropriate vocabulary of linguistic labels for the attributes. In our opinion, e-learning systems (and virtual community systems) will need these extensions that go in the direction of cognitive computing, thus transforming the e-learning, passive environment (where actors simply download slide-ware) into an intelligente cognitive system able to support us in decisions related with our daily life, education included.

VI. CONCLUSIONS

The paper presented our point of view respect to the current state of evolution of LMSs, specifically their capabilities of reacting to new stimulus from end-users that require a deeper integration with the hosting information system. Our view is that customized platforms could perform largely better in these context rather than general purpose LMS. The research will be expanded with some extra comparison, but the empirical evidences collected so far seem to confirm that, when learning processes are not isolated islands inside the information system but core component of internal processes, LMSs provide a much higher rigidity and total cost of ownership. On the contrary, a customized platform, where the source code has been developed internally, could have its' Return on Investment exactly in these situations, furthermore providing extra advantages like seamless integration with the rest of the information system, greater customization capabilities and a much higher flexibility in adapting educational processes to the changing organizational needs.

Looking onwards, it rarely happens that we will witness a radical change in technology and business. It typically happens every 25 years or so, and it's happening now. What this will entail is mainly related with exponential learning, a process of exponential growth of training demand because new knowledge and skills must be delivered at a speed never seen before (see Industry 4.0 but also other community programs, cognitive managers, cognitive architecture engineers, cognitive system programming, etc.). So the paradigm should be extended, shifting from classrooms to communities, talking no longer of men *or* machines, but men *and* machines, then the technology will be an appendix extending the learning processes of individuals, enhancing their faculties and assisting them in the transformation of skills. This will happen through the definition, design and use of cognitive services that can be implemented in a platform like the one presented in this paper, that has already acquired and historicized its big data, but will have to offer a new set of cognitive services. We will be forced to respect two fundamental constraints: time and content, with contents that will have to be ready within the time learners will need them. Probably the services will be profiled for different users levels, such as learning professional and learning business consumer. We are on a turning point of training processes, a very challenging and important moment in which cognitive approaches will transform everything, and e-learning processes and platforms are not excluded.

REFERENCES

[1] Colazzo L., Molinari A., (2007) Shifting from Traditional LMSs to Virtual Community Systems. Journal of e-learning and Knowledge society, NuovaSerie, v. 3, n. 2, p. 95-104. , DOI: 10.1400/77970

[2] Molinari A. (2015) Designing Learning Objects For Italian Public Administration: A Case Study, Problems of Education in the 21st Century, 2015; 68(68) 52–63 ISSN 1822-7864

[3] Kazançoğlu, A. P. D. Y., & Aksoy, M. (2011). A Fuzzy Logic-Based Quality Function Deployment For Selection Of E-Learning Provider. TOJET, 2011, 10(4).

[4] Redondo, M.A., Bravo, C., Bravo, J., Ortega, M. (2003) Applying Fuzzy Logic to Analyze Collaborative Learning Experiences in an e-Learning Environment.USDLA Journal. (United States Distance Learning Association).17.2, 19-28

[5] Perakovic D., Grgurevic I., Remenar V. (2008), Possibility of applying fuzzy logic in the e-Learning system, In proceeding of: Information and intelligent systems CECIIS 2008 : 19th International conference, September, 24th - 26th, Varaždin, Croatia, 2008

[6] Hogo M. (2010), Evaluation of E-Learners Behaviour using Different Fuzzy Clustering Models: A Comparative Study, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010

[7] Yongqiang H.; Jianxin W., "A Study on Fuzzy Evaluation of E-learning Teaching Quality," e-Business and Information System Security (EBISS), 2010 2nd International Conference on , vol., no., pp.1,4, 22-23 May 2010 doi: 10.1109/EBISS.2010.5473769

[8] Rheingold, H. (1993). *The virtual community: Homesteading on the electronic frontier*. Reading, Mass: Addison-Wesley Pub. Co.

Where do People Look while Identifying Colors in Images

Soraia M. Alarcão Ruben Pavão Manuel J. Fonseca
LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
salarcao@lasige.di.fc.ul.pt, rubenpavao@gmail.com, mjfonseca@ciencias.ulisboa.pt

***Abstract*—In this paper, we present a study with users to identify the dominant and the search colors users associate to a set of images. We supplement this information with gaze coordinates, collected with an affordable eye tracker, to register the regions at which people looked while identifying colors in the images. The analysis of the data revealed that users use a small set of color names, and the colors used for searching were similar to those considered dominant. Moreover, there is no strong correlation between the regions to which people looked and the colors they identified. As a result of the study, we make available a dataset of 100 images annotated with their dominant colors, the colors that users would use to search for them, and the areas where they looked while identifying both types of colors.**

I. Introduction

Color is one of the most distinctive visual features. Various systems for exploring, searching and presenting images to users take advantage of it through the use of image's dominant colors. Although there are mechanisms to search for or explore images through their dominant colors, these are usually identified from the perspective of the system and making several assumptions (e.g. more importance to the center, salient objects, etc.) and not based on the human perception of colors. That is, typically the dominant color is the one that occupies the largest area of the image. However, from the point of view of people, the dominant colors are not always those that cover more pixels. Additionally, most works consider too many colors as possible dominant colors, making their naming almost impossible, when users may want/need to explore or retrieve images by specifying the colors names.

The aim of this paper is to investigate whether there is a relation between the regions at which people look and the colors they identify as dominant in an image or the colors they would use for searching the image. To that end, we collected eye tracking data in an experiment with 40 participants in our research lab. In particular, we designed two setups, one where we asked participants to identify up to three dominant colors in an image, and another where we asked them to mention up to three colors they will use to search for the presented image. Additionally, we collected eye tracking data of the regions of the image at which users looked while identifying colors.

From the data collected, we can conclude that there is no strong correlation between the regions at which people looked and the colors they identified. In most cases users looked at one region (e.g. faces) and mentioned a color that is presented in another region (e.g. t-shirt). In terms of the dominant and search colors, we found that the colors used for searching are similar to those considered dominant, which means that we can develop a retrieval system where images are described by their dominant colors. Our contributions are: 1) the confirmation of the JNS 11 colors as a valid reduced set of colors; 2) a dataset of 100 images annotated with their dominant colors and search colors identified by people; 3) gaze coordinates of users while identifying dominant and search colors in images.

DOI reference number: 10.18293/DMSVIVA2018-012

II. Background and Related Work

In this section, we provide some background about color perception, color naming, dominant colors, and the use of eye-tracking to identify where people look at in images.

A. Color Perception

Color is the perceptual phenomenon related to the spectral characteristics of the electromagnetic radiation in the visible wavelengths (approximately from 380-750 nm).

Our vision starts on the eye retina with two types of photoreceptors that receive the light stimulus. Rods are responsible to operate at low light levels (scotopic vision), while cones operate at higher light levels (photopic vision). Cones are the ones responsible for the color vision, having a high spatial acuity. These signals are then interpreted in the cortex with other visual information received simultaneously, as well as our previously accumulated visual experience (memory).

As so, we can say that color is the result of interpretation in the brain of the perception of light in the human eye and our visual memory.

B. Color Naming

In everyday life, we mainly identify colors by their names, which requires a general color vocabulary that is far from being precise. Given the importance of color naming, a variety of models and studies describing how people associate names and colors were introduced. Berlin and Kay studied the color naming behavior with subjects from multiple languages [1]. They concluded that the basic color terms in a culture can be predicted by the number of color terms the culture has. For English, they identified the following 11 basic terms: black, white, red, green, yellow, blue, brown, pink, orange, purple, and gray. Mojsilovic *et al.* presented a computational model for color categorization and naming of the 11 basic colors plus beige and olive [2].

Weijer *et al.* used real-world images to learn the 11 basic colors [3]. Moroney *et al.* conducted an unconstrained web-based study where they identified the 20 most commonly used color terms: green, blue, purple, red, pink, light, lime, dark blue, brown, yellow, black, orange, sky, bright, violet, olive, navy, sea, teal, and royal [4]. Menegaz *et al.* proposed a discrete model for color naming, where each of the 11 basic color terms were modeled as a fuzzy set [5]. Benavente *et al.* presented a parametric model for automatic color naming, where each of the 11 basic color terms were modeled as a fuzzy set with a parametric membership function [6].

As we can see, various authors adopted the set of 11 colors proposed by Berlin and Kay, probably because it is considered to contain colors that can be named by all cultures. Indeed, in 2000, Chang *et al.* coined it as the "Just Not the Same" colors (JNS), because any two colors from this set are not perceived as the same [7].

C. Dominant Colors

In general, color is a very distinctive feature, and as such several image search systems take advantage of it. In particular, they use the dominant colors of the images as a mechanism to describe and index their content. Usually, these systems rely mostly on color histograms to provide both the description of the colors present in an image and their quantities. Histograms are obtained by counting the number of pixels for each color, after quantizing the image colors into a reduced set of colors.

The VisualSEEk was one of the first systems for searching images using the dominant colors. It used the HSV color space to compute a histogram of 166 colors, from which it identified the dominant ones [8]. Deng *et al.* presented a feature descriptor that uses segmentation and color clustering to identify representative colors in each image's region [9]. Mojsilovic *et al.* proposed a method to compute dominant colors by considering both information captured through the image histogram and extracted from spatial relationships between frequently occurring colors [10].

Atsalakis *et al.* proposed the use of a neural network to automatically identify the significant colors with the minimum number of color classes [11]. Younnes *et al.* [12] and Amante *et al.* [13] proposed methods based on a fuzzy representation of colors to identify the dominant colors. Talib *et al.* proposed a method to reduce the background effect on the computation of dominant colors. Authors assigned weights to each dominant color in accordance with its belonging to the object or the background. The background colors, which are in contact with the image borders and out of salient object area, received a lower weight [14].

Although there are mechanisms for content-based image retrieval using dominant colors, most of them identify the dominant colors from the perspective of the system and not taking into consideration the human perception of colors.

D. Eye-tracking

Eye-tracking consists on cameras continuously tracking the position or orientation of the eyes [15]. Fixation consists on maintaining the visual gaze on a single location, and is useful to determine the focus of attention, i.e., to identify what triggered the attention change. Datasets of images annotated with eye-tracking information are important for the development of saliency models, i.e., to identify which information on an image attracts visual attention from the person looking at it.

In Table I, adapted from [16], we present some of the existing datasets available in the public domain (for detailed information, see [17], [16]). DUT-OMRON has only the locations of the fixations, while the GazeCom Image, MIT CSAIL, MIT LowRes, and VAIQ have only raw data. IRCCyN Image 1 and Memorability have both the locations and duration of fixations, while McGill ImgSal also have raw data. KTH has the locations and durations of fixations combined with the inter-fixation durations. Fifa, LIVE DOVES, and MIT CVCL have the locations and durations of fixations with the raw data.

Table I
DATASETS OF IMAGES ANNOTATED WITH EYE-TRACKING INFORMATION.

	Fixations		Inter-Fixation	Raw
	Locations	Durations	Durations	Data
DUT-OMRON	yes			
GazeCom Image				yes
MIT CSAIL				yes
MIT LowRes				yes
VAIQ				yes
IRCCyN Image 1	yes	yes		
Memorability	yes	yes		
McGill ImgSal	yes			yes
KTH	yes	yes	yes	
FiFA	yes	yes		yes
LIVE DOVES	yes	yes		yes
MIT CVCL	yes	yes		yes

As far as we know, all of them contain eye tracking information but none is related to the tasks of looking at images while identifying the dominant colors or the colors to be used for searching.

III. USER STUDY

In this section, we describe the study carried out to collect information about the way users identify colors in images (both for searching and as dominant), the names of colors they mention, and for what regions of the image they look while enumerating the colors.

A. Participants

Forty participants, divided into two groups of 20, completed the study. The first group (G1) was composed of 14 males and 6 females, with an average of 22 years old (SD=2.86). Six users wore glasses and one wore contact lenses. In the second group (G2) there were 12 males and 8 females, with an average of 21 years old (SD=2.96). Seven wore glasses and two contact lenses. All participants were voluntaries and had never used an eye tracker. Participants from group G1 answered question Q1 *"What are the (up to) three colors that you identify as dominant in this image?"*, while participants from group G2 responded to question Q2 *"What (up to three) colors would you use to search for this image?"*.

B. Apparatus and Material

We used a desktop computer with an application to present the images to the users and register the gaze coordinates collected by the eye tracker. We used TheEyeTribe[1] (an affordable eye tracker), placed under a 20" LCD monitor with a resolution of 1600 x 900 pixels. To collect the coordinates, we used the eye tracker API with the maximum sampling rate supported (60 Hz). Participants were placed at a distance between 50 cm to 70 cm of the monitor (and the eye tracker). All users used the same computer and eye tracker, in the same place, with the same setup.

For the study, we used a set of 100 images (all with Creative Commons licensing) collected from Flickr, and organized into 30 categories: animal, architecture, baby, beach, bird, building, car, clouds, dog, flowers, food, girl, graffiti, lake, landscape, nature, night, people, portrait, river, sea, sign, sky, snow, street, sun, sunset, trees, urban, and water. These categories were based on the ones used in the MIRFLICKR[2] dataset. We did not use this dataset because its images have a reduced size (500 x 500 pixels), which would produce poor results for the gaze coordinates.

To gather the images for our dataset, we performed an advanced search on Flickr, using the category name as tag and "Large" as the minimum size. For each category we selected four images (the first, third, fifth and seventh). After this initial step, we ended up with 120 images. From these, we discarded 20 images that were very similar to others in the dataset, thus getting 100 images. All images were resized, keeping the aspect ratio, to have their width or height equal to the width (1600) or height (900) of the screen (e.g. 1350 x 900; 669 x 900). By doing this, we had a direct correspondence between the images and the screen (and eye tracker) coordinates.

C. Research Questions

Taking into consideration the goals of our study, we identified six research questions that we wanted to answer:

- RQ_1 Can we reduce the name of all mentioned colors to a small subset (palette) of colors?
- RQ_2 Do users use the colors they consider dominant in an image to search for it?
- RQ_3 Where do people look at more often in an image while mentioning its colors?
- RQ_4 Do users look at the regions where the mentioned colors are?
- RQ_5 Does the category of the image affect the gaze pattern of the users?
- RQ_6 Does the type of color (e.g. warm, pure, etc.) influence the set of mentioned colors?

D. Procedures

The sessions took place in a room properly prepared for the study, with adequate lighting and isolation from external interferences. We started the study by showing to the users three plates (4, 7 and 17) from the Ishihara 24 plates test [18], to check for color blindness. Participants who did not pass the test were discarded.

For those who passed the test, we started by collecting demographic information about them, namely age, gender and whether they were wearing glasses or contact lenses, and calibrated the eye tracker. Then, we presented 100 images to each user, one at a time, during seven seconds. For each image users verbally enumerated the names of the colors, while our application registered the coordinates of the image at which they looked using the eye tracker.

Half of the users (G1) enumerated up to three colors that they consider to be the dominant ones, while the other half (G2) enumerated up to three colors that they would use if they wanted to search for the image. The names of the colors were not defined a priori, so users could say any name they wanted. We registered those names as users enumerated them.

IV. Results

This section presents the main results from our study and answers our research questions. Finally, we describe the resulting dataset containing the images, their dominant and search colors, and the gaze coordinates collected.

A. Color Names

After collecting the color names and the gaze coordinates for each user and image, our first step was to group the names of the colors mentioned by users, to see if we could reduce them to a small palette. We performed this separately for each group (G1 - dominant colors, G2 - search colors).

From the analysis of the names, we found that they could be grouped into a reduced number of colors. In fact, the names mentioned more often by the users were the 11 JNS colors, defined by Berlin and Kay. Table II presents the colors enumerated by the participants and how we grouped them into the 11 colors palette. As we can see, for each color of the palette, the color most mentioned was equal to that of the palette. In fact, 90.7% (dominant colors) and 94.0% (search colors) of the names mentioned by the users belonged to the 11 colors palette. These results are in line with our previous study [13], where we found an agreement of 94.6%.

From this, we can conclude that the 11 JNS color palette is appropriated for the identification of dominant colors and the specification of colors for searching. Furthermore, it contains colors whose names people can easily enumerate, enabling them to specify colors using various modalities, such as speech, writing or sketches, making the creation of queries for content-based retrieval or color exploration systems more natural, easier, and simpler to perform.

We could have used the palette introduced by Ware in the scope of an application for nominal information coding [19, p. 126], which is composed of the 11 JNS colors plus the cyan, but from our analysis people mentioned cyan a very reduced number of times (only twice for dominants and three times for search). Thus, and despite this 12 colors palette being used by Google and Bing in their image search engines, we found that the 11 JNS colors palette is more natural to users.

[1] https://theeyetribe.com/

[2] http://press.liacs.nl/mirflickr/

Table II
COLORS ENUMERATED BY THE PARTICIPANTS AND HOW WE GROUPED THEM INTO THE 11 COLORS PALETTE.

Color Palette	Dominant Colors (G1)			Search Colors (G2)		
	Total	#	Names	Total	#	Names
White	942	935	White	993	992	White
		3	Off-White		1	White Light
		3	White Light			
		1	Transparent White			
Black	520	519	Black	555	554	Black
		1	Ebon		1	Black Gray
Gray	336	320	Gray	290	284	Gray Light
		7	Light Gray		3	Gray
		5	Dark Gray		1	Grayish Brown
		1	Cement		1	Gray Cream
		1	Gray-medium		1	Silver
		1	Gray Tree			
		1	Silver			
Red	487	451	Red	507	481	Red
		14	Brick		14	Brick
		12	Bordeaux		6	Bordeaux
		4	Wine		3	Red Pink
		2	Red Pink		1	Dark Red
		1	Red Brown		1	Wine
		1	Red-sly		1	Garnet
		1	Red wine			
		1	Vermilion			
Brown	724	497	Brown	687	524	Brown
		59	Beige		51	Beige
		59	Skin color		49	Skin Color
		46	Cream		23	Cream
		25	Light Brown		20	Light Brown
		21	Dark Brown		11	Dark Brown
		2	Cream Brown		2	Beige Yellow
		2	Sepia		1	Yellowish Brown
		1	Light Beige		1	Reddish Brown
		1	Dark Beige		1	Brown Earth
		1	Camel		1	Skin Brown
		1	Yellowish Brown		1	Cream Brownish
		1	Brown Beige		1	Maroon
		1	Camel Brown		1	Honey
		1	Gray-brown			
		1	Greenish Brown			
		1	Dirty Brown			
		1	Brown Earth			
		1	Brownish Brown			
		1	Brown Tree			
		1	Creamy			
Orange	170	160	Orange	172	166	Orange
		1	Reddish orange		5	Redhead
		1	Orange Brick		1	Reddish orange
		1	Peach			
		7	Redhead			
Yellow	561	525	Yellow	582	566	Yellow
		13	Golden		6	Golden
		5	Light Yellow		5	Sand Yellow
		5	Yellow Roasted		2	Diarrhea Yellow
		4	Ocher		1	Yellow Yellow
		3	Dark Yellow		1	Sand Yellow
		3	Blond		1	Earth Yellow
		2	Yellowish			
		1	Yellow Vomit			
Green	898	836	Green	959	937	Green
		29	Dark Green		10	Dark Green
		8	Light Green		4	Light Green
		6	Lettuce Green		3	Greenish Yellow
		5	Forest Green		1	Vegetation Green
		4	Acid Green		1	Greenish Brown
		2	Greenish Yellow		1	Greenish Blue
		2	Green Petroleum		1	Grass Green
		2	Greenish		1	Aqua Green
		1	Olive Green			
		1	Greenish Blue			
		1	Lime Green			
		1	Pale Green			
Blue	803	707	Blue	853	794	Blue
		31	Dark Blue		20	Dark Blue
		27	Turquoise		16	Light Blue
		6	Aquamarine		8	Turquoise
		5	Light Blue		4	Sea Blue
		5	Indigo Blue		3	Cyan
		5	Sea Blue		1	Greyish Blue
		5	Navy Blue		1	Dark Blue Gray
		2	Cobalt Blue		1	Navy Blue
		2	Blue Baby		1	Night Blue
		2	Blue Cyan		1	Sky Blue
		2	Blue Green			
		2	Sky Blue			
		1	Greyish Blue			
		1	Blue Gray			
Purple	94	69	Purple	105	64	Purple
		14	Lilac		24	Lilac
		11	Violet		15	Violet
					2	Light Purple
Pink	116	106	Pink	136	128	Pink
		4	Magenta		5	Magenta
		1	Pink Skin Color		2	Hot Pink
		1	Pink Bordeaux		1	Pink Skin Color
		1	Light pink			
		1	Pink Fluorescent			
		1	Salmon			
		1	Fuchsia			

B. Dominant Colors vs Search Colors

One of our research questions (RQ_2) seeks to know whether the colors that people use to search for an image are related to the dominant colors of that image. To that end, we started by identifying the most voted colors for each image and for each situation (dominant and search).

We consider a color to be a dominant or search color for an image if it has more than 10% of the votes for that image. We defined this threshold based on our previous tests, where we found that a color with less than 10% has a very low importance on an image [13].

With this approach, we could assign more than the three colors that we asked users to mention, i.e., we decided not to limit the number of colors to three because: 1) some colors can have the same percentage of votes, and we should not ignore one of them just because there are more than three colors; and 2) people perceive colors differently, e.g., some shades of red can be perceived as orange or as brown. Thus, if a significant amount of people identify that in a specific image the existing reds are "brown" or "orange", this should be reflected on the colors that describe the image. As an example, consider an image that has the following distribution of votes: 35% black, 24% red, 14% white, 14% yellow, 6% blue, 4% orange, and 3% gray. The resulting set of colors will be black, red, white, and yellow, since they have more than 10% of the votes.

After assigning the most voted colors (dominant and search) to all images, we aligned the similar dominant and search colors for each image. We ended up with a set of 400 pairs, some composed of two colors that are similar on both sides (e.g. green-green) and others where we have only one color on one of the sides (e.g. green-none, or none-green). The latter means that there was no similar color on the other side.

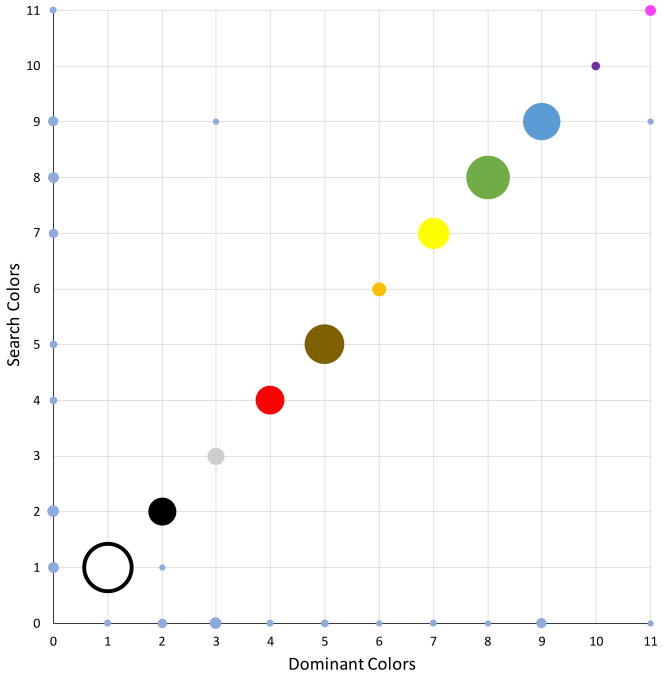


Figure 1. Distribution of the votes for all the images across the eleven colors: white (1), black (2), gray (3), red (4), brown (5), orange (6), yellow (7), green(8), blue (9), purple (10), and pink (11). Zero represents a color that was used as dominant/search but was not used for search/dominant.

Figure 1 presents the distribution of these pairs across the eleven colors. In the diagonal, we can see the colors that were used simultaneously as dominant and for search, while the size of the bubble represents the amount of times that this pair occurred. We have a correspondence of 80.5% between the dominant colors and the search colors, 0.75% where the two colors are different, 10.25% where we have a color for search but not for dominant, and 8.50% on the opposite case.

To assess the agreement between the dominant and search colors, we used similarity metrics to quantify how similar two sets of colors are (dominant colors are denoted by D, while search colors are denoted by S). The measures used were the *Jaccard index* [20] (see Eq. 1), the *Sørensen-Dice index* [21], [22] (see Eq. 2), and the *Overlap coefficient* [23] (see Eq. 3). For all these metrics, the closer its value is to one (or 100%), the more similar the two sets are.

$$jaccard(D,S) = \frac{|D \cap S|}{|D| + |S| - |D \cap S|} \quad (1)$$

$$sorensenDice(D,S) = \frac{2\,|D \cap S|}{|D| + |S|} \quad (2)$$

$$overlap(D,S) = \frac{|D \cap S|}{min\Big(|D|, |S|\Big)} \quad (3)$$

Let us consider the following example: we have an image with dominant colors white, red, and green, while the search ones are white, red, green, and blue. White, red, and green colors are common to dominant and search, but blue is not.

If we we are concerned with exact matches, we should use the $jaccard$ or $sorensenDice$ to assess the agreement. In such case, we would have an agreement of 75% for $jaccard$ and 86% for $sorensenDice$, i.e., in both cases we would be penalizing the result due to the existence of an extra color (blue). Otherwise, we should used $overlap$ that will only consider the exact matches, even if there are more colors assigned to dominant than search, or vice-versa. In this case, we would have an agreement of 100%.

Table III presents a summary of our dataset. We present the number of images per category, the average number of dominant colors and search colors assigned to each category, and the average agreement percentage for each similarity metric. As we can see, around half of the categories (53.44%) have the same average for dominant and search colors, while 33.33% have an average of search colors bigger than the dominant.

For the dominant colors, the following categories have at least an average of four dominant colors: portrait, street, baby, dog, graffiti, lake, night, river, sign, and water; while building, clouds, snow, and landscape categories have three or less dominant colors. Regarding the search colors, the following categories have at least an average of four colors: street, animal, baby, dog, lake, portrait, river, and urban, while clouds, snow, and landscape categories have three or less colors.

Table III
OVERVIEW OF OUR DATASET, SHOWING THE NUMBER OF IMAGES PER CATEGORY, THE AVERAGE NUMBER OF COLORS PER CATEGORY AND THE AVERAGE VALUES FOR EACH METRICS.

Category	#	AvgDC	AvgSC	jaccard	sorensenDice	overlap
Animal	4	3.75	4.00	72.5	82.8	85.5
Architecture	3	3.67	3.67	85.0	92.7	100
Baby	3	4.00	4.00	70.0	81.7	89.0
Beach	4	3.20	3.60	71.3	86.0	100
Bird	4	3.50	3.75	85.5	91.5	100
Building	3	3.00	3.33	75.0	84.3	89.0
Car	4	3.25	3.75	87.5	93.0	100
Clouds	2	3.00	3.00	75.0	83.5	83.5
Dog	3	4.00	4.00	86.7	92.7	100
Flowers	4	3.50	3.50	75.0	84.8	100
Food	3	3.67	3.67	85.0	91.7	100
Girl	4	3.25	3.75	88.8	93.8	100
Graffiti	4	4.00	3.50	76.3	87.5	100
Lake	3	4.00	4.00	89.0	93.3	93.3
Landscape	4	2.75	3.00	79.3	86.8	91.8
Nature	3	3.67	3.67	75.0	84.3	100
Night	3	4.00	3.67	78.3	87.0	91.7
People	4	3.50	3.50	75.0	84.8	91.8
Portrait	3	4.67	4.00	74.0	85.0	100
River	3	4.00	4.00	100	100	100
Sea	3	3.67	3.67	83.3	89.0	89.0
Sign	3	4.00	3.67	91.7	95.3	93.8
Sky	4	3.50	3.50	90.0	93.8	93.8
Snow	3	3.00	3.00	90.0	93.8	93.8
Street	3	4.33	4.33	63.3	77.0	83.3
Sun	3	3.33	3.33	83.3	90.7	100
Sunset	3	3.33	3.67	91.7	95.3	100
Trees	3	3.33	3.33	83.3	90.7	100
Urban	4	3.25	4.00	83.8	90.3	100
Water	3	4.00	3.67	93.3	96.3	100
Total	100	3.59	3.66	82.1	89.3	96.2

If we analyze our results considering the most restrictive measures, we have a $jaccard$ agreement varying from 72% to 100%, and a $sorensenDice$ agreement varying from 82.23% to 100%. The most permissive of the three measures, the $overlap$ varies from 89% to 100%. If we now consider the overall dataset, we have an average agreement of 82.12%±17.04% using $jaccard$, 89.28%±10.80% using $sorensenDice$, and 96.22%±9.99% using $overlap$.

From these values, we can conclude that there is virtually no difference for users when asked about dominant colors in an image and colors to be used for searching for that images. In conclusion, a possible algorithm that identifies dominant colors in images according to human perception, will also serve to highlight the colors that would be used by a user to search for the same image.

C. Focus Regions

Before we analyzed the gaze information, we validated for each participant if there were any corrupted data to be removed (e.g., coordinates outside the image). Across all the images and participants, we had a total of 287 538 gaze coordinates for the dominant colors and 268 716 for the search colors. We discarded around 7% of corrupted data from the former and around 11% from the latter.

To analyze and identify the gaze patterns, we created heatmaps for each image, groups of categories, and the overall dataset, considering the dominant and search colors separated. Since we have images with different orientations and sizes, we normalized the gaze coordinates for each image according to their max width and height. This way, we ensure that our conclusions are correct regardless of the orientation and size of the images. Figure 2 presents the normalized heatmaps of our dataset for both dominant and search colors. To simplify the analysis, we created groups of categories by joining related ones (e.g. animal, bird, and dog). We can see that people look at the central area of images, regardless of being questioned about dominant or search colors (Figure 2a). This is also true for the different groups of categories (Figures 2b - 2f).

Figure 3 present examples of images from our dataset with the corresponding heatmaps overlapped, and the dominant and search colors associated to each one. Figures 3a and 3f depicts a building illuminated at night. People looked more at the center of the image, where we can find the main part of the building, the lamp light and the red lights of traffic. The white, black and yellow colors reflect this gaze behavior, but black (for dominant and search) and blue (for search) are not predominant in the areas where people looked. Figures 3b and 3g depict a street with parked cars. Although, people identified white (surroundings and buildings), gray (car on front and street), red (car), and green (trees) as the dominant and search colors, in both cases, they mainly looked at the red car.

Figures 3c and 3h show a dog resting on grass. In this case, people mainly looked at the dog face and dog-collar. The predominant colors were green (grass), blue (dog-collar), and finally brown (dog body and face). It is interesting to notice that regardless of the small size of the dog-collar (when compared with the size of the dog), the blue color had more votes than the brown. Figures 3d and 3i depict a purple flower. Here, people mainly looked at the stigma of the flower (white/yellow area in the middle of the flower), the top part of the flower, and some leaves. The identified search colors were purple and green (flower), while for dominant colors, the black and blue colors were also identified.

Figures 3e and 3j show a young girl laying on the grass. We can see that people mainly looked at the girl face, but indicated white (dress), green (grass), and brown (hair and maybe skin) as the predominant colors for both search and dominant. Figures 3k and 3p depict a nightscape with buildings across the river. Similarly to Figures 3a and 3f, people mainly looked at the center of the image where the buildings and lights are concentrated. For this image, the dominant colors were gray (from the sky and maybe buildings), yellow (from the buildings lights), pink (maybe the central building resembles light pink, and the top structure at its left, dark pink), and brown (surroundings and shadows). It is interesting that in search colors, people also looked at the top of the building with a white light (right top part of image) and the building front illuminated with a white light (right middle part of image). As a result, white was one of the predominant colors identified.

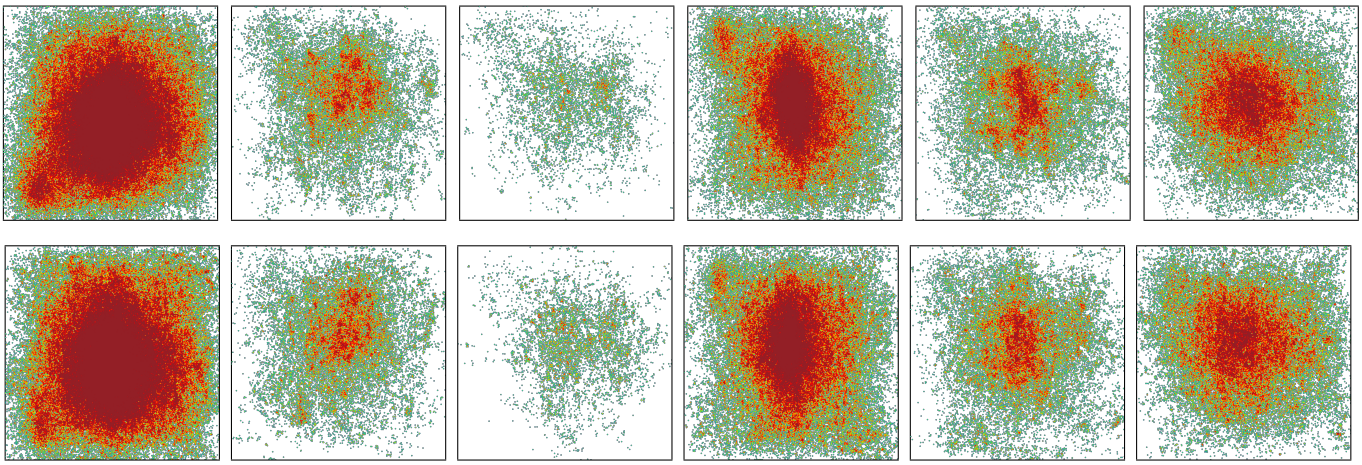
In Figures 3l and 3q, we have the face of a man surrounded by packages of chocolates. In both cases, people mainly looked at his face. However, in both cases, people identified the colors of the chocolate packages (e.g., orange, yellow, white, black). Figures 3m and 3r depict a river with some vegetation. People looked more at the top of the image, where the vegetation and the narrowest river area are. In both cases, people identified white (from the water foam), green (vegetation), blue (from the narrowest part of the river), and brown (from the banks and wider area of the river) as the predominant colors.

In Figures 3n and 3s, we have the sky with clouds. People mainly looked at the center of the image, where the biggest portion of the clouds are. Not surprisingly, the predominant colors identified were white and blue. Finally, in Figures 3o and 3t, we have a sunset on the river. People looked to the sun and the area around it. However, the most predominant color was black, where people barely looked at.

D. Discussion

Based on the results from our study, we will answer now the research questions that we raised in Section III.

According to Table II, we can say that the answer to our RQ_1 is yes, that is, we can reduce all the color names mentioned by users to a small subset of colors, such as the 11 colors palette suggested by Berlin and Kay. From the comparison and the assessment that we made on Section IV-B, we verified that there is a strong similarity between the dominant colors and search colors mentioned for each image. Thus, we can say that the colors that users would use to search for an image are the dominant colors of the image (RQ_2). Although, the gaze pattern differs a bit among the groups of categories (RQ_5), as illustrated in Figure 2, the most looked region is the center of images (RQ_3).



(a) All (b) Animal (c) Food (d) Nature (e) People (f) Urban

Figure 2. First row depicts the heatmaps for the dominant colors, and the second row for the search colors. At a given position, a darker shadow of red represents a stronger number of eye gazes, yellow and green represent a medium number, and blue a lower number. (a) all the categories; (b) animal, bird, and dog categories; (c) food category; (d) beach, clouds, flowers, landscape, lake, nature, river, sea, sky, snow, sun, sunset, trees, and water; categories; (e) baby, girl, people, and portrait categories; (f) architecture, buildings, car, graffiti, night, sign, street, and urban categories. (best seen in color)

(a) White, black, yellow (b) White, gray, red, green (c) Green, blue, brown (d) Black, green, purple, blue (e) White, green, brown

(f) White, black, yellow, blue (g) White, gray, red, green (h) Green, blue, brown (i) Green, purple (j) White, green, brown

(k) Gray, yellow, pink, brown (l) Orange, yellow, pink, brown (m) White, green, blue, brown (n) White, blue (o) Black, orange, yellow

(p) White, gray, yellow, brown (q) White, black, yellow, blue (r) White, green, blue, brown (s) White, blue (t) Black, orange, yellow

Figure 3. Examples of heatmaps and images for the dominant colors (first and third row) and search colors (second and forth row). (best seen in color)

Moreover, we noticed that people do not look at some regions of the image, but enumerate their colors, and look at other parts of the image (e.g. faces, bright spots, lights) and do not mention their colors (RQ_4). We noticed that people identify as predominant colors, colors from small areas of the image probably because they have striking colors (e.g., red car, blue dog-collar) (RQ_6).

In summary, we can say that users mentioned colors from the whole image and not only from the area where they looked at. In particular, we noticed that users focus on faces, but identify as predominant colors those of the surrounding objects (e.g. hair, clothes). This focus on faces was also observed by Cerf *et al.* in their study [24]. Finally, and although people use the same "scanning" method for the identification of the dominant and search colors, they slightly tend to disperse more their gaze while identifying colors for searching purposes.

E. Resulting Dataset

Our dataset, named UL-GDSC (Gaze on Dominant and Search Colors), is composed of 100 images collected from Flickr and resized to match the largest size of the screen (width of 1600 or height of 900 pixels). Images are organized in 30 categories, as shown in Table III, and are annotated with their dominant colors, the colors that people would use to search for them, and the coordinates where people gaze at while identifying the colors. We made UL-GDSC dataset publicly available to the community[3].

Each image has two sets of colors (dominant and search colors) based on the colors that received more than 10% of the votes. On average, images have three to four colors associated. The gaze coordinates in the dataset are the average of three consecutive raw coordinates provided by the eye tracker. Thus, we were able to have more stabilized gaze coordinates, with the cost of having less values per second, since we indirectly reduced the sampling rate (from 60 Hz to 20 Hz).

A salient aspect of the UL-GDSC is that it contains not only the colors that people identified as dominant and for searching, but also the eye movements people performed while doing it.

V. Conclusion

In this paper, we presented the results of a study with users to identify the predominant colors in images and at which they look while mentioning those colors. From the data collected, we were able to confirm that the JNS palette contains a set of colors that is representative of the color names that users mentioned. Additionally, we measured the similarity between the dominant colors associated to an image and the colors used to search for it, and found that they are very similar. So, we can use the dominant colors of the images as a content descriptor, since users would use them for searching.

The analysis of the gaze data revealed that overall there is no strong relation between the colors of the regions where people look at and the predominant colors identified in the image. Furthermore, people look mainly at the center of the image, regardless of its category.

[3]http://www.di.fc.ul.pt/~mjf/research/ul-gdsc/

Acknowledgment

This work was supported by national funds through Fundação para a Ciência e Tecnologia, under LASIGE Strategic Project - UID/CEC/00408/2013.

References

[1] B. Berlin and P. Kay, *Basic color terms : their universality and evolution*. University of California Press, 1969.

[2] A. Mojsilovic, "A computational model for color naming and describing color composition of images," *Transactions on Image Processing*, pp. 690–699, 2005.

[3] J. van de Weijer, C. Schmid, and J. Verbeek, "Learning color names from real-world images," in *Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[4] N. Moroney, "Unconstrained web-based color naming experiment," in *Color imaging VIII: Processing, hardcopy, and applications*. International Society for Optics and Photonics, 2003, pp. 36–47.

[5] G. Menegaz, A. Le Troter, J. Sequeira, and J.-M. Boi, "A discrete model for color naming," *Journal on Advances in Signal Processing*, 2007.

[6] R. Benavente, M. Vanrell, and R. Baldrich, "Parametric fuzzy sets for automatic color naming," *Journal of the Optical Society of America A*, pp. 2582–2593, 2008.

[7] E. Y. Chang, B. Li, and C. Li, "Toward perception-based image retrieval," in *Workshop on Content-based Access of Image and Video Libraries*, 2000, pp. 101–105.

[8] J. R. Smith and S.-F. Chang, "Visualseek: a fully automated content-based image query system," in *ACM international conference on Multimedia*, 1997, pp. 87–98.

[9] Y. Deng, B. Manjunath, C. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *Transactions on Image Processing*, pp. 140–147, 2001.

[10] A. Mojsilovic, H. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis," *Transactions on Image Processing*, pp. 1238–1248, 2002.

[11] A. Atsalakis and N. Papamarkos, "Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas," *Engineering Applications of Artificial Intelligence*, 2006.

[12] A. A. Younes, I. Truck, and H. Akdag, "Image retrieval using fuzzy representation of colors," *Soft Computing*, pp. 287–298, 2007.

[13] J. C. Amante and M. J. Fonseca, "Fuzzy color space segmentation to identify the same dominant colors as users," in *International Conference on Distributed Multimedia Systems*, 2012, pp. 48–53.

[14] A. Talib, M. Mahmuddin, H. Husni, and G. Loay E, "A weighted dominant color descriptor for content-based image retrieva," *Journal of Visual Communication and Image Representation*, pp. 345–360, 2013.

[15] J. Lazar, J. H. Feng, and H. Hochheiser, *Research Methods in Human-Computer Interaction*. John Wiley & Sons, 2010.

[16] S. Winkler, F. M. Savoy, and R. Subramanian, "X-eye: A reference format for eye tracking data to facilitate analyses across databases," in *Human Vision and Electronic Imaging*, 2014.

[17] S. Winkler and R. Subramanian, "Overview of eye tracking datasets," in *International Workshop on Quality of Multimedia Experience*, 2013.

[18] S. Ishihara, *Tests for color-blindness*. Handaya, Tokyo, Hongo Harukicho, 1917.

[19] C. Ware, *Information Visualization: Perception for Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[20] M. J. Zaki and W. M. Jr, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

[21] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, pp. 297–302, 1945.

[22] T. A. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content, and its application to analyses of the vegetation on {Danish} commons," *Biologiske Skrifter*, pp. 1–34, 1948.

[23] Y. A. Pesenko, "Principles and Methods of Quantitative Analysis in Faunistical Researches," *Moscow (Nauka) [in Russian]*, 1982.

[24] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in Neural Information Processing Systems*, 2008, pp. 241–248.

A multi-level approach for forecasting critical events in Smart Cities

Francesco Colace, Marco Lombardi, Francesco Pascale
DIIn
University of Salerno
Fisciano (SA), Italy
{fcolace, malombardi, fpascale}@unisa.it

Domenico Santaniello
DICIV
University of Salerno
Fisciano (SA), Italy
dsantaniello@unisa.it

***Abstract*— Nowadays, the development of new technological innovations and the concept of the Smart City leads us to new approaches to improve the quality of life. The aim of this paper is to investigate the possibility of providing a multi-level approach capable of monitoring critical events. A system based on this methodology, able to gain the behaviors of people when a critical event occurs, may be fundamental to decision-makers and the authorities for rapid response and monitoring during such scenarios. Typically, it can be possible modelling the context according to various data, such as meteorological, traffic, social ones and many others, in light of this will be introduced a methodology which tries to merge Context Aware techniques with probabilistic approach based on Bayesian Network. In this paper, the ability of this methodology, capable of providing performances comparable to standard forecasting systems, are shown by applying it to a case of study in the area of London.**

Keywords-Context Aware Computing; Smart City; Internet of Things; Big Data.

I. INTRODUCTION

Almost 54 percent (4 billion) of the earth's total population live in world's cities; by 2050, that percentage is projected to increase to 66 percent, that's what a United Nation report says. Today's mass urbanization, while an undoubted driver of growth, is also unsustainable, by putting many people at risks. In fact, while urban living continues to offer many opportunities, jobs and services, urban areas could also represent unsafe place to live. Several critical events, such as terrorist attack, a flood or a traffic accident occur frequently. Try to represent critical scenarios, in which many people are involved, could be a good starting point in order to mitigate the human life risk. However, globalization and technological development leads us toward new approaches, tools and capabilities able to help our decisions, under uncertain conditions. Recently, key advances in computing, smartphone, worldwide mobile internet access, social media and industrial big data have all contributed to break through barriers of information exchange which help disaster managers working on data-driven solutions to disaster management problems [1]. Through Big Data and Internet of Things, is possible to identify critical events, predict population involved, and provide a probabilistic prevention system in order to avoid risk scenarios [2]. Nowadays unprecedented interconnectedness and interdependencies at the local and global scale, are capable of making the risks systemic, contagious, devastating and unpredictable.

For these reasons is important to develop a methodology, based on context approach, to minimize risks may occur in cities. The aim of this paper is to show a system capable of monitoring critical events. Our system, using a top-down approach, analyzes the data collected through sensors scattered along interest areas and mine risk scenarios could be fundamental to predict the occurrence of certain events [3]. In this context, our methodology, correlating different information each other's such as weather conditions, crowding, mood of people and other factors that may influence human behaviors, seems to be capable to forecast critical situation such as traffic accidents, flooding, overcrowding, critical weather conditions, even terrorist attacks.

Different theories exist in the literature regarding to forecast risk scenarios, much of those studies pay particular attention to using contextual representation approaches [14] [15] [16]. In general, it is possible to provide efficient weather forecasts using the Bayesian Network approach. In [12], is shown a methodology for prediction of catastrophic risks such as natural disasters, in which BNs, learned to data sets, confer an advantage to support of decision making. According to some studies, is possible to forecast traffic flow using BN or using combined approach between BN and Neural Network [10]. In [5] is provided insights into the impact of weather conditions on traffic flow through correlation analysis and how critical weather condition affect the future traffic flow prediction. It has been considered deep belief networks for traffic and weather prediction and decision-level data fusion scheme to enhance prediction accuracy using weather conditions. According to [11] seems to be possible predicting a terrorist attack at critical transportation infrastructure facilities, on the other hand [13], investigate towards sentiment and social network behaviors, demonstrated that through a specific framework is possible to yield meaningful graphical visualizations of information, in order to reveal potential response to terrorist threats, in addition, is possible to build a knowledge base that would be of great utility to decision-makers and the authorities for rapid response and monitoring during such scenarios.

The proposed methodology wishes to include many aspects, frequently dealt with separately in other studies, defining a large context that takes into account the meteorological conditions, human behavior and social behavior.

DOI reference number: 10.18293/DMSVIVA2018-002

In the next paragraphs, more details about the system architecture and the application of the proposed approach in real context will be furnished..

II. Motivating Example

In this section, we want to describe how the system works and which data we can collect and use to obtain the right information to determinate the risks. Nowadays, a lot of data are exchanged through computer systems, so we have the possibility to store huge data quantities. This system needs meteorological data observed and provided, that strongly impact on the scenario; if we consider them alone they are just static information. Other key aspect is the using of human behavior data: we have tried to identify this data through travel and social network behavior. Relate the abovementioned data, that are constantly growing up, represents our real challenge.

Nowadays, is usual to have an almost constant interaction with mobile devices in order to check any information via Internet; on the other hand to be considered usable by user and it should provide a set of smart services for:

- monitoring the weather condition;
- monitoring the traffic (accidents, delays, road works) and transport (delays, disruptions) condition;
- showing the current crowding status in our interest locations;
- providing a pre-alarm system for alert the users in case of difficulty in achieving a predetermined location;
- suggesting alternatives: not only the recalculation of a route to a predetermined place but also the change of locations;
- suggestion of the proper transport (walk, car, tube, bus, cycle, hire bike, carsharing, etc.) according to weather conditions during the journey, travel times and user preferences;
- recommending special places or events deemed of interest according to user preferences;
- warning user participation in some events considered overcrowded (customized);
- predicting of the attendance degree, according to the weather forecast, to a planned event or place to be reached.

Thought these data is possible to know which is the correlation between causes and events. There are, for example, bad wheatear conditions, we know that a large amount of bike have been rent and that there is a place in the city in which there is a high percent of accident in particular days of the week: it is possible to prevent risks for the people, if there are big events (e.g. concerts) in this zone.

This system will be not only an app that suggests what to do: it will give us the chance to become part of data knowledge increasingly precise, punctual and reliable.

Therefore, this knowledge, which is part of our system, could help power users, those who work in the company, to handle emergency situations through feedback and alarms or provide suggestions to the resolution of critical situations for the liability of the environment.

III. The Proposed Approach

In a great number of cases, the problem we have to sort out is the following one: availing of a series of data, facts or observations, we are interested in identifying their most likely source, the reason which has generated them, with a view to optimizing our own decisions. Although this seems quite a simple operation, making a decision in uncertain conditions is a process which is far from being trivial.

In this respect, the goal here is to identify an architecture to be used as an extremely flexible inferential as well as decision-making tool. Such architecture does not only allow to manage complex problems, featuring a great variety of variables inter-linked through both logical-deterministic and probabilistic relationships, but also provides an effective graphic representation of the phenomenon at stake, making the problem description as well as the summary easier, enhancing the degree of comprehension and allowing to identify the key variables among those at stake.

The innovative characteristics of the proposed architecture mainly have to do with the informational content that is intended to be made available to the end users, suggesting three point of view:

- Data management and organization
- Representation of the context
- Inferential engines

A. Data management and representation

In such a context, data therefore represent the key to build up and enable services and actions to take: the goal is then to implement a Knowledge Base (KB) with a view to collecting, elaborating and managing information in real time. In this respect, by Knowledge Organization System (KOS), we mean in particular well known schemes such as Taxonomies, Theasaurus and further types of vocabulary that, together with Ontologies, constitute valid tools to shape the reality of interest into concepts and relations between concepts [9].

Many benefits stem from this: usin ontologies, for instance, allows to fix a series of key concepts and definitions relating to a given domain that can be shared, thus making the appropriate terminologies available (collaborative knowledge sharing); furthermore, an ontology allows a full re-usage of the knowledge that it codifies, even within other ontologies or rather for their completion (non-redundancy of information) and, being susceptible to interpretation by electronic calculators, enables the automatic treatment of knowledge with relevant significant advantages (Semantic Web).

B. Representation of the context

The goal is primarily to deliver to different categories of users, in a given moment, information which is useful in a given context; in practice, the objective would be to set up an

architecture characterized by a high degree of Context Awareness [17]. Real time understanding of the context where users are, via a representation by means of graphs, allows to provide a wide array of personalized, "tailored" services and hints regarding the decisions to make, that can help them in professional as well as private daily life, managing in the best possible way both the time and resources they have and showing them what is around, hence meeting their needs [8].

Context Awareness should be understood as a set of technical features capable to provide added value to services in different operational segments. Context Aware Computing applications can exploit, in this specific case, such features in order to provide context-related information to users, or suggest them an appropriate selection of actions. In order to achieve a better representation of the various features, formal tools of context representation have been adopted, capable to define in details the user's needs in the context where he is acting, through an approach « where, why, when, how ».

More in detail, the representation of the context has been implemented by means of formal models of representation, such as the Context Dimension Tree (CDT) [18].

CDT is a tree composed of a triad <r; N; A> where r indicates its root, N is the set of nodes of which it is made of and A is the set of arcs joining these nodes. CDT is used to be able to represent, in a graphic form, all possible contexts that you may have within an application. Nodes present within CDT are divided into two categories, namely dimension nodes and concept nodes. A dimension node, which is graphically represented by the color black, is a node that describes a possible dimension of the application domain; a concept node, on the other hand, is depicted by the color white and represents one of the possible values that a dimension may assume. Each node is identified through its type and a label. The children of the root node r are all dimension nodes, they are called top dimension and for each of them there may be a sub-tree. Leaf nodes, instead, must be concept nodes. A dimension node can have, as children, only concept nodes and, similarly, a concept node can have, as children, only dimension nodes.

A Context Element is defined as an assignment $dimension\_name_i = value$, while a Context is specified as an *"and"* among different context elements: several context elements, combined with each other, damage the origin of a context.

C. Inferential engines

The equations are an exception to the prescribed spec Lastly, the system, thought to be continuously functioning, collects data from various sources without interruption and immediately process them, with a view to activating precise actions, depending on the users and on the events. These latter, detected and analysed, will have to be translated into facts associated to specific semantic values: it is therefore necessary to use an inferential engine capable to draw conclusions by applying to reported facts certain rules, that could be imagined as a sequence of *if-else*. The approach selected to implement this inferential engine stems from the so called "Bayesian networks": powerful conceptual, mathematic and application tools allowing to manage complex problems with a great number of variables interlinked by means of both probabilistic and deterministic relations. Such networks also allow to update the probabilities of all variables at stake, any time that new information on some of them are collected, by using the Bayes theorem.

IV. The System Architecture

The system architecture, sketched out in the figure 1, envisages functional blocks with three main phases of functioning.

In the first phase, defined as *Collection Phase*, data, referred to as "rough data", are provided by different types of sensors. The set of data that are most significant with a view to the analysis that is meant to be carried out, is saved within a database. Then, in the *Pre-Processing Phase*, data are transformed in order to adapt them to the system that will have to use them. In general, data come indeed from different sources and therefore show inconsistencies such as, for instance, the usage of different denominations to identify the same value of a feature. In addition, this phase envisions the cleaning of the collected data, in order to eliminate any error, and the treatment of missing data. Such phase ends up with sampling and making such data discrete. Finally, the *Elaboration Phase* aims at providing a representation and interpretation of the acquired knowledge, starting from information correctly memorized, which can easily be expressed in terms of if-then constructs. To this end, an approach is followed which is based on the three views previously described, leading to implementing and using "decisional models". Such models are constantly improved based on newly collected data and experiences, or previously treated cases.

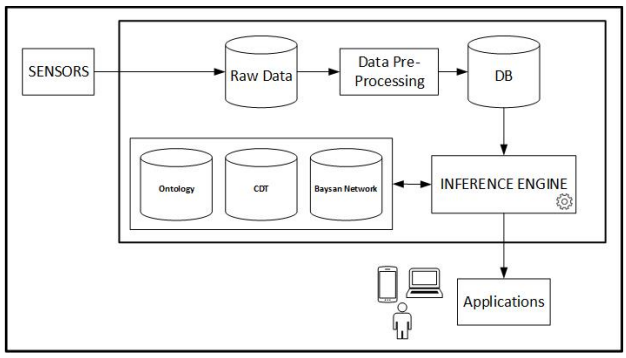


Figure 1. The system architecture

Summing up, the *need* to make a decision, in a given context, can be *met* through the fruition of the right information delivered by the architecture. This latter is featured by innovative elements based on: formal context representation, knowledge management and organisation, inferential engines.

V. Experimental Results

The system is designed to collect and analyze a vast amount of data, making it available to different categories of users. The strengths of the system would to be the ability to adapt quickly and the attempt to exploit human-machine interaction, in order to provide short time answers. The study area is the city of London where, for data availability reasons, it was possible to

collect an adequate number of data to provide a preliminary example that allow as to show the capability of the system.

The data gathered by the platform are divided in two main categories: meteorological data and data related to human behavior. One of the goals of the proposed approach is to connect these two macro "worlds". The meteorological data, acquired through various API services, include wind, temperature, visibility, UV index, precipitation, etc. Sometimes those are not uniform, available only in some areas, or presenting different time frequencies, furthermore, a preprocess and homogenization phases are essential. The second set of data is focused on the attempt to model human behavior. These data are divided into two main categories: the crowding degree and social network behavior. Through crowding degree data, the platform tries to model the physical movement of people, a large majority of these are provided by the "Open Data" service of Transport for London (www.tfl.gov.uk) which provides information on subways, buses, bike rental, service status, interruptions, accidents, etc. In addition, the other bit of data is represented by road traffic and information about social events taking place on the area of interest. Data from social networks represent the second part of the human behavior data set. The platform stores a variety of information, geolocated, on the mood of people, through API services. Selected by keywords and hashtags, thousands of posts a day are collected and analyzed either Sentiment Analysis Approach in order to extract a set of information such as "sentiment score", "sentence polarity sentiment" and "text presence sentiment", which constitute an important part of the knowledge database.

The following case of study includes all 2016-year data such as rainfall, temperature, wind speed, number of accidents and rental bicycle. The area chosen for this sample is the borough of Islington where are available each three hour aggregated data.

The proposed approach takes full advantage of three different methodologies able to represent the context through formal naming, definition of the types, properties, and interrelationships, which are:

1) Ontology
2) CDT
3) Bayesian Network

First of all, the ontological view, which represents the more general view of a context, shows the general description of text in terms of concepts and their relationships. Moreover, these description systems can infer actions from the analysis of the semantic information. In this case, reference ontologies from the published literature are used. In the figure 2, is shown an example of used ontology.

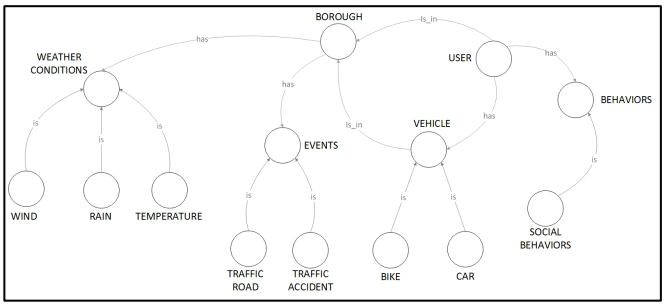


Figure 2. The ontology view

Subsequently, our methodology needs a CDT. In figure 3, it is shown a general designed CDT, so-called Meta CDT, which is the starting point for the design of a specific CDT that can be exploited in contextual applications [19] [20]. You may note six top dimensions, which correspond to the questions of the 5W1H method: Where, Who, When, Why, What and How. In particular, the CDT building methodology consists of three main phases:

- *Design phase of the Context Dimension Tree*, to identify significant context elements for the considered application.
- *Definition phase of partial views*, to find the appropriate value for a given dimension.

Composition phase of global views, to use all the information obtained in order to identify the right context and offer data customized for the user.

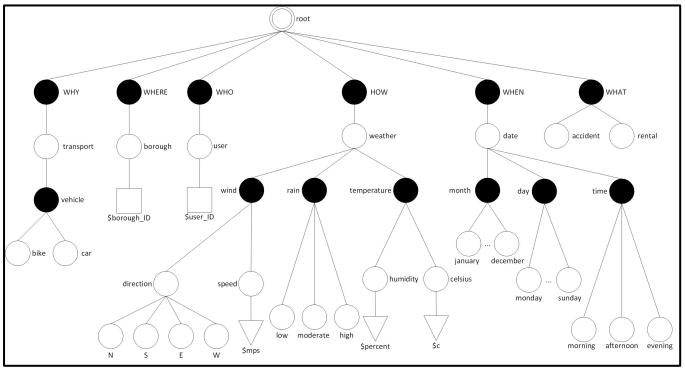


Figure 3. Meta CDT

The first two approaches provide us constraints required by the platform in order to build a reliable Bayesian Network. The so-built Network gives a description of the context in terms of its main components, their relationships are obtained by the use of the Bayesian inference. In this way, given in input a certain set of data it will be possible to predict actions and states.

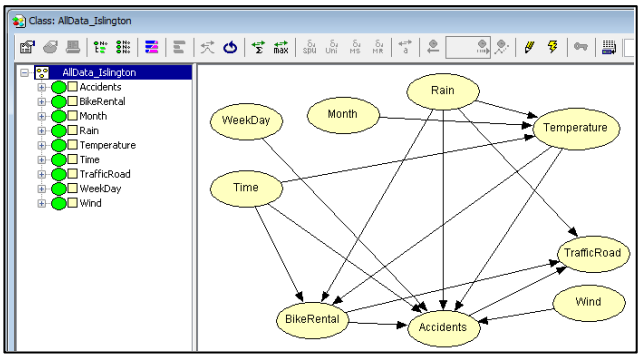


Figure 4. The Bayesian Network

In order to test the proposed approach, a set of actions has been scheduled. In particular, in the table 1, the report of our experimentations based on obtained Bayesian network is shown.

For this reason, a dataset of collected data taken in various borough of London during last year was used.

TABLE I. EXPERIMENTAL RESULTS

Parameter	*Number*	*Percent*
Correctly Classified Instances	2677	91.7197%
Incorrectly Classified Instances	243	8.2803%
Total Numer of Intances	2920	

In this first experimentation, the results are quite satisfying: about 91% the system predict correctly.

VI. CONCLUSIONS

In this paper, we have presented a system able to minimize risks may occur in cities. The approach is based on the adoption of various views that are able to shape the context and the actions to implement. The first results are quite interesting. For the future, we plan to implement a more complete experimental phase.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of D'aniello, G., Gaeta, A., Gaeta, M., Loia, V., Reformat, M.Z. Collective awareness in smart city with fuzzy cognitive maps and fuzzy sets (2016) 2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, art. no. 07737875, pp. 1554-1561 10.1109/FUZZ-IEEE.2016.7737875

[2] Chuanjie Yang, Guofeng Su, Jianguo Chen “Using Big Data to Enhance Crisis Response and Disaster Resilience for a Smart City”, 2017 IEEE 2nd International Conference on Big Data Analysis

[3] Francesco Colace, Domenico Santaniello, Mario Casillo, Fabio Clarizia “BeCAMS: a Behaviour Context Aware Monitoring System”, Conference: 2017 IEEE International Workshop on Measurements & Networking (M&N)

[4] Gaetano Fusco, Chiara Colombaroni, Luciano Comelli, Natalia Isaenko, “Short-term traffic predictions on large urban traffic networks: applications of network-based machine learning models and dynamic traffic assignment models”, 2015 Models and Technologies for Intelligent Transportation Systems (MT-ITS) 3-5. June 2015. Budapest, Hungary

[5] Arief Koesdwiady, Ridha Soua, Fakhreddine Karray, “Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 65, NO. 12, DECEMBER 2016

[6] Jieling Jin, Yuanchang Deng, “A Comparative Study on Traffic Violation Level Prediction Using Different Models”, 2017 4th International Conference on Transportation Information and Safety (ICTIS), August 8- 10, 2017, Banff, Canada

[7] Daniel Morris, Andreas Antoniades, Clive Cheong Took, “On making sense of neural networks in road analysis”, Neural Networks (IJCNN), 2017 International Joint Conference on Neural Network

[8] Giuseppe Annunziata, Francesco Colace, Massimo De Santo, Saverio Lemma, Marco Lombardi, “ApPoggiomarino: A Context Aware App for e-Citizenship.”, 2016, ICEIS (2)

[9] Mario Casillo, Francesco Colace, Massimo De Santo, Saverio Lemma, Marco Lombardi, Antonio Pietrosanto, “An ontological approach to digital storytelling”, Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016

[10] Shiliang Sun, Changshui Zhang, “A Bayesian Network Approach to Traffic Flow Forecasting”, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 7, NO. 1, MARCH 2006

[11] Manoj K. Jha, “Dynamic Bayesian Network for Predicting the Likelihood of a Terrorist Attack at Critical Transportation Infrastructure Facilities”, Journal of Infrastructure Systems Vol. 15, Issue 1 (March 2009)

[12] Lianfa Li, Jinfeng Wang, Hareton Leung, Ghengsheng Jiang, “Assessment of catastrophic risk using Bayesian network constructed from domain knowledge and spatial data.”, Risk Analysis, Vol. 30, No. 7, 2010

[13] Marc Cheong, Vincent C. S. Lee, “A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via Twitter”, Springer Science+Business Media, LLC 2010

[14] Annunziata, G., Colace, F., De Santo, M., Lemma, S. & Lombardi, M. 2016, "Appoggiomarino: A context Aware app for e-citizenship", *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems*, pp. 273.

[15] Colace, F., De Santo, M., Greco, L., Lemma, S., Lombardi, M., Moscato, V. & Picariello, A. 2014, "A context-aware framework for cultural heritage applications", *Proceedings - 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014*, pp. 469.

[16] Colace, F., Lemma, S., Lombardi, M. & Pascale, F. 2017, "A context aware approach for promoting tourism events: The case of artist's lights in Salerno", *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 752.

[17] Casillo, M., Colace, F., Pascale, F., Lemma, S. & Lombardi, M. 2017, "A Tailor made System for providing Personalized Services", *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, pp. 495.

[18] Casillo, M., Colace, F., Pascale, F., Lemma, S. & Lombardi, M. 2017, "Context-aware computing for improving the touristic experience: A pervasive app for the Amalfi coast", *2017 IEEE International Workshop on Measurement and Networking, M and N 2017 - Proceedings*.

[19] Clarizia, F., Lemma, S., Lombardi, M. & Pascale, F. 2017, An ontological digital storytelling to enrich tourist destinations and attractions with a mobile tailored story. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 10232 LNCS, 2017, Pages 567-581.

[20] Clarizia, F., Lemma, S., Lombardi, M. & Pascale, F. 2017, A mobile context-aware information system to support tourism events. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 10232 LNCS, 2017, Pages 553-566.

Supporting Living Lab with Life Cycle and Tools for Smart City Environments

Paolo Nesi, Michela Paolucci
Distributed Systems and Internet Technology Lab, DISIT Lab, University of Florence
{Paolo.nesi, michela.paolucci}@unifi.it, http://www.disit.org, https://www.km4city.org , http://www.snap4city.org

***Abstract*—Smart Cities are becoming proactive environments in which municipalities are engaging stakeholders in contributing and participating to the life of the smart city and to the development and updating of the smart solutions in the infrastructure. To this end, the methodologies and tools for Smart City Living Lab start up, management and life cycling are becoming relevant. In this document, the solution developed for Snap4City project is described. Snap4City has been developed on the basis of Km4City tools for supporting Living Lab in response to the competitive call of Select4Cities European commission project directed by three major cities in Europe: Helsinki, Antwerp and Copenhagen. The proposed Snap4City solution includes a IOT/IOE development model, a life cycle and a set of tools for data collection, data sharing, processes and analytics development, collection and management, and sharing. The paper also reports the experience of using these tools.**

Keywords— Smart City, Living Lab, Open Data, Collaborative Systems, Co-Creation Activities

I. INTRODUCTION

Smart cities are complex ecosystems in which many distinct aspects coexist, and many kinds of actors interact, such as public administration, citizens, SMEs, stakeholders, research organizations, universities. The main developing areas for a Smart City identified by the IEEE regards: economy, mobility, environment, people, living, governance. In this context, Smart City applications need to support multiple paradigms as data driven, stream and batch processing, to promote new working paradigms to enhance the collaboration among all the actors involved. One of the most diffused ways to apply this paradigm is set up a Living Lab (LL) as a place/methodology to cope with the city evolution in terms of services and city users' needs and capabilities. Living Labs are instruments where develop and implement technology to accelerate innovations cities [1]. Many studies have been made on Living Labs, for example two different approaches have been identified that can be adopted to realize them: top down and bottom up. The first approach provides technologically deterministic ideas, such as the use of: smart control room, dashboards, centralized architecture to manage the city, ICT-based overview of the activities of the citizens and it is realized in collaboration with stakeholders and companies. The bottom-up approach is a more experimental view point, it is taking place in the last years and it is centered on the idea that the innovation comes from the citizens (e.g., guerrilla bike), so that the main feature is the interaction among the final users [1], [2]. A Living Lab is also seen as a starting point to collaborate and generate models to create Smart Cities [3], as a way to develop collaborative systems capable to engage the community (students, lecturers, computer scientists, electronics engineers, politician, tourists, etc.) [4]. Living Labs are also an instrument to go toward an open innovation business model in which play a fundamental role, aspects such as: co-creation activities, sustainability, multi-disciplinarity, collaborative networked development approach, self-organizing, collaboration of various stakeholders, etc. [5]. Useful activities to promote the collaboration among the different actors involved in a Living Lab, can be: i) organization of events involving the citizens; ii) improvement of co-creation activities, or hackathon to produce new useful services [6]. Over the course of time, Living Labs evolved and moved from the older Three Helix (TH) model to the Quadruple Helix (QH) model. The TH proposes that the three major parties in innovation are industry (wealth generation), universities (novelty production) and public control (government), while the QH poses the attention on the users, identifying them as the *fourth helix*. The QH classifies the different kind of users as capable to transform the classic innovation actions into user-centric processes: i) enabler (financier and provider of the infrastructure); ii) decision maker (city guidelines, R&I programs supporting user innovation activities); iii) supporter (who promotes the users actions and activities); iv) utilizer (who uses the services produced); Developer; Marketer; Quality controller, [7], [8]. The Living Labs are obviously connected with the concept of Open Innovation, they exploit the possibility of bringing together people with different skills, experiences, roles, expertise, motivations, etc. and many kind of organizations (universities, public administrations, SMEs, stakeholders, industries, etc.) to collaborate and realize useful service considering territorial aspect, sustainability, policies adopted by the cities, [6]. In the Living Lab it is possible to test some technologies and new paradigms to foster the innovation, to shape the applications and services being developed for their citizens, at both micro- and macro-levels. In the living Lab users are not treated only as objects in the innovation process or as mere customers but also as early stage contributors and innovators [9], [5]. In Europe Living Labs are taking place in the Smart Cities and are increasingly adopted as a new paradigm to accelerate innovation actions: the European Network of Living Lab (EUoLL) is a valid reference to be considered to realize a successful LL model. EUoLL recognizes almost 400 Living Labs in the EUoLL present in its network [10].

The main technical issues regarding smart city solutions are related to manage data, consequently they have to solve problems, such as: data access, aggregation, reasoning, access and delivering services via Smart City APIs [11]. The final aim is serving city users in a smarter and more efficient manner, stimulating their participation to the city strategies and

DOI reference number: 10.18293/DMSVIVA2018-019

collaborating with all the actors involved. Therefore, collected and produced data are used to facilitate the creation of smart and effective services exploiting city data and information. Specific end-users' smart services should be developed and managed by enterprises and city operators, rather than by the municipality. On the other hand, the municipality has to provide a flexible data access and services. This means to make effective and efficient the data access with their semantics, the service delivering, the access to define and control dashboards, and the interoperability with any other smart control systems active in the city (e.g., mobility, energy, telecommunication, fire brigade, security, etc.). In the world, municipalities/cities and public administrations are publishing huge amount of open data. These data can be coarsely aggregated for integration by using solutions such as CKAN [12], OpenDataSoft [13], ArcGIS and OpenData [14]. In most cases, these solutions for open data are suitable for collecting open data files and make their indexing on the basis of corresponding descriptive metadata. Open data, in those cases, can be uploaded by providing files in different formats: CSV, XLS, XML, SHP, etc. In some cases, they provide access to effective datasets, by using some data integration and visualization tools which provide the possibility of creating graphic charts, such as distributions or pies, on the basis of the values contained in the dataset. In the extreme case, they also provide access to datasets as Linked Data (LD), Linked Open Data (LOD), coding data information in terms of RDF triples [15], [16]. Very rarely, they can provide data from some RDF store endpoints to make SPARQL queries on the data exploiting some ontology and other entities [17], rather than working only on metadata. The access to RDF stores for data browsing can be performed by using visual browsers as in [18]. In the case of directly accessible LOD, we are in presence of the so called 5 stars' open data [15]. On the other hand, in most cases the integrated LOD are not supported by multi-domain ontologies. We could state that 6 stars data would also provide a data access and SPARQL queries exploiting a semantic ontology for the integrated data model and data inference [19]. Real-time data are provided by city operators through some APIs as Web Services or REST calls. The APIs for providing data to the data aggregator of the city may be compliant with multiple standards (such as DATEX II for mobility, intelligent transport system [20]) for public services, parking; IETF [21], ETSI [22] or OneM2M [23] for Internet of Things (IOT), Green Button Connect [24] for energy data collection. However, some of the peripheral data kinds collected are not supported by any standard, thus custom solutions are adopted, such as the status of hospitals' emergency units (triage), the status of earthquakes in the regional area, etc. The effective deploy of smart services for city users is very frequently viable only by exploiting the semantic integration of data as: open data, private data and real time data coming from administrations and different city operators. This implies specific processes of reconciliation and the adoption of unifying data models and ontologies as in Km4City multi-ontology [25]. The semantic aggregation of data coming from several domains is unfeasible without a common ontology, since data are produced by different institutions/companies, by using: different formats and aims, different references to geographical elements, and different standards for naming and identification adopted in different moments [26]. Thus, datasets are rarely semantically interoperable each other since have been produced in different time, by different systems, by different people, etc. In addition, they may present different licensing models: some of them can be open, while other may be private of some city operator that would not be interested to lose the ownership by releasing them into an unregulated environment, or could simply provide some restrictions (e.g., no commercial); see for example the data of car sharing companies that are typically private of the company. For open data, as well as for private data, several different licensing models can be adopted [26], [27] enabling or preventing some business models, or simply their usage [28]. Therefore, well aggregated and re-conciliated data for the identification of services and locations (open and private) can be exploited by reasoning algorithms for enabling sophisticated service delivering. For example, by providing suggestions and hints on rout planning, inter-modality routing, parking, hospital finding in the case of emergence, finding specific point of interests, setting predictions (for parking and traffic) and detecting anomalies for early warning.

In order to create an efficient Living Lab, the following key principles have to be taken into account: i) Value (create value for users and customers as a key aspect for business success: involve SMEs, mitigate the competition, open new markets); ii) Influence (users seen as active, competent partners and domain experts, concretize new services coming from the citizens ideas); iii) Sustainability (e.g. choose right materials, implementing user-friendly approaches, considering the social and economic impact of the innovation), iv) Openess (open collaboration between people with different expertise and backgrounds. Different perspectives could lead to a successful innovation process), v) Realism (the innovation actions are carried out in the real-life, realistic and natural settings, this to increase understanding on how innovation can bring advantages valid in the real market) [8]. Another fundamental aspect is also connected to territorial needs, this is the reason why sometimes it is stated about the Urban Living Labs, that "*are environments in which innovation is spatialized, i.e., it is generated within a specific spatial environment. These are environments in which the openness of innovation manages to transcend the organizational infrastructures that are traditionally operating in the city and to invent new institutional figures for, or ways of, dialoguing between citizens and institutions*" [29], [30], [31].

In the context of Smart City, also supported by Living Labs, a set of key features for an infrastructure capable to manage all the complex aspects described above, have been identified. A Smart City Living Lab infrastructure have to be: i) able to activate practice-based knowledge production in collective and private environments (e.g., taking into account both Private and Open Data and contexts); ii) able to learn internally but also externally, by experimenting with other cities that share the same problems; iii) aware of new civic engagement models being experimented all over the world; iv) aware of the growing demand for new citizenship models; v) capable of directing investments toward opportunity creation (i.e., experiments) rather than pre-developed solutions, [32]; vi) stimulating the participation of all stakeholders in the activities of data collection and process/solution production.

Those types of infrastructures manage a massive set of data (Big Data context) that can be shared, processed, used to generate new knowledge; moreover, are capable to ingest both Open and Personal/Private Data (e.g. the user's profiles and actions done in the city). All the processes realized by this kind of infrastructures imply the necessity to pose attention on a relevant aspect: the treatment of the privacy rights on data (especially the sensitive ones), underlined also by the General Data Protection Regulation (GDPR), [33], [34], [35], [36].

In this paper we propose the Snap4City solution as a Smart City infrastructure, responding to the key features/requirements above described, and enabling the Living Lab paradigm especially in the acquisition of City Data and developing smart city solutions. The paper is structured as follows. In section II, the Snap4City Life Cycle and Architecture are described. Section III presents Datagate as collaborative tool to upload Open Data in the Snap4City Knowledge Base. Section IV contains the description of the Process Loader as a tool to publish, share and launch data Extraction/Transforming/Load processes. In Section V, the Results obtained in terms of Data acquisition, thanks to the collaborative work done, are reported. Section VI describes the conclusions and the future work.

II. ARCHITECTURE

Smart Cities need to set up a flexible Living Lab to cope with the city evolution in terms of services, city users' needs and capabilities. To this end, the Snap4City solution provides a set of tools and a flexible method and solution to quickly create a large range of smart city applications exploiting heterogeneous data and stakeholder services also enabled by Internet of Thing (IoT)/Internet of Everting (IoE) technologies and Big Data analytic. The Snap4City solution and all the innovation activities carried out for its development, have been realized involving different kind of Organizations (Universities, SME and Large Industries, Public Administrations) and users (City Operators, Resource Operators, Inhouse companies, Tech providers, Category Associations, Corporations, Research groups, Strat-ups, Early Adopters, large industries, advertisers, City users, Community builders, etc.), thus reflecting the features, described in Quadruple Helix (QH), [7], to facilitate the Living Lab approach in a Smart City, Fig. 1. The innovative aspects of the solution proposed are related to semantic computing of entities for discovering and search information, resources management, parallel and distributed computing and cloud management, applications based on microservices and external services, dashboard and development tool kits, etc. The proposed solution is flexible enough to support extensions at distinct levels of granularity: data, analytics, tools and applications.

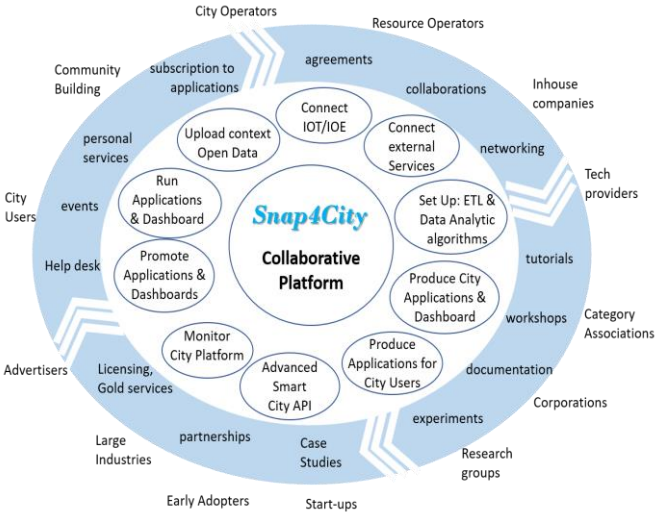


Fig. 1. Snap4City Life Cycle.

One of the first activities for creating a Living Lab in a city is the process of setting up the technical infrastructure which in turn is grounded on many valuable enabling tools. They must support the city in: the modeling of data; the upload of context data and open data; the connection of IoT/IoE sources and external services; the creation of Extract Transform and Load (ETL) processes and data analytics algorithms, to arrive at producing smart city dashboards and at starting the production of Snap4City Applications based on Microservices. All these phases must be accompanied and supported by the availability of a set of development tools, easy to use, accessible and open. To this aim, the Snap4City solution has been designed to create a collaborative environment in which different kinds of stakeholders can mutually collaborate. At the same time in which the setup is created, the collaboration among stakeholders can start by creating: agreements, collaborations, networking, producing tutorials, workshops, hackathon, etc. Fig. 1, so at to arrive at involving the stakeholder around case studies, and finally to sign contracts of partnership, licensing, etc. Thus, the delivering of specific solutions to city users, operators, etc., is becoming possible. This process must be driven by the municipality and, on the other hand, the municipality needs support for technical aspects if it is not very large and technological oriented. Typically, the single companies even if participated by the city or the city operators, do not have the view and the mission to put in common a so large multi-domain multiservice framework and environment.

From the technological point of view, the above process can be released using a set of tools, to provide a support for collaboration and sharing at distinct levels. We propose the Snap4City architecture (see Fig. 2) as capable to solve all the problems described above and which consists in:

- A layer to **ingest all the different kind of data coming from a smart city** that can be classified in: Open Data, Personal Data, IoT and IoE, Social Media, static and real time. The set of data regard many distinct categories such as: transport systems, Mobility, Car park; Public Services, Security, Museums; Sensors, Cameras, IoT, IoE; events; Environment, Water, Energy; Shops, Services, operators;

Social Media, Wi-Fi, Networks, etc. The flow of data coming from the city data can be: static, slow, real time.

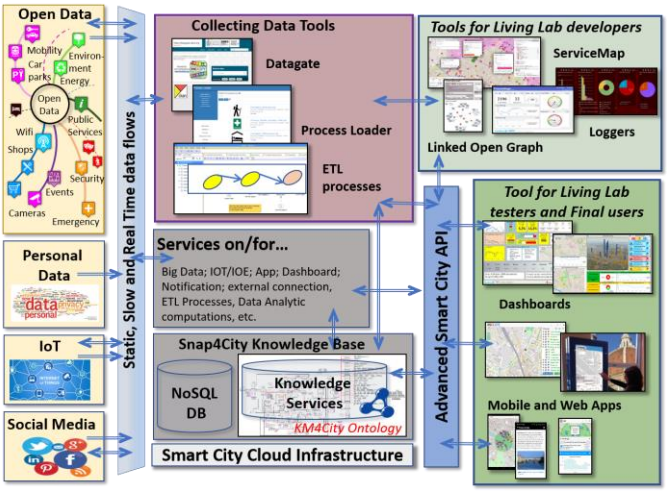


Fig. 2. Snap4City Architecture.

- **Collecting Data tools.** These tools are both for developers (ETL processes) and for users with no technical skills, and were developed precisely to exploit the possibility of bringing together people with different skills, experiences, roles, expertise, motivations, etc. and many kinds of organizations such as universities, public administrations, SMEs, stakeholders, industries, etc.
  - **ETL processes**, for developers and based on the Pentaho Kettle Tool, [38]. The Extract Transform and Load (ETL) processes can be personalized to manage many different kinds of data: open and personal, static and real time or periodic, geo-localized. The ETLs consider that data can come from: any kind of sources and providers, IoT/IoE or Sensor Networks, city users' devices, social media, open street map and in different protocols and formats. Their final aim is transform data so that it conforms to km4city multi-ontology and load them in the Sna4City Knowledge Base (in the form of RDF Store for e the static data, and in an HBase - NoSQL- store for the real time data, [39]).
  - **ProcessLoader** and Scheduler has been developed to manage processes to be executed in specific scheduling applications (for example ETL processes analyzing real time or periodic data which need to be launched every day/Hour/minute, as well as Data Analytics processes in R Studio, Java, Python, etc.).
  - **Datagate**: is a web-based open source management system for the storage, distribution, qualification, reconciliation and aggregation of Open Data. It is an extension of the very diffused Open Data platform CKAN (Comprehensive Knowledge Archive Network), [12].
- **Services for**:
  - **executing data analytics and computations** that can exploit data to provide advanced smart services one demand, early warning, both periodically and in real time modality.
  - **creating applications** that can be: data driven and/or periodic, based on Micro Services. These Services can be applications running on the platform itself (for example by using NodeRED or Pentaho [40], [38]), such as Dashboards and Mobile or Web applications.
- **Data storage layer,** collecting data in a **Knowledge Base (KB) connected to the Km4City Multi-Ontology** and making data indexing to prepare services on the data themselves, such as: data retrieval with the capability of inference and reasoning, search and retrieval, etc. [25]. While the real time Data are collected in a NoSQL database (HBase, [39]).
- **Advanced Smart City API,** capable to provide access to Snap4City data and services. The APIs can be exploited by web and mobile applications, as well as by many tools and cities, [11], [41].
- A set of **Tools for all the Living Lab Actors**, useful to test the effectiveness utility of the Snap4City solution in different contexts:
  - **Tools for Living Lab developers,** such as the web applications *Service Map* and *Linked Open Graph* for navigating on data results considering both the geographical metadata and the semantics aspects [42].
  - **Tools for Living Lab testers and final users** (community builders, city users, advertisers, Category Associations, etc. Fig. 1) showing and via dashboards and in turn make easy the production of specific dashboards for decision makers, city operators, etc. [41], [42]

In the following paragraphs, *Datagate* and *ProcessLoader*, as tools relevant to enable the Living Lab activities in a Smart City, will be described.

III. DATAGATE

Datagate (https://datagate.snap4city.org/ , Fig. 3) has been designed with the aim to offer a collaborative environment enabling the data providers to archive, manage, share datasets. It primarily manages static (or periodic) data. Moreover, it allows the data providers to upload their content, with the following features and advantages, which go behind the simple i) data storage on the portal: ii) data enrichment and geo-localization (with the city street graph or directly with Open Street Map); iii) data aggregation, with all the other data contained in the Datagate portal and with those archived in the Snap4City Knowledge Base, thanks to the semantic aggregation made through the KM4City multi-ontology; iv) data sharing (with the IPR license chosen); v) data visualization using different tools (Service Map, Dashboards, web and Mobile Apps, etc., Fig. 2), Fig. 3. These advantages can be attained in an automatic or semi-automatic modality and are described here after.

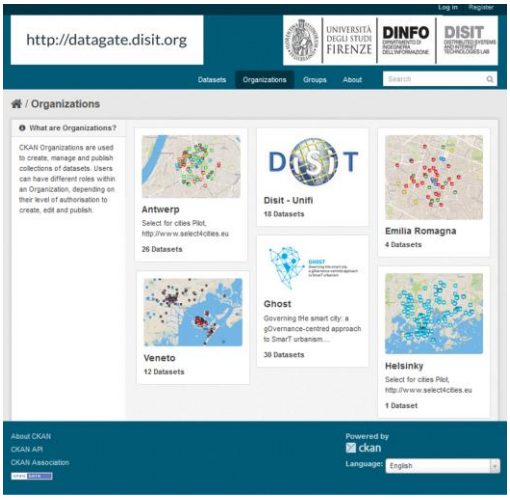


Fig. 3. Datagate: Organizations.

Data storage and sharing. These features are directly derived from the CKAN standard Open Data portal. Each data provider can be registered on the web portal and upload its datasets. A set of instruments to visualize the data are offered by CKAN (views, statistics on downloads, etc.), [12]. It is what can be viewed in Fig. 3: a set of datasets have been uploaded by their providers and then published on the web and visible to all the public Datagate users.

Data enrichment and geo-localization. This, and the following features have been added by the *'DataEnhancer'* plugin developed in the Snap4City context. If a dataset is uploaded following the specific template, the '*DataEnhancer'* features will be available on it, in addition to all default services offered by CKAN. The template is written in the form of a csv file and provides a set of fields (both mandatory and optional) to be filled to have the most possible advantages: i) mandatory fields: data name, geometry following the Well-known text (WKT) geometric objects: points (`POINT`), lines (`LINESTRING`) and areas (`POLYGON`) or address (city, street, civic number, etc.); ii) optional fields (description, web page, phone number, links, e-mail, etc.). Some fields are automatically enriched by Datagate (e.g. it can automatically calculate: postal code from address, latitude and longitude from address with civic number, moreover it reports incorrectly formed fields such as: e-mail, web portals, links, etc.), Fig. 4.

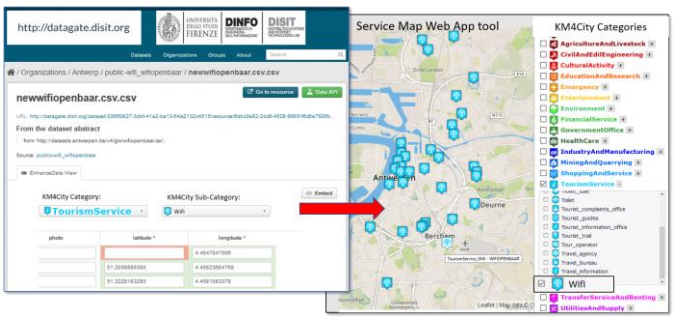


Fig. 4. Datatagate: autocompletion and error reporting. From Datagate to ServiceMap.

Data aggregation and visualization. Once the datasets are uploaded in the Datagate Portal, their responsible can connect the data to the KM4City multi-ontology by selecting one of the categories and subcategories present in the ontology and accessible from Datagate thanks to the possibility of selecting 'KM4City Categories' and the 'KM4City Sub-Categories' from a drop-down menu, Fig. 4. Click the publish on Snap4City button and upload the data also in the Snap4City Knowledge Base. In this way the data will be visible

IV. PROCESS LOADER

Algorithm/Process Loader is a web application, developed for allowing the creation and management of processes to be executed in specific scheduling applications with a user interface that receives input data in the form of compressed files that are analyzed, archived and finally transmitted to the desired scheduler. The main application's activities are focused on uploading many compressed zip archives containing files and directories required to create and execute a process on an external scheduling application through a series of API requests sent to the application. The processes cab be realized by developers coming from different context and smart cities and realized using different kind of technologies (ETL processes, R, Dashboards, nodeRED, R, etc.).

The user interface provides the following services:

- Ingestion of processes (in the form of compressed files) by authorized users. Each process is analyzed, archived and finally transmitted to the desired scheduler and properly launched.
- Process Execution: once the authorized users have uploaded their processes, they can launch and execute them, thanks to the presence of a scheduler, Fig. 5. The Process Loader users have only to insert some mandatory metadata such as the frequency with which a process has to be launched and other parameters that can depend on each process features (e.g. the web server from which the data are taken, the smart city related to the data, etc.). For example, an ETL process that links to an Open Data portal providing busses' timetables in a certain city, updated once a month. The user can create an ETL capable of collecting data and transform them as he or she sees fit (for example, to insert them in the Knowledge Base of Snap4city in order to take advantage of all the services offered by this solution). Then he/she can upload the ETL process in the process Loader, set the ETL to run periodically (e.g. once a month). In this way he/she can always have (for example on one of the services offered by Snap4City such as ServiceMap) the updated data as results without having to do any further work.
- Archiving and indexing of all the processes, that can be shared and easily retrieved, downloaded, re-used (basing on the license associated to each of them), ranked. All the metadata related to the processes are indexed via Apache Solr [44]. Each process publisher can make public its processes so that they can be shared. A web page (Fig. 6) offers the list of the public processes and an easy form to

search the ETLs basing on the metadata associated to them.

Fig. 5. Process Loader: processes running in the scheduler.

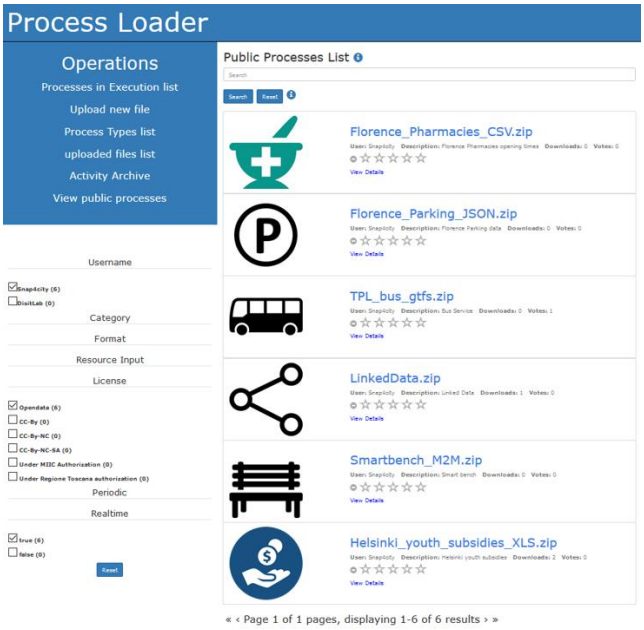


Fig. 6. Process Loader faceted search.

V. EXPERIMENTAL RESULTS

In the following tables, the main data related to the DataGate and ProcessLoader activities in terms of results obtained and services offered to the citizens are reported. These results can be navigated thanks to the ServiceMap on using all the other Snap4city tools both for Final users and for developers descripted in Section II.

DataGate	
Organizations	10
Data providers	31
Datasets number	167
Smart cities (medium and small)	Total number: more than 20 • Italy: o Medium: Florence, Venice, Bologna, Cagliari o Small: Sassari, Pisa, Lucca, Arezzo, Livorno, Pistoia, Siena, Prato, Grosseto, Massa-Carrara, Nuoro, Oristano, Sud Sardegna, etc. • Europe: Helsinki and Antwerp (medium)
ServiceMap Web Portals	• Tuscany: http://snap4city.km4city.org/ServiceMap/ • Italy (all other regions, without Tuscany) http://www.disit.org/smosm/ • Europe (mainly Helsinki and Antwerp): http://antwerp.km4city.org/ServiceMap/
Triples produced	Total: 84412 (62836 in Italy, 21576 in Europe: Helsinki and Antwerp)
Arguments treated	• cultural activities (e.g. libraries, churches, museums, theaters, monuments, etc.), hospitals, Wi-Fi, entertainments (e.g. beaches, cinemas, gymnasium), accommodations (e.g. B&B, hotels), limited traffic zones, cycling paths, etc.
Services/ point of interest in terms of PIN visible in the service map tool	• Florence: 365 • Italy: (total number: 5169) o Emilia Romagna: 367 o Veneto: 769 o Sardegna: 4033 • Europe (total number: 1749) o Antwerp: 1547 o Helsinki: 202

Table 1. Datagate numbers.

ProcessLoader:	
Processes uploaded	32
Processes published and shared	47
Processes running	10
Users	2

Table 2. ProcessLoader numbers.

Note that the numbers reported are only related to the data obtained thank to the tool presented. The Snap4City Knowledge base contains a huge amount of data that are considerably increasing in number and in terms of kind of users, different smart cities, organizations involved, thanks to the collaborative tool described.

VI. CONCLUSIONS

Methodologies and tools for Smart City Living Lab start up, management and life cycling are becoming relevant. In this document, the solution developed for Snap4City project has been described. Snap4City has been developed on the basis of Km4City ontology and tools with the aim of adding solution for supporting Living Lab in response to the competitive call of Select4Cities European commission project directed by three major cities in Europe: Helsinki, Antwerp and Copenhagen.

The proposed Snap4City solution included: (i) development model suitable for IOT/IOE applications; (ii) life cycle presented in Figure 1; (ii) a set of tools for data collection, data sharing, processes and analytics development, collection and management, and sharing as presented in this paper as DataGate and ProcessLoader. The paper also reported the experience of using these tools. As a conclusion, in order to create an efficient Living Lab, in the following table a summary of the relations among the key principles described in [10] and the Snap4City services and tool are presented.

Key principles for a Living Lab	**Snap4City solution**
Value	The data managed, and the services available in the Snap4City solution, come from many different providers: municipalities, SMEs, research centers, etc. enabling the connection among users and stakeholders considering the business aspects as playing a fundamental role for the solution success.
Influence	Many of the services proposed put the users as key actors to: give suggestions, upload data in the system, share data and opinions, etc. Moreover thanks to the Datagate and to the Process Loader tools, it is possible to: i) see the activities realized in other city; ii) to apply a set of a results in a city a starting point to realize the same services in other smart cities (e.g. use data coming from a city as a proof of concept in another one); iii) take inspiration from has already be done, to concretize new services directly coming from the citizens ideas.
Sustainability	Snap4City has been produced as an Open Architecture (also open source) that is applied in some Italian (e.g., Tuscany, Sardinia, Veneto, Emilia Romagna) and European Region and smart cities (e.g., Helsinki, Antwerp, Copenhagen). It can be reused without additional costs in every smart city context, thanks to its versatility and openess.
Openness	Thanks to several types of services offered, the work done on the Snap4City solution has involved, and continues to involve, many people with different skills and who play a different role in society enabling the interaction among different perspectives and needs.
Realism	All the data managed, and the services offered (previsions, dashboards, etc.) are made in a real context and are directly used by citizens, stakeholders, public administrations. This increase the possibility to study both innovations or advantages than cab be useful in the real market.

Tab. 1. Key principles for a Living Lab & Snap4City solution.

ACKNOWLEDGMENT

The authors would like to thank to the European Commission for founding. All slides reporting logo of Snap4City http://www.snap4city.org of Select4Cities H2020 are representing tools and research founded by European Commission for the Select4Cities project. Select4Cities has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 688196)

REFERENCES

[1] N. Villanueva-Rosales, L. Garnica-Chavira, V. M. Larios, L. Gómez and E. Aceves, "Semantic-enhanced living labs for better interoperability of smart cities solutions," 2016 IEEE International Smart Cities Conference (ISC2), Trento, 2016, pp. 1-2. doi: 10.1109/ISC2.2016.7580775

[2] Coenen, Tanguy & van der Graaf, Shenja & Walravens, Nils. (2014). Firing Up the City – A Smart City Living Lab Methodology. Interdisciplinary studies journal. Vol.3. January 2014.

[3] Ellie Cosgrave, Kate Arbuthnot, Theo Tryfonas. Living Labs, Innovation Districts and Information Marketplaces: A Systems Approach for Smart Cities. Procedia Computer Science, Volume 16, 2013, Pages 668-677.

[4] Majeed A., Bhana R., Haq A.U., Shah H., Williams ML., Till A. (2017) Living Labs (LILA): An Innovative Paradigm for Community Development - Project of "XploR" Cane for the Blind. In: Benlamri R., Sparer M. (eds) Leadership, Innovation and Entrepreneurship as Driving Forces of the Global Economy. Springer Proceedings in Business and Economics. Springer, Cham.

[5] Concilio G. (2016) Urban Living Labs: Opportunities in and for Planning. In: Concilio G., Rizzo F. (eds) Human Smart Cities. Urban and Landscape Perspectives. Springer, Cham. DOI https://doi.org/10.1007/978-3-319-33024-2_2.

[6] Maya Alba, Manuel Avalos, Carlos Guzmán, Victor M. Larios. Synergy Between Smart Cities' Hackathons and Living Labs as a Vehicle for Accelerating Tangible Innovations on Cities. 2016 IEEE International Smart Cities Conference (ISC2). 12-15 Spet. 2016.

[7] Arnkil, R., Järvensivu, A., Koski, P., & Piirainen, T. (2010). Exploring the Quadruple Helix. Report of Quadruple Helix Research for the CLIQ Project. Tampere.

[8] Anna Ståhlbröst and Marita Holst, Social Informatics at Luleå University of Technology and CDT – Centre for Distance-spanning Technology, Sweden. The Living Lab Methodology Hand book.

[9] Paskaleva, Krassimira & Cooper, Ian & Linde, Per & Peterson, Bo & Gotz, Christina. (2015). Stakeholder Engagement in the Smart City: Making Living Labs Work. 115-145. 10.1007/978-3-319-03167-5.

[10] EnoLL: http://www.openlivinglabs.eu/node/1429

[11] C.Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci. Analysis and assessment of a knowledge based smart city architecture providing service APIs. Future Generation Computer Systems 75 (2017) 14–29.

[12] CKAN: http://ckan.org.

[13] OpenDataSoft: https://www.opendatasoft.com

[14] ArcGIS OpenData: http://opendata.arcgis.com

[15] 5 Stars Open Data from Tim Barneers Lee. http://www.slideshare.net/TheODINC/tim-bernerslees-5star-open-data-scheme

[16] RDF https://www.w3.org/RDF/

[17] SPARQL: https://www.w3.org/TR/rdf-sparql-query

[18] P. Bellini, P. Nesi, A. Venturi, Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views, Int. J. Visual

Lang. Comput. (2014) http://dx.doi.org/10.1016/j.jvlc.2014.10.003, http://log.disit.org

[19] C. Badii, P. Bellini, D. Cenni, G. Martelli, P. Nesi, M. Paolucci, Km4City Smart City API: an integrated support for mobility services, in: (SMARTCOMP) IEEE International Conference on Smart Computing, IEEE, 2016.

[20] DATEXII: http://www.datex2.eu/sites/www.datex2.eu/files/Datex_Brochure_2011.pdf.

[21] IETF: https://www.ietf.org.

[22] F.J. Lin, Y. Ren, E. Cerritos, A feasibility study on developing IoT/M2M applications over ETSI M2M architecture, in: 2013 International Conference on Parallel and Distributed Systems, ICPADS, IEEE, 2013.

[23] J. Swetina, et al., Toward a standardized common M2M service layer platform: Introduction to oneM2M, IEEE Wirel. Commun. 21 (3) (2014) 20–26.

[24] Green Button Connect: http://www.greenbuttonconnect.com

[25] P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, 2014, http://dx.doi.org/10.1016/j.jvlc.2014.10.023, http://www.sciencedirect.com/science

[26] N. Korn, C. Oppenheim, Licensing Open Data: A Practical Guide. In: Discovery [online]. June 2011 [cit. 2012-02-20]. Retrieved from http://discovery.ac.uk/files/pdf/Licensing_Open_Data_A_Practical_Guide.pdf

[27] S. Villata, N. Delaforge, F. Gandon, A. Gyrard, An Access Control Model for Linked Data, in: OTM Workshops, in: LNCS, vol. 7046, Springer, Heraklion, Greece, 2011, pp. 454–463. Oct.

[28] P. Bellini, L. Bertocci, F. Betti, P. Nesi, Rights enforcement and licensing understanding for RDF stores aggregating open and private data sets, in: second IEEE International Smart Cities Conference, ISC2 2016, Trento, Italy, SLIDES, 12 to 15 September 2016. http://events.unitn.it/en/isc2-2016.

[29] Luciano De Bonis, Grazia Concilio, Eugenio Leanza, Jesse Marsh, Ferdinando Trapani. Co-Creative, Re-Generative Smart Cities. Smart Cities and Planning in a Living Lab Perspective. TeMA, Journal of Land Use, Mobility and Environment. 2014

[30] Luciano De Bonis and Ferdinando Trapani. "For a "Living (Lab)" Approach to Smart Cities", Smart Cities Atlas- Western and Eastern Intelligent Communities. Pp 143-158- November 2016.

[31] Bastiaan Baccarne, Dimitri Schuurman, Peter Mechant, Lieven De Marez . The role of Urban Living Labs in a Smart City - XX.V ISPIM Conference – Innovation for Sustainable Economy & Society, Dublin, Ireland

[32] Grazia Concilio. Urban Living Labs: Opportunities in and for Planning. Human Smart Cities, Rethinking the Interplay between Design and Planning - Springer International Publishing Switzerland 2016

[33] P. Bellini, I. Bruno, P. Nesi, N. Paolucci, "IPR centered Institutional Services and Tools for Content and Metadata Management", International Journal on Software Engineering and Knowledge Engineering, World Scientific Publishing Company, Volume 25, Issue 08, October 2015.

[34] GDPR, General Data Protection Regulation, https://www.eugdpr.org

[35] Melbourne Networked Society Institute. Cities as Living Labs Creating Innovative, Connected Cities - Discussion Paper 01/2015. http://networkedsociety.unimelb.edu.au/__data/assets/pdf_file/0007/1663756/MNSI-D01-15-Cities-as-Living-Labs.pdf

[36] Sinta Dewi Rosadi; Suhardi, Samuel Andi Krystian. Privacy Challenges in The Application of Smart City in indonesia. 2017 International Conference on Technology Sistems and Innovation. Bandung, October 23-24, 2017

[37] Nesti G. (2017) Living Labs: A New Tool for Co-production? In: Bisello A., Vettorato D., Stephens R., Elisei P. (eds) Smart and Sustainable Planning for Cities and Regions. SSPCR 2015. Green Energy and Technology. Springer, Cham. DOI https://doi.org/10.1007/978-3-319-44899-2_16.

[38] Pentaho Kettle tools, http://www.pentaho.com/

[39] HBase, Apache HBase, https://hbase.apache.org/

[40] Nodered, IoT programming tools: https://nodered.org

[41] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs, Future Generation Computer Systems, Elsevier, 2017, http://dx.doi.org/10.1016/j.future.2017.05.001

[42] C. Garau, P. Zamperlin, M. Azzari, P. Nesi, G. Balletto, M. Paolucci, THE ROLE OF KM4CITY DASHBOARD IN URBAN POLICIES: GOVERNANCE STRATEGIES FOR DYNAMIC URBAN SYSTEMS from 2nd International Conference on Smart and Sustainable Planning for Cities and Regions 2017, Bolzano/Bozen (Italy), 22-24 March 2017.

[43] C. Badii, P. Bellini, D. Cenni, G. Martelli, P. Nesi, M. Paolucci, "Km4City Smart City API: an integrated support for mobility services", 2nd IEEE International Conference on Smart Computing (SMARTCOMP 2016), St. Louis, Missouri, USA, 18-20 May 2016.

[44] Apache Solr, http://lucene.apache.org/solr/

Smart City Control Room Dashboards Exploiting Big Data Infrastructure

P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci
DISIT lab (http://www.disit.org http://www.km4city.org)
University of Florence, {name.surname}@unifi.it

Abstract: Smart City Control Rooms are mainly focused on Dashboards which are in turn created by using the so called Dashboard Builders tools or generated custom. For a city the production of Dashboards is not something that is performed once forever, but is a continuous working task for improving city monitoring, to follow special events and/or works, to monitor critical conditions and cases. Thus, relevant complexities are due to the data aggregation architecture and to the identification of modalities to present data and their identification, prediction, etc. In this paper, the architecture of a Smart City Control Room Dashboard Builder is presented. As a validation and test, it has been adopted for generating the dashboard in Florence city and other in Tuscany area. The solution proposed has been developed in the context of REPLICATE H2020 European Commission Flagship project on Smart City and Communities. **Keywords**: smart city dashboard, decision support system, widget, control room.

1. Introduction

In the development of a Smart City there is a great emphasis to the set-up of the so called Smart City Control Room, SCCR. A SCCR is an area (of one of more rooms) in which all the data are collected and high-level data/results are summarized and made accessible for the decision makers and for the city operators. In large metropolitan cities, the SCCR includes large panels/monitors (even covering large walls) in which the status of the city is reported presenting the view of the city with some synthesis, predictions, alert of data regarding: mobility, energy, social activities, environment, weather, public transportation, people flow, health, water, security, ICT, governmental, first aid, civil protection, police (118/112/911), fire brigade, hospital triage, and thus almost all the city resources expressed via KPI (Key Performance Indicator). Most of the KPI are representative of the status of resources deployed in the city. Some of the city monitored resources are critical infrastructures for the city functionality and life of city users such as: transportation, energy, security, health, water, civil protection, ICT, etc. In medium sized cities, the daily management of city resources is performed by a set of *separated city operators.* They autonomously manage their control rooms, accessing and rendering their own data to take their own decisions which may be limited in scope. For example, when the energy network has a problem in an area of the city the energy is rerouted to reach the all possible subareas via a different path; when the water network has a problem on major distribution tubes when possible the water is provided in other means; in presence of traffic congestion the red-light timing is acted to facilitate the flow and bus paths may be changed/rescheduled.

Once identified and understood the needs of having an integrated SCCR, immediately growth up the issue regarding what has to be shown on (i) the panels on walls and what on (ii) computers of the operators, (iii) how the data are collected and computed (in the case of prediction and early warning). These processes are very complex to be managed since the amount of information is heterogeneous and large and has to be easily understood by the observers of the panels on the wall and of the computers. It is not only a problem of usability is a problem of understandability, a problem of data representation, a problem of competence of the observers/operators, and of the decision makers. In most cases, they have to be trained to understand the data and graphics representations. They have to become confident on what they see to understand in deep all the single details represented on the screen. For example, we are used to understand: (i) a traffic representation observing the city map with red, orange, yellow, and green segments on the streets; (ii) a temperature and the humidity percentage, etc., while it is more difficult to read the tables of pollution, pollination, traffic flow trends by numbers, etc. Alarm signals in red, blinking signal, etc., may help on learning and understanding [Few 2006].

From the technical point of view, the tools for rendering information on SCCR are typically called Dashboards and are supported by big data aggregation tools [Badii et al., 2017]. The Dashboards should be capable to present real time data in several different manners with real time updates on screen autonomously H24 7/7 days, according to the refresh time of each data source.

Dashboards for control rooms should not be confused with business intelligence tools that produce graphics from the combination of data extracted from some sources (database, files, API, etc.). In most cases, business intelligent tools may access to data with faceted indexing and search, for example in SOLR or ElasticSearch. Those kind of Dashboards are focused on single view of data, filtering the drilling down on data, rather that representing the city KPI and status for example by using Apache Banana, or HUE.

DOI reference number: 10.18293/DMSVIVA2018-020

Moreover, the concept of Dashboard for SCCR is also often confused with the data aggregator tool that is a fundamental tool for the Control Room and city control in general, and can be regarded as the back office of the Control Room. Many solutions for control rooms and their backoffices, has been proposed such as IBM [IBM 2013] on services for citizens, business, transport, communication, water and energy; [Alcatel 2013] on governmental, educational, e-health, safety, energy, transport and utilities; etc. Most of these solutions present a multi-tier architecture ranging from 3 to 6 layers [Anthopoulos et al., 2014], [Filipponi et al., 2010], [Chourabi et al., 2012].

In this document, the Dashboard solution developed in the context of REPLICATE research and development projects of the European Commission, it is an SCC1 project of the European Commission on H2020 (http://www.replicate-project.org). The solution proposed is been based on Km4City Smart City Ontology Http://www.km4city.org [Badii et al., 2017], [Bellini et al., 2014]. Please note that the Dashboard Management System of DISIT Lab is released as Open Source on GitHub, see DISIT lab page. The present solution is managing more than 1.2 million of complex events/data per day.

The paper is structured as follows. In Section 22, the requirements of smart city control room are discussed. Section 3 presents the adopted smart city architecture. In Section 4, the dashboard system for the smart city control room is presented with its architecture. Section 5 reports a set of experimental results and lesson learnt. Conclusions are drawn in Section 6.

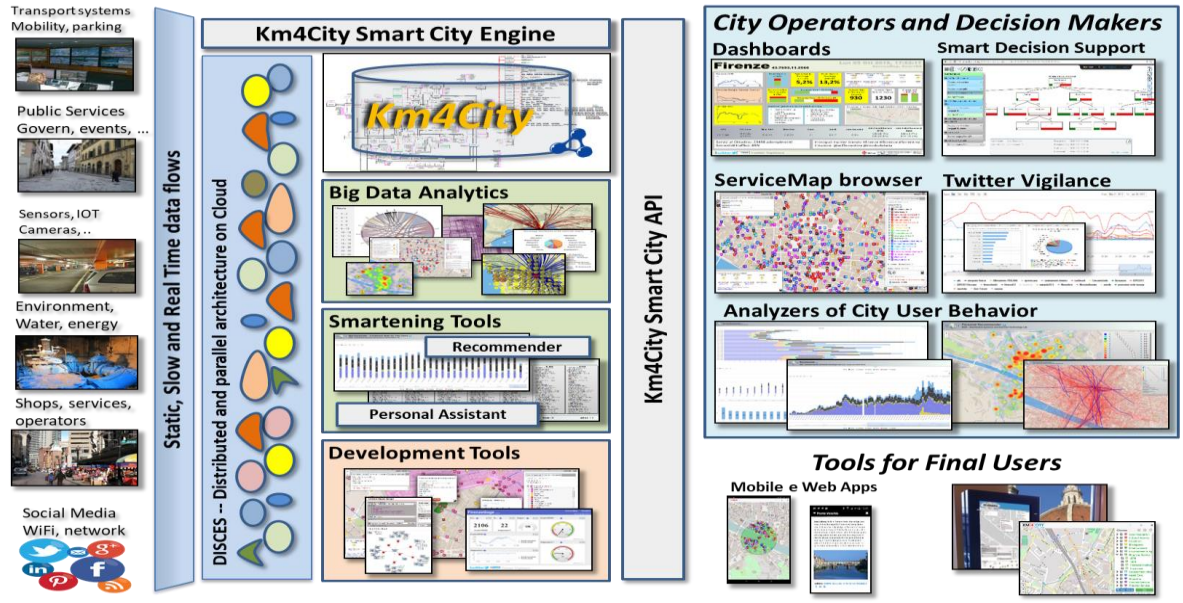
2. Smart City Control Room Requirements

In this section, the main requirements of Dashboards for SCCR are summarized. The have been collected during the research project by interviewing a number of operators and decisions makers belonging to several cities and nationalities.

A SCCR dashboard is substantially a decision support system tool, DSS, since it provides evidence of critical conditions, and may offer solutions. On this regard, it may integrate/exploit artificial intelligence algorithms, for example, reporting prediction, early warning, providing relationships among entities exploiting inference geospatial reasoning about what is located in the city: resources, structure, people, areas, critical infrastructures, etc. [Bellini et al., 2014], [Suakanto, 2013], [Gavin et al., 2016], [DeMarco et al., 2015].

According to our analysis, a Dashboard system for smart city has to be capable to:

- show data on widgets according to several graphic paradigms (tables, graphs, histograms, maps, kiviat, lists, tv camera, heatmaps, weather, critical city events, etc.) with a level of interactivity and animation;
- show data on autonomous and connected/ synchronized widgets;
- collect, show and keep update on screen data with automated refresh for each views;
- collect and show data coming from different big data and classic data sources (SQL, NoSQL, RDF, P2P, API, SOLR, etc.) also in aggregated manner;
- compose the Dashboard as a set of graphic and integrated widgets that can be separately set up assigning a number of parameters: data source, size, colors, shape, etc.;
- work with large amount of data providing high performances, as short response time;
- support by a flexible notification systems computing alarms and sending alerts, activating tickets for maintenance, automatizing actuators, etc.;
- provide actuators widgets together with showing graphs;



- provide support for collaborative production of dashboards and for co-working;
- provide support for embedding dashboard into third party web pages;
- provide data engine for collecting connection response time on different protocols, and for verifying the consistency of web pages via HTTP;
- integrating with IOT Applications by managing real time data and connecting its actuators to real time IOT applications;
- integrate dashboards in more complex dashboards;
- support authentication and authorization with the most general approaches such as LDAP, Kerberos, etc.;
- collect and get data from batch resources and in real time.

This means that each Dashboard should be composed by a number of configurable Widgets, each of them can collect data coming from several data sources. On each data stream, one or more criteria as firing conditions should be set up for the notification of alerts, intervention, etc.

In small cities of at least 100.000 inhabitants the number of sources to be integrated by the aggregator and represented in Dashboard can be in order of 10-20 while in larger cities they can rapidly grow for the presence of multiple operators for each utility. Thus, the complexity is also greater as the actors to be involved in.

Therefore, before starting with the development of the proposed Dashboard solution, a number of state of the art solutions and proposals have been analyzed. As nonfunctional requirements, the Dashboard system has to be scalable, interoperable with several tools, open source, usable, secure in protecting data views, and flexible.

To this and, a number of commercial and noncommercial solutions have been analyzed to identify a viable functional platform to be adopted, and then we decided to start the development of the solution since they have not satisfied all the requested functional and functional aspects. Most of the solutions which are proposed on the state of the arts derive from business intelligence solutions (e.g., SpagoBI, Tableux, OpenDataSoft, etc.), in which the tools provide some data mart (data virtualization) tool to access data sources and thus have powerful tools in this sense while provide poor tools on rendering and dashboard for control rooms that have to stay H24/7, rendering specific kind of structured data. For this reasons, a number of specific custom solutions have been proposed by many cities such as: London, Amsterdam, Dublin, etc.

3. Smart City Architecture

This section presents the overview of the Smart City architecture which is presently in place in the Tuscany area. With the aim of producing a smart city infrastructure for stimulating sustainable mobility, smart energy, and smart utility in the city, a data aggregator has been developed. It presents a front end layer for the City Dashboard and control room, Smart City API for web and mobile App, decision support tools, personal assistants, participative portals, crowd sourcing, etc. The data aggregation also support Data Analytics and Data Intelligence based on integrated data collected from public administrations open data, private data from operators, and personal data coming from social media and city users. In this paper, the architecture enabling the construction of the Control Room in terms of Dashboard and Data Aggregator is reported.

In the Km4City solution, **City Operators and Data Brokers** provide data which are collected by using streams or ETL processes which are scheduled on the Big Data processing back office based on DISCES (Distributed Smart City Engine Scheduler) tool. Among the data collected those provided in Open Data from the municipalities, Tuscany region (Observatory of mobility), LAMMA weather agency, ARPAT environmental agency, etc., and several private data coming from City/Regionals Operators: mobility, energy, health, cultural heritage, services, tourism, wine and food services, education, wellness, environment, civil protection, weather forecast, etc. Once the data are collected, the back office activates a number of processes for improving data quality, reconciliating data and converting data into triples for the RDF store of the Knowledge Base [Bellini et al., 2014], [Badii, et al., 2017], implemented by using a Virtuoso triple store. DISCES is allocating processes on a number of virtual machines allocated on the cloud according to their schedule.

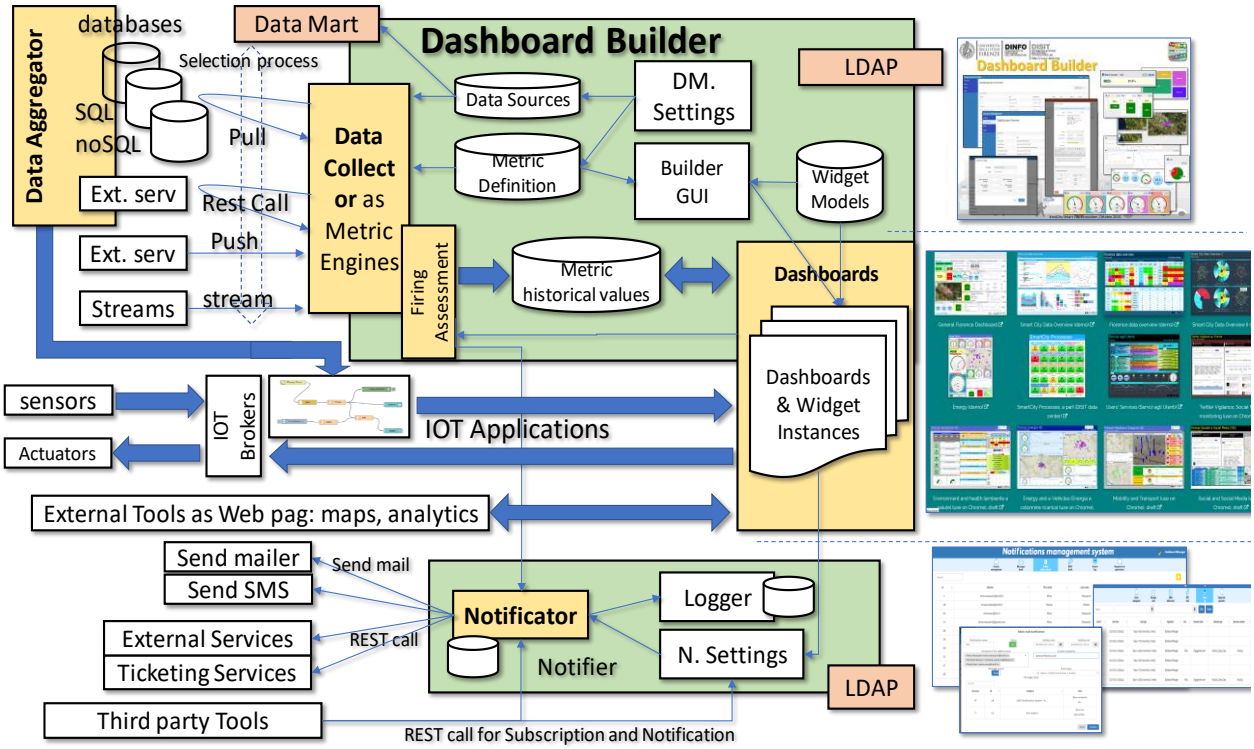
For semantic aggregation of data and service, it has been decided to exploit and improve the Km4City Ontology (http://www.km4city.org) [Bellini et al., 2014], as the main ontological model. Km4City is modeling multiple domain aspects related to mobility, services, Wi-Fi, cultural services, energy, structure (streets, civic numbers, green areas, sensors, busses, smart sensors, public structures, parking, city services, transportation, events, geographic locations, pharmacies, hospital and real time data of first aid, environment (with pollution and pollination, weather forecast, and private mobility with fuel prices.

In the smart city architecture in addition to the RDF store for the knowledge base, a number of noSQL Stores (namely: HBase and MongoDB) are adopted for storing tabular data as those arriving from sensors and user profiles.

4. Dashboard System Architecture

In Figure 2, the general architecture of the Dashboard solution is presented. The main components of the architecture are described in the following.

- **Data Aggregator** is a set of tools for collecting data from the field and from external services and reconciling them to the same city entities. To this



end in the proposed architecture the Km4City Aggregation has been adopted to have all data fit into the Km4City Ontological model [Bellini et al., 2014]. Thus. Data are provided from the Aggregator with Smart City API (rest CALL), SPARQL, SOLR and/or SQL queries.

- **Data Collector** is a multi-process engine (also called Dashboard Engine) for acquiring data from multiple data sources: SQL, noSQL, RDF/SPARQL, API, SOLR, etc., by using multiple protocols: HTTP, HTTPS, ODBC, etc. The Data Collector needs to have a configuration for each acquisition process, which produce a result, also named **Metric** or Measure. Some of these **Metrics** may be saved into a local data base for historical reason. In order to finalize the queries to be performed for collecting Metrics, specific tools may be used for **Data Mart** such as database browsers and drilling down into Data Sources. The collected data can be (i) saved into a data base of **Metric Historical Values**, or can be (ii) directly accessed from Widget/Dashboard for their visualization. The Data Collector, with its processes to acquire Metrics, is also capable to perform the real time Firing Conditions Assessment.
- **Firing Assessment** allows to compute the firing conditions on all the data. In the case in which a Firing Condition becomes true, a message event is sent to the Notificator service. Conditions can be estimated:
    - on Metric Historical Values taken in Push from the data base of the builder,
    - on the streams directly arriving from the Data Collector;
    - from the data stream arriving directly from outside,
    - From the external tools embedded as IFRAME into widget if there is some integration.
- **Dashboard Builder** is the core tool for creating dashboards. In the tool the user can set up **Data Sources**: IP address, protocol, user name and password for accessing at each specific the data source. Once the data sources are identified a number of **Metrics** can be defined. Then new Dashboards can be created taking interested **Metrics** and associating them to one or more **Widget.** A Widget may render/exploit the same Metric by means of different graphic models. For example, a temperature read every 5 minutes, can be visualized as current value on a thermometer, while the trend in the last 24 hour, last week, last month, etc. can be show on a graph. Therefore, composing a **Dashboard consists of** placing and configuring a set of Widgets into the Dashboard frame. The Dashboards are created by using the visual interface of the **Dashboard Builder**.
- **Integration with IOT.** This feature has been covered by (i) producing special MicroServices as blocks that can be used into IOT applications developed in NodeRED, (ii) connecting NodeRED applications with a number of IOT brokers. Point (i) implied the development of a layer that allowed the traditional

Dashboard Widget to be connected to IOT applications by using IOT Brokers (for example in NGSI and/or MQTT) and/or WebSockets. To this end, the most suitable IOT broker has been the Fi-Ware Orion Broker.

Dashboards are typically adopted for reporting KPI of the city and thus of specific infrastructures and services. This means that specific alerts and notifications have to be activated and managed at level of single Metric. On the other hand, the same Metric can be used on different dashboards and widgets for different purposes. So that, for each Widget of each Dashboard specific alerts and firing conditions can be set up. For example, when Metric M is adopted in Widget W of Dashboard D, the certain criterion C is saved and computed for firing (M, W, D, C). One or more Criteria can be defined (M, W, D, C1….Cc), each of them may produce multiple Notifications,
N: (M, W, D, C1 (N1,……., Nn),…., Cc(…)).
Therefore, the solution has been to design and develop a **Notifier** to

- Accept registrations of possible touples <m,w,d,c>, to enable the reception of *Notification Requests*, that are used to send Notifications according to different approaches.
- Accept registration by third-party tools, in addition to those of the Dashboard Builder, to send alarms about the firing of the registered conditions.
- Produce emails and REST calls, that can be used for calling SMS, as well as for the activation of maintenance ticketing system on OpenMaint tool for example.
- Log all the registrations and Notification Requests for further analysis and security evidence.
- Maintain and use a list of Notification recipients, that are the users which are going to receive the notifications. This list of uses is just listed as: name, surname, email, telephone (if any), role, organization.

To this end we suggest using specifically development tools, Such as the ServiceMap (http://servicemap.disit.org) which is used for knowledge base browsing over the city data as Km4City knowledge base, which is RDF store as well, exploiting geospatial reasoning and inference [Bellini et al., 2014]. In addition, the technical browsing on the RDF Graph Store may be needed to discover relationships. To this end, the LOG (http://log.disit.org) tool for browsing into any RDF store by using SPARQL and visual interface has been developed in the past and now used. This tool allows you to browse all the RDF stores accessible in the world which provide a public end point for SPARQL queries, from dbPedia, to Europeana, Geonames, Km4City, Camera, Senato, Getty, etc. [Bellini et al., 2014b].

As a conclusion, **Dashboard Instances** are available for view and activate corresponding widgets according to their Settings. The saving of data into the database of **Metric Historical Values**, allows keeping track of what has been visualized/monitored and thus enabling the replay of data logged. On the other hand, it is also possible to adopt widgets that (i) directly show/provide the data from in/out streams, respectively (for example, Civil protection alert status, etc.); (ii) directly render/visualize web page segments into the Dashboard (for example for showing social media analytics, traffic flow reconstruction tools).

5. Experimental Results and Validation

The solution proposed in this paper has been developed in REPLICATE flagship project SCC1 H2020 of European Commission for Florence City. It is also used in other large projects as Sii-Mobility Smart City Nazionale on Mobility and Transport of MIUR, RESOLUTE H2020 project for critical infrastructure and resilience of transport systems, and GHOST MIUR for Cagliari smart city experimentation. The proposed Dashboard solution is strictly connected with a number of tools of the Sii-Mobility/Km4City suite of tools which are used to perform smart city analytics, semantic browsing, and decision support, etc., such as: ServiceMap (http://Servicemap.km4city.org), smartDS, system thinking decision support based on Bayesian models (http://smartds.disit.org [Bellini et al., 2016]), Wi-Fi monitoring and predictions, smart parking prediction, traffic flow reconstruction and prediction, energy metering, first aid monitoring, environmental monitoring, social media monitoring and alerting, weather forecast, etc.; most of them based on clustering, machine learning, etc. [Badii et al., 2017].

In general, the decision makers in the city are politicians, assessors, and director of units. Some of the units have already adopted a high level of technology, for example, the mobility and transport, the civil protection, etc. In other units, the level of control is low since the technical activity is mainly delegated to City Operators such as: energy operators, public transportation, water management, health care hospitals, environmental agency, waste management, police and alert (112, 118, 911), etc. All of them have their own monitoring system, that is tuned to vertical control their own information and status. In some cases, the general information about weather forecasts and status is shared among the different operators. The dashboard can organize data according to different views/paradigms: vertical and horizontal view.

5.1 Example of Vertical thematic oriented

public transportation: position of busses, number of active busses, average delay at the bus stops, number of active taxis with respect to the plan, number of recharging stations for public vehicles, number of people on busses, percentage of busses with respect to the plan, number of events/incidents on traffic, status of the

underpasses, status of the bridges, number of tickets, number of free lots in parking, events in the city, etc.

private mobility: level of traffic flow, traffic flow reconstruction, number of free lots in parking, number of cycling people on paths, number of vehicles entered into the RTZ, number of vehicles entered in the city, number of truck on the main streets, number of shared bikes in percentage, events in the city, etc.

Energy: KW/h or MW/h consumed in the last hour for public services, KW/h or MW/h consumed in the last hour for recharge stations, KW/h or MW/h saved by public services since the usage of renewable energy production, KW/h or MW/h saved in the store and available for consumption, number of monitored meters grid, saved energy by following suggestion provided by Apps, etc.

Environmental data referring to different area of the city: temperature, humidity, PM10, PM2, CO2, wind, pollination, etc.; weather forecast; level of water in the rivers; level of drinkable water; Tons of collected garbage; Tons of collected garbage differentiated; earthquake monitoring; etc.

Social: status and stream of the social media; the most cited users on Tweet; the most mentioned hashtags on Tweet; the sentiment analysis on Tweets connected to the city; number of people moving the main area; number of people arrived by train in the City; TV cameras about the main points of interest in the city; number and list of the major entertainments, political, and sport events in the city; etc.

Security: data also presented on the Social Dashboard describing the presences in the city of people; any kind of event in the city from entertainment, sport, political, religious, etc.; eventual paths and area of the events; TV cameras observing the areas of the events; number of resources available for controlling the city and their deploy on the city map (cop, ambulances, 118, 911); aspects related to the environmental data; aspects related to mobility for planning the evacuations; aspects related to the public transportation for eventual change the paths.

Health: data reporting the status of the triage in the major hospitals; position of the emergency stations; number and position of the ambulances; environmental data for hot waves, temperature, etc., which can increase the risk of stroke.

5.2 Examples of Horizontal User Oriented

Event oriented: a dashboard for controlling the status of city with respect to a large event (such as the visit of Pope, or US President). In that case, the dashboard would be dedicated to monitor the paths that would be probably used, the status of traffic in those area, the number of police and security resources in those area, the Tv cameras, the hospitals, the other events and micro-events (accidents, crashes, fights), etc.

Tourism oriented: a dashboard reporting the major events in the city, the number of arrivals in the city, the number of people in the major points of the city, the number of accesses to the museums, etc.

As a conclusion, after the production of a set of Dashboards some of them where selected for trial and are reported in the web site for your public access from Http://www.km4city.org where most of them are presenting data that can be rendered on screen to public. This does not mean that the published data are open and that can be downloaded to be reused and published/exploited for other purpose. Moreover, the Dashboard can also contain data that cannot be visualized by public, for safety reasons, and thus are protected by some conditional access solution. For instance, since they are sensitive data describing the city status in real time or by predictions. Many examples of dashboards produced by the presented tool can be accessed from Http://www.km4city.org Presently about 10 qualified users have developed a total of 153 dashboards that have been accessed by thousands of viewing users per month.

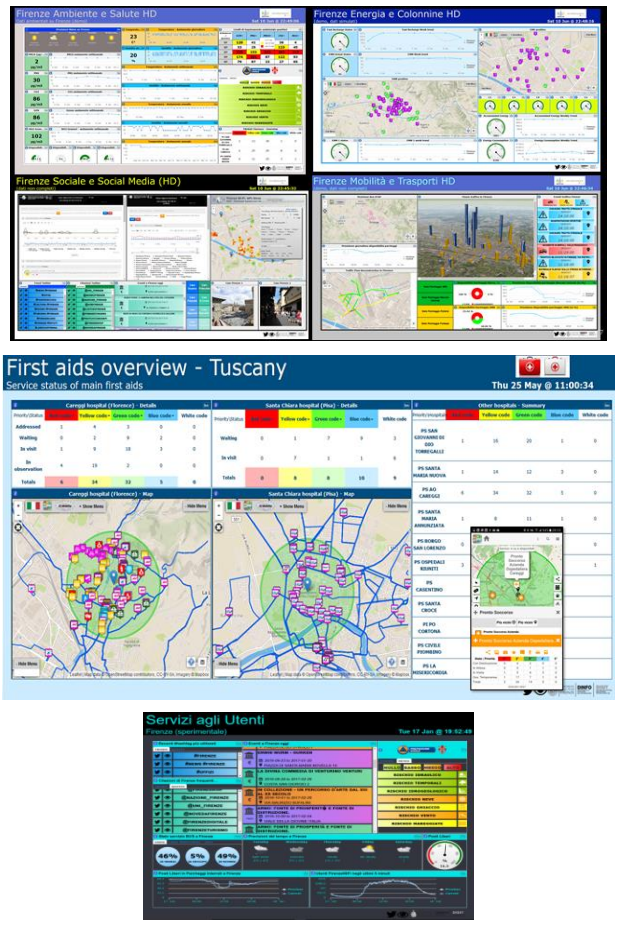


Figure 3: Florence Smart City Dashboards, dashboard reporting first aid status, and a final user dashboard for hotels

6. Conclusions

Smart City Control Rooms are focused on Dashboards. The dashboard production is a continuous working for improving city monitoring, adding more data, focusing on critical issues that may have seasonal aspects, follow

special events, and/or works for city improvement and maintenance. Complexity is due to the needs of data aggregation and to the identification of modalities to present data, their prediction, early warning, etc., and corresponding notifications. In this paper, the architecture and principles of the Dashboard Builder has been presented. As a validation, the tool has been adopted for generating the dashboards in Florence city and Tuscany area and accepted. An analysis of the possible views has been also provided. The solutions proposed have been developed in the context of REPLICATE H2020 European Commission Flagship project on Smart City and Community, and it has been validated during the usage with real city users. The dashboards produced and considered are all accessible online, and the Dashboard Builder is released in Open Source on github/disit.

7. Acknowledgements

Thanks to the European Commission for founding. All slides reporting logo of REPLICATE H2020 are representing tools and research founded by European Commission for the REPLICATE project. REPLICATE has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n° 691735).

8. References

[Alcatel 2013] Alcatel-Lucent Market and Consumer Insight team, “Getting Smart about Smart Cities Understanding the market opportunity in the cities of tomorrow”, Oct. 2013

[Anthopoulos et al., 2014] Anthopoulos L., Fitsilis P. "Exploring architectural and organizational features in smart cities." Advanced Communication Technology (ICACT), 2014 16th Int. Conference on. IEEE, 2014.

[Badii et al., 2017] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs, Future Generation Computer Systems, Elsevier, 2017,

[Bellini et al., 2013] Bellini P., DiClaudio M., Nesi P., Rauch N., "Tassonomy and Review of Big Data Solutions Navigation", as Chapter 2 in "Big Data Computing", Ed. Rajendra Akerkar, Western Norway Research Institute, Norway, Chapman and Hall/CRC press, ISBN 978-1-46-657837-1, eBook: 978-1-46-657838-8, july 2013, pp.57-101, DOI: 10.1201/b16014-4

[Bellini et al., 2014] P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, 2014

[Bellini et al., 2014b] Bellini P., Nesi P., Venturi A., "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", http://log.disit.org International Journal of Visual Language and Computing, Elsevier, 2014

[Bellini et al., 2016] E. Bellini, P. Nesi, G. Pantaleo, A. Venturi, "Functional Resonance Analysis Method based Decision Support tool for Urban Transport System Resilience Management", IEEE Int. Smart Cities Conference (ISC2), 12-15 Sept. 2016, Italy

[Chourabi et al., 2012] Chourabi, Hafedh, et al. "Understanding smart cities: An integrative framework." System Science (HICSS), 2012 45th Hawaii Int. Conference on. IEEE, 2012.

[DeMarco et al., 2015] De Marco, Alberto, Giulio Mangano, and Giovanni Zenezini. "Digital Dashboards for Smart City Governance: A Case Project to Develop an Urban Safety Indicator Model." Journal of Computer and Communications 3.5 (2015): 144-152.

[Few 2006] Few, Stephen. "Information dashboard design." (2006).

[Filipponi et al., 2010] Filipponi L., Vitaletti A., Landi G., Memeo V., Laura G.; Pucci P., "Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors," in Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on , vol., no., pp.281-286, 18-25 July 2010.

[Gavin et al., 2016] McArdle, Gavin, and Rob Kitchin. "The Dublin Dashboard: Design and development of a real-time analytical urban dashboard." (2016): 19-25.

[IBM 2013] IBM Institute for Business Value, “How Smart is your city? Helping cities measure progress”, [online]. Available: oct 2013, http://www.ibm.com/smarterplanet/global/files/uk__en_uk__cities__ibm_sp_pov_smartcity.pdf

[Suakanto, 2013] Suakanto, Sinung, Suhono H. Supangkat, and Roberd Saragih. "Smart city dashboard for integrating various data of sensor networks." ICT for Smart Society (ICISS), 2013 International Conference on. IEEE, 2013.

RADS: a smart Road Anomalies Detection System using Vehicle-2-Vehicle network and cluster features

Walter Balzano and Fabio Vitale
University of Studies of Naples Federico II, Italy
E-mail: w.balzano@unina.it, fvitale86@gmail.com

Abstract

Vehicle-2-Vehicle is an emerging and interesting field of research area due to the several possible application in IoT and self-driving vehicles. It allows communication between vehicles, allowing them to share information about traffic and road conditions. Road accidents are nowadays one of the major causes of casualties worldwide, and therefore increasing road safety is very important.

In this paper we present RADS: a smart Road Anomalies Detection System using Vehicle-2-Vehicle network and cluster features, a methodology which uses V2V and car distances in order to warn users of incoming dangers on the road, such as road blocks or existing accidents.

Keywords: VANET, Vehicle-2-Vehicle, Traffic detection, Road anomalies

1. Introduction

Latest development in network communication has contributed to the diffusion of powerful physical devices able to exchange information about their status and act upon reception of commands. These devices, well known under the name of Internet of Things (IoT), are getting increasing interest from the literature in the latest years. One of the most interesting application of these technologies is in communication between vehicles (as Vehicle-2-Vehicle or V2V) or between vehicles and fixed devices (known as Vehicle-2-Infrastructure or V2I). Moreover, allowing vehicles to "sense" the surrounding area, using technologies like LIDAR and RADAR, allows exploiting communication capabilities in order to inform nearby vehicles of traffic conditions and issuing warnings about road condition.

These technologies are also interesting considering the possible application of cloud algorithms, allowing better distribution of computation complexity over the network, reducing load while increasing service opportunities for the nodes[3, 4, 5].

According the World Health Organization[1], more than 1.25 million people die every year as a result of road traffic crashes. Moreover, road traffic crashes cost most countries 3% of their gross domestic products. Without proper action, it is predicted that it will become the seventh leading cause of deaths by 2030. We believe that technology, in particularly related to vehicle communication and cooperation, may help reducing these numbers by a large amount.

In this paper we present *RADS: a smart Road Anomalies Detection System using Vehicle-2-Vehicle network and cluster features*, which uses a smart combination of V2V and car distances evaluation in order to recognise anomalies (like accidents, road blocks, etc.) and promptly alert nearby vehicles of the potential danger.

2. Related works

Positioning systems and related services, in particular based on V2V and/or various localization systems are having a lot of attention from literature in the latest years[16]. With regards to positioning systems, many proposals were presented in the latest years, either based on satellites (like GPS or GLONASS), wireless networks (Wireless Positioning Systems or WPS)[8, 9, 10] or Inertial Navigation Systems (INS, based on sensors like accelerometers and gyroscopes)[7].

Vehicle Ad-hoc NETworks (VANETs), moreover, are having a lot of attention from literature due to several characteristics which have large improvement possibilities. For instance, routing information through the network poses issues due to limited bandwidth[15, 2, 17].

In *Vehicle mobility and communication channel models for realistic and efficient highway VANET simulation*[1] authors provide meaningful models for simulation of realistic VANET on highways. Deploying a real-world test bed is expensive, and is therefore useful to have a reliable model for testing purposes.

[1] http://www.who.int/mediacentre/factsheets/fs358/en/

DOI reference number: 10.18293/DMSVIVA2018-007

VANETs have many real-life potential usages. One of the most common usage for VANET regards traffic detection[18, 20, 14], but most cloud-based services may also be implemented in VANETs. For instance, in *Scalable VANET content routing using hierarchical bloom filters*[19], for example, authors discuss scalable routing for contents which also considers storage and searching of information on the network. It uses a hierarchical bloom filter in order to take users mobility in consideration. It shows an improvement in response time up to 45% while also reducing the traffic up to 85%.

In *VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks*[13] authors propose a way to offer standard cloud services (computation, storage or storage) using VANETs, considering two distinct sub-models: one for standard cloud services, like Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), and one consisting of vehicles which form a sort of temporary cloud.

In *DiG-Park: A Smart Parking Availability Searching Method Using V2V/V2I and DGP-Class Problem*[11] authors consider a novel methodology which allows users to find an appropriate place in a crowded parking lot using V2V in combination with a positioning system based on a Distance Geometry Problem algorithm.

In *PAM-SAD: Ubiquitous Car Parking Availability Model Based on V2V and Smartphone Activity Detection*[12] authors provide a method to determine available roadside parking slots using V2V in combination with GPS for localization and activity detection via smartphone.

Finally, for statistical purposes, it may be useful to record travel data for users, in order to further optimize trips based on preferred routes. For this scope, in *Hypaco–a new model for hybrid paths compression of geodetic tracks*[6] authors provide a method to compress geodetic data in a hybrid environment (GPS, WPS, INS) limiting needed storage memory.

Outline

The rest of the paper is organized as follows: first of all we have some considerations on localization systems and ways to measure distances between elements in a VANET situation (section 3), then we broadly describe our proposed system (section 4) and finally we present our conclusions and some ideas for future development (section 5).

3. Evaluation of vehicle distances, relative speeds and accelerations

In the latest years several methods have been proposed for distance evaluation between smart devices.

In outdoor environments it is possible to use satellite-based localization systems (such as GPS or GLONASS), which are very precise, but are also not reliable in narrow urban canyons and in indoor areas as they need to have clear sky-visibility for satellite connection.

For indoor situations, several localization systems based on WiFi signal strength (WPS localization) and inertial navigation systems (or INS), based on gyroscopes, compass and accelerometers, have been proposed.

WPS grants a good level of localization accuracy, but relies on a deployed network infrastructure, which needs to be calibrated over time in order to keep the positioning precision reliable.

On the other hand, inertial navigation systems do not need calibration. However, since the positioning is based on a previously calculated position (Dead-reckoning technique), error tends to grow over time due to sensors error which cumulates. Therefore, it is important to find a way to correct these errors over time.

Other methods include usage of RADAR/LIDAR devices which respectively use sound and light reflections in order to determine distance from nearby elements. The main usage of these technologies is nowadays in self-driving vehicles for detection of road obstacles.

Distance between vehicles has been used in several previous work in order to determine absolute vehicles position in space using Distance Geometry Problem algorithms (DGP). DGP uses evaluated distances in order to build a graph, which is then roto-translated using well-known fixed points (normally a network infrastructure) in order to determine the absolute position of each element in an area. This approach is interesting, but it is also quite expensive in terms of computation time and complexity.

Given distances, it is quite easy to calculate relative speeds and accelerations. Let's consider two successive distance matrices, we have that their point-per-point difference gives the relative speed matrix. Given two successive speed matrices, their difference indicates the relative acceleration matrix. Therefore, we only need three successive distance matrices in order to determine relative speeds and accelerations.

It is possible to parallelize all the needed $N \times N$ subtractions over the N vehicular nodes with little synchronization effort, distributing the $\theta(n^2)$ computational complexity to $\theta(n)$ on each node.

4. Proposed system

In this paper we propose a methodology which allows detection of anomalies in road movement for a large number of vehicles.

We use a smart combination of GPS localization and Vehicle-2-Vehicle network to determine cars distances and

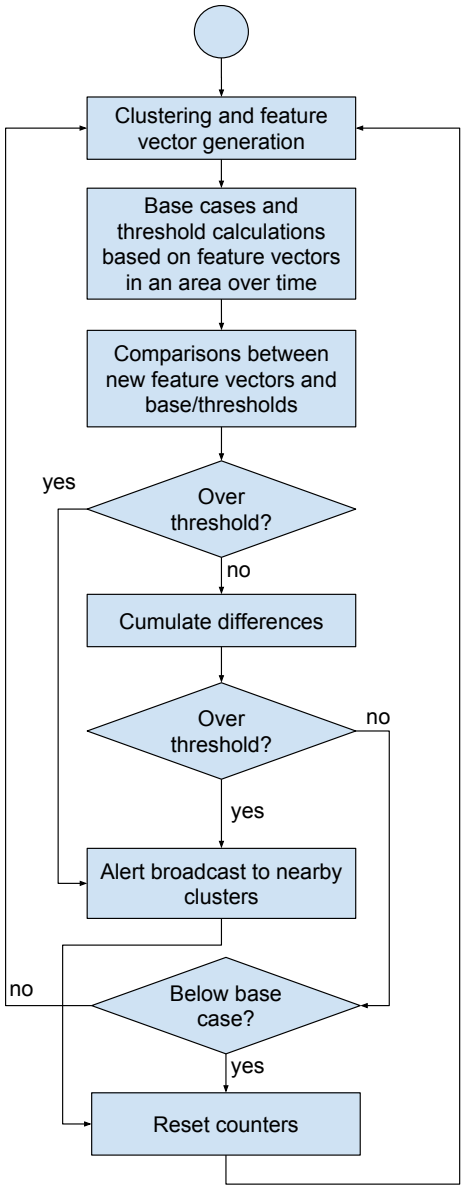


Figure 1. Project flowchart for detection of anomalies.

build an appropriate clusterization. Each cluster, using vehicles relative positions, is able to calculate its extension and density. This calculation is repeated with a set frequency, and irregularties in detected density (with respect to the base evaluated condition) are marked on the map and broadcasted to nearby clusters. The base condition is considered with regards to road features (like points of interest, signs, one-way/two-way roads, semaphores, number of lanes and so on).

There are several critical factors to be taken in consideration:

Distance matrix scan frequency higher scan frequencies increase system responsiveness at the cost of a higher computation complexity (over time), while lower scan frequencies have better tolerance for short-lived issues which should not be detected;

Environmental constraints it is important to consider different situation with regards to the type of road we are in: highways require different metrics from city streets or countryside roads. Moreover, it is also important to take time in consideration: for instance, during the night city streets are less crowded than in rush hour, while highways situation does not change as much in different time of days;

Proper network segmentation it is mandatory considering vehicles heading: vehicles which travel on the same road but in opposite directions should belong to distinct graphs;

Anomalies caused by single vehicles misbehavior cars which overtake several vehicles in a row or sideroad emergency stops should be ignored as they do not impact on proper road traffic. It is important to distinguish these special cases and apply proper corrections to the algorithms in order to reduce false-positives;

Distance matrix maximum size each vehicle should only consider a limited number of nearby vehicles for distance evaluation: a high vehicle count in distance matrix leads to higher computation time, which in turn reduces system effectiveness. On the other hand, having a small dataset, while being faster is also less accurate in detection of anomalies;

4.1. Base situation evaluation

Evaluating a proper base condition is mandatory in order to determine whether a variation should be considered an anomaly or just a common variation in values. For instance, when near a traffic light, it is normal to expect distances to reduce, while it is unexpected in highways. Therefore, we decided to create a grid, and for each cell a base condition is evaluated based on road type, average traffic and presence of road signs and traffic lights. This base condition, however, changes over time: for instance, a highway should have average traffic at all times, with slight better conditions during night time; city roads, on the other hand, may have peak hours during which the traffic is severely slowed, but also times at which the traffic is almost absent, for example during the night but not on weekend. One possible way to

find a proper base condition is by averaging values gathered over the last hour in each zone. All these information have a small footprint and can be stored and shared through the V2V network and optionally to a nearby local infrastructure (V2I).

Once a proper base situation has been established, it can be used in order to perform the needed comparisons with the realtime-measured values.

4.2. Anomalies recognition

For anomalies recognition, several possible algorithms have been proposed in the latest years. Since our project is focused on detection of traffic jams and slowdowns, we are going to clusterize the vehicles in an interesting area and only consider their in-cluster density.

Algorithm 1 K-means clustering algorithm with dynamic cluster count, based on number of nodes

Input: V = set of nodes (vehicles) in an area
Output: F = detected clusters feature vectors

1: n = ceil($|V|/30$) {cluster count is calculated dividing vehicles in the area by 30, then rounding up}
2: C = selectRandomCenters(V, n)
3: **repeat**
4: $C' = C$
5: **for all** $v \in V$ **do**
6: nc = findNearestCluster(C', v)
7: assignToCluster(v, nc)
8: **end for**
9: C = recalculateCenters(V, n)
10: **until** $C \neq C'$
11: $F = []$
12: **for all** $c \in C$ **do**
13: $F \leftarrow$ calculateFeatureVector(c)
14: **end for**
15: **return** F

Clusterization is made using a common *k-means* algorithm, which is efficient and lightweight. It is a partial clustering algorithm which is able to subdivide a set of objects in n subsets based on their attributes (position, speed and so on). The objective of the algorithm is to minimize the variance between elements of the same cluster. Each cluster has a center element, which is a centroid or average value. The algorithm begins by assigning n random centers, then grouping each element with the nearest center. Centers are then recalculated and the procedure is repeated until it converges to a stable solution. It is normally very fast and it does not require much computation power, and is therefore usable on embedded devices with slow processors. Cluster count is normally passed as parameter to *k-means* algorithm, but in our case we decided to go for a different approach, calculating the cluster number as a fraction of the total number of vehicles in an interesting area. This, however does not ensure that each cluster has exactly the same number of vehicles. We may have larger and smaller clusters, based for instance on the number of lanes of a single road in an area (see algorithm 1).

Algorithm 2 Data collection and threshold definition

Input: F = Clusters feature vectors sequences (calculated using algorithm 1, last 10 minutes) subdivided by map area
Output: T = Base conditions threshold array, one element for each area

1: $T = []$
2: **for all** $S \in F$ {for each area S in F} **do**
3: $t = \{\}$
4: **for all** $f \in S$ {for each feature vector f in sequence S} **do**
5: $t = t + f$
6: **end for**
7: **if** $T > MAX_T$ {MAX_T is the maximum possible threshold} **then**
8: $T = MAX_T$
9: **end if**
10: $T \leftarrow t/|c|$ {calculate thresholds and concatenate}
11: **end for**
12: **return** T

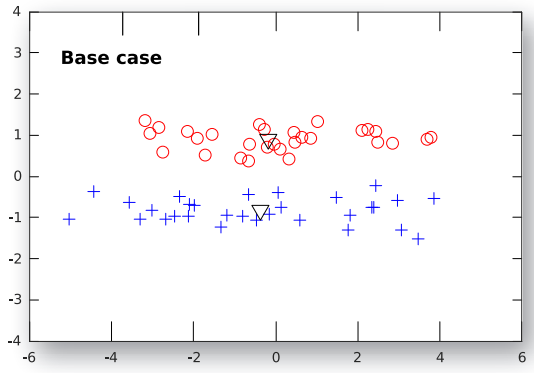


Figure 2. Vehicles are clustered (algorithm 1) and base case is evaluated over time(algorithm 2).

Once clusters are determined, each cluster is able to cooperatively calculate it size and vehicle density. This density is strictly monitored, and any alteration is considered. If the alteration value is larger than a set threshold (evaluated in algorithm 2), the alteration is considered an anomaly (see algorithm 3), and the information is broadcasted to nearby

Algorithm 3 Anomalies detection

Input: F = Clusters feature vectors sequences (calculated using algorithm 1, last 10 minutes) subdivided by area
T = Base conditions threshold array
A = Previous anomalies array, if available, else nil
Output: A' = New anomalies array

1: $A' = []$
2: **for** $i = 0$ **to** $|F|$ {for each aligned couple $f, t \in F, T$} **do**
3: **if** $A[i] =$ nil **then**
4: $A[i] = 0$
5: **end if**
6: $a = A[i] + F[i] - T[i]$
7: **if** $a < 0$ {no alert detected, resetting} **then**
8: $A' \leftarrow 0$
9: **else if** $a > T$ {over threshold, alert} **then**
10: broadcast(a, i)
11: $A' \leftarrow 0$
12: **else**
13: $A' \leftarrow a$ {cumulate possible alerts}
14: **end if**
15: **end for**
16: **return** A'

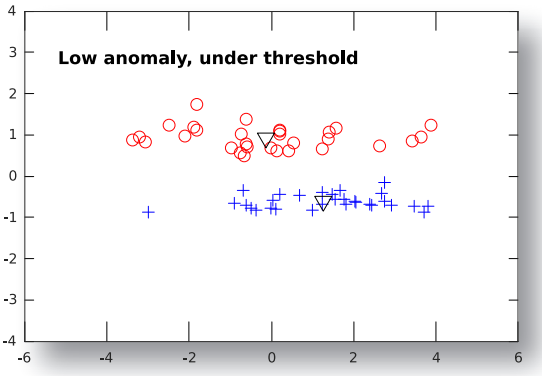


Figure 3. Small compression detected by algorithm 3. This anomaly does not trigger an alert because is does not go over calculated threshold.

clusters. However, since anomalies may cumulate over time (without going over the threshold), we also consider a case when situation is worse than the base condition. In this case we cumulate anomalies over time until a alert is triggered. If the situation returns to normality, every alert is reset.

For instance, if a group of vehicles speed decreases rapidly, their distance from the following cars would reduce as well, and this event increases the cluster density (more

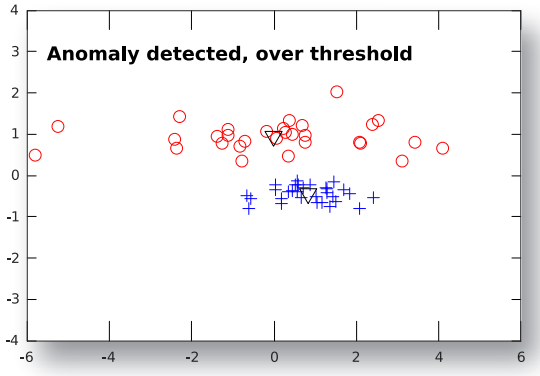


Figure 4. Heavy increase in vehicular density detected by algorithm 3 in an area triggers an alert to nearby clusters.

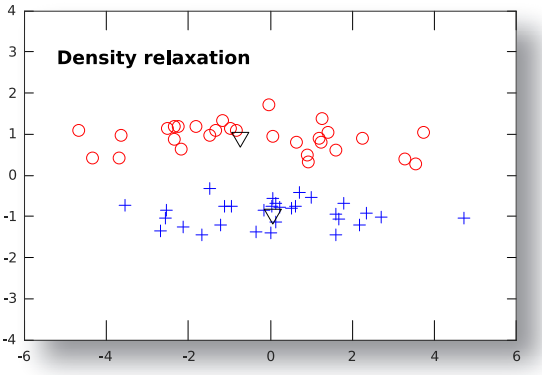


Figure 5. Situation is more sparse than base case. No alert is issued, and further base case evaluation will reduce the area threshold (algorithm 2).

vehicle in a smaller area). If we consider a low threshold, for example in an area where traffic is normally moving smoothly, this alteration is detected as a potential anomaly, and nearby clusters are alerted. On the other hand, if an area is subject to frequent slowdowns, the alteration may be considered "normal" and ignored. However the threshold has a well-established maximum value: even if an area is normally subject to heavy traffic, a traffic halt is always marked as anomaly. This allows users to avoid areas which are in a critical state, even if that specific area has a high threshold. Without this setting, some areas may never get marked.

5. Conclusions and future work

In this paper we presented RADS: a smart Road Anomalies Detection System using Vehicle-2-Vehicle network and cluster features, a novel methodology for detection and broadcasting of road anomalies using V2V and a modified k-means algorithm. After clusterization, vehicles cooperate in order to determine a valid base condition for a certain area, which is then used for determination of anomalies. When an anomaly is detected, is broadcasted to nearby clusters and users are alerted.

Future work may include finding other means for clusterization, a more reliable method for determination of the amount of clusters, or finding a faster algorithm for base condition determination.

References

[1] N. Akhtar, S. C. Ergen, and O. Ozkasap. Vehicle mobility and communication channel models for realistic and efficient highway vanet simulation. *IEEE Transactions on Vehicular Technology*, 64(1):248–262, 2015.

[2] F. Ali, F. K. Shaikh, A. Q. Ansari, N. A. Mahoto, and E. Felemban. Comparative analysis of vanet routing protocols: On road side unit placement strategies. *Wireless Personal Communications*, 85(2):393–406, 2015.

[3] F. Amato and F. Moscato. Model transformations of mapreduce design patterns for automatic development and verification. *Journal of Parallel and Distributed Computing*, 2016.

[4] F. Amato and F. Moscato. Pattern-based orchestration and automatic verification of composite cloud services. *Computers & Electrical Engineering*, 56:842–853, 2016.

[5] F. Amato and F. Moscato. Exploiting cloud and workflow patterns for the analysis of composite cloud services. *Future Generation Computer Systems*, 67:255–265, 2017.

[6] W. Balzano, A. Murano, and F. Vitale. Hypaco–a new model for hybrid paths compression of geodetic tracks. In *CCPS-2016: The International Conference on Data Compression, Communication, Processing and Security*, 2016.

[7] W. Balzano, A. Murano, and F. Vitale. V2v-en–vehicle-2-vehicle elastic network. *Procedia Computer Science*, 98:497–502, 2016.

[8] W. Balzano, A. Murano, and F. Vitale. Wifact–wireless fingerprinting automated continuous training. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on*, pages 75–80. IEEE, 2016.

[9] W. Balzano, A. Murano, and F. Vitale. Eenet: Energy efficient detection of network changes using a wireless sensor network. In *Conference on Complex, Intelligent, and Software Intensive Systems*, pages 1009–1018. Springer, 2017.

[10] W. Balzano, A. Murano, and F. Vitale. Snot-wifi: Sensor network-optimized training for wireless fingerprinting. *Journal of High Speed Networks*, 24(1):79–87, 2018.

[11] W. Balzano and F. Vitale. Dig-park: A smart parking availability searching method using v2v/v2i and dgp-class problem. In *Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on*, pages 698–703. IEEE, 2017.

[12] W. Balzano and F. Vitale. Pam-sad: Ubiquitous car parking availability model based on v2v and smartphone activity detection. In *International Conference on Intelligent Interactive Multimedia Systems and Services*, pages 232–240. Springer, 2017.

[13] S. Bitam, A. Mellouk, and S. Zeadally. Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, 22(1):96–102, 2015.

[14] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy. A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Communications Surveys & Tutorials*, 17(1):125–151, 2015.

[15] B. Feng, X. Kong, H. Yao, J. Li, and J. Peng. Study on routing protocol based on traffic density in vanet. *International Journal of High Performance Computing and Networking*, 10(6):481–487, 2017.

[16] A. Murano, G. Perelli, and S. Rubin. Multi-agent path planning in known dynamic environments. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 218–231. Springer, 2015.

[17] O. Salman, R. Morcel, O. Al Zoubi, I. Elhajj, A. Kayssi, and A. Chehab. Analysis of topology based routing protocols for vanets in different environments. In *Multidisciplinary Conference on Engineering Technology (IMCET), IEEE International*, pages 27–31. IEEE, 2016.

[18] C. Xin, C. Na, and B. Yeshuai. Analysis on key technologies of traffic prediction and path guidance in intelligent transportation. In *Intelligent Transportation, Big Data & Smart City (ICITBS), 2016 International Conference on*, pages 5–8. IEEE, 2016.

[19] Y.-T. Yu, M. Gerla, and M. Sanadidi. Scalable vanet content routing using hierarchical bloom filters. *Wireless Communications and Mobile Computing*, 15(6):1001–1014, 2015.

[20] Q. Zhang. *A pervasive prediction model for vehicular adhoc network (VANET)*. PhD thesis, Nottingham Trent University, 2017.

Enriching IAPS and GAPED Image Datasets with Unrestrained Emotional Data

Soraia M. Alarcão and Manuel J. Fonseca
LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
salarcao@lasige.di.fc.ul.pt, mjfonseca@ciencias.ulisboa.pt

***Abstract*—Elicitation of emotions is typically done through the presentation of emotionally salient material, like images or videos, thus requiring reliably annotated datasets. Although there are datasets with emotional information, these only describe either emotional polarities or discrete emotions. The only available dataset with both types of information restrained the participants during the study by separating a priori the images according to their polarity (positive or negative). In this paper, we describe an unrestrained study with 60 participants, where we asked them to rate the polarities and discrete emotions elicited by a set of images. The analysis of the emotional ratings made by the users revealed the most frequent correlations between the basic emotions. Furthermore, the analysis of the ratings' agreement among participants and existing datasets shows that our results are aligned with the existing ones. As a result of our study, we make available to researchers a more informative picture dataset annotated with emotional polarities and multiple emotions, as a complement to existing datasets.**

1. Introduction

The role of emotions in human cognition is essential given their importance in the daily life of human beings. Emotions play a critical role in rational decision-making, perception, human interaction, and intelligence [1], [2].

In the last decade, there has been an increasing body of work involving emotions: to improve content-based classification for both music and video, using photos and emotions conveyed by multimedia [3]; to gather emotional information from images through their visual content [4]; to observe the emotional state of a person using Electroencephalography [5]; and finally, to enhance the quality of recommendation systems [6]. Besides these examples, many studies in psychology and computer science involve manipulating emotions via emotional stimuli [7]. If a stimulus is relevant enough, an appraisal is automatically executed and will trigger reactions in measurable components of emotion, such as physiological responses, expressivity, action tendencies, and subjective feelings. Several methods have been introduced for priming participants, such as the presentation of emotionally salient material like images [8], audio [9], video [10], or text [11]. The use of the visual channel remains the most common to convey emotional stimulation [12].

DOI reference number: 10.18293/DMSVIVA2018-009

In the different areas of research based on visual stimulation, reliable datasets are important for the success of emotion induction. To that end, in 1997, the International Affective Picture System (IAPS) dataset was presented [13]. Later, in 2011 and 2014, two new datasets were created: Geneva Affective PicturE Database (GAPED) [12], and Nencki Affective Picture System (NAPS) [14]. These increased the availability of visual emotion stimuli, while trying to solve the problem of a limited number of pictures for specific themes. IAPS only provides valence and arousal, while GAPED has some information about the emotional polarity (negative, neutral or positive) of their images, but it is not enough for the cases where there is the need to use discrete emotions.

To minor the lack of emotional information, in 2005 and 2016, Mikels [15], [16], [17] and NAPS Basic Emotions (NAPS-BE) [18] were presented. Mikels collected descriptive emotional data on a subset of the IAPS to identify the elicited discrete emotions. Although this work enriched the emotional information associated to the IAPS dataset, we believe that the authors have restrained the choices of the participants by asking them to select discrete emotions only in a specific polarity (positive or negative), according to the subset where the image was placed a priori by the authors. This restriction prevented mixtures of positive and negative emotions. However, it is possible that an image arouses positive emotions in a person and negative in another. Finally, authors did not consider that images could be neutral.

In this paper, we present a study about the experience of viewing a set of images from the IAPS and GAPED datasets. We focused on the process of rating the images according to the emotions and polarities they elicited in the viewer, as well as the participants' insights during the experience. Although it would be interesting to use images from the NAPS-BE, it was not yet available when we conducted the study. Our contributions are: 1) a more complete and realistic picture dataset composed of 169 images, each annotated with information about the predominant emotional polarity (positive, neutral, and negative), the intensity of each discrete emotion elicited by the image, and the valence and arousal values from the original datasets; 2) the relationship between multiple emotions that arise when visualizing images, that are in line with the literature, thus confirming the quality of our dataset emotional annotation; 3) our experimental procedure designed to provide more comfort to the users, avoiding stress and fatigue.

2. Background and Related Work

In this section, we briefly explain what are emotions and how we can represent them. We also describe the most commonly used datasets of images to elicit emotions.

2.1. Emotions

Polarity provides a coarse indication of the emotional image content (positive, neutral, and negative). Emotions, on the contrary, give a more detailed description of the emotional information conveyed. These have been described as discrete and consistent responses to external or internal events with particular significance for the human organism [19]. This finer distinction of emotions provides a richer emotional classification, making it suitable for specific research purposes, like for instance studying the neuroanatomical correlations among basic emotions when a person is exposed to multimedia stimuli [20].

When talking about emotions, it is important to mention the subjectivity inherent, since multiple emotions can appear in the same subject while looking, for example, at a picture, as well as different subjects can feel different emotions when viewing the same picture, mainly due to each subject's current emotional state and "life experiences" [21], [22]. However, the expected affective response can be considered objective, as it reflects the more-or-less unanimous response of a general audience to a given stimulus [23].

Regarding the existence of multiple emotions while viewing an image, these correlations of basic emotions are a well-known phenomena in the field of psychology. One of the most important results was that when happiness rises, all other emotions decline; another one is that fear correlates positively with sadness and anger [24], [25].

2.2. Emotions Representation

There are two different perspectives towards emotion representation: categorical and dimensional. The first indicates that basic emotions have evolved through natural selection. Plutchik proposed eight basic emotions (acceptance, anger, curiosity, disgust, fear, joy, sadness, and surprise), from which we can define all the others [26]. Ekman based his work in the relationship between facial expressions and emotions derived from a number of universal basic emotions (anger, disgust, fear, happiness, sadness, and surprise) [27]. These emotions are considered universal since their external manifestation seems to be independent of culture and personal experiences [28].

In the dimensional perspective, which is based on cognition, the emotions are mapped into the Valence, Arousal and Dominance (VAD) dimensions. Valence goes from unpleasant to pleasant, arousal goes from states like sleepy to excited, and finally, dominance corresponds to the strength of the emotion [13], [29]. The most common model used is the Circumplex Model of Affect (CMA), where all affective states arise from cognitive interpretations of core neural sensations that are the product of valence and arousal [30].

TABLE 1. COMPARISION AMONG THE MOST COMMONLY USED DATASETS OF IMAGES.

Dataset	#Images	V-A	Polarities	Emotions
IAPS	1182	Yes	No	No
EmoPics	378	Yes	No	No
GAPED	730	Yes	Yes	No
NAPS	1356	Yes	No	No
POFA	110	No	No	Yes
KDEF	4900	No	No	Yes
NimStim	646	No	No	Yes
ArtPhoto	807	No	No	Yes
Abstract	228	No	No	Yes
Mikels	330	Yes	Yes[2]	Yes
NAPS-BE	510	Yes	No	Yes

In this work, we used Ekman's set of universal emotions (anger, disgust, fear, happiness, sadness, and surprise) complemented with the neutral emotion.

2.3. Image Datasets

In all the different areas of research based on visual stimulation, reliable databases are important for the success of emotion induction. In Tables 1 and 2, we briefly present the most commonly used datasets of images to elicit emotions.

As we can see in Table 1, only GAPED and Mikels provide information about the polarity of an emotion, i.e., negative, neutral or positive (Mikels does not consider the neutral polarity). In Mikels, the authors defined the emotional polarity of an image before the participants performed their rating about the discrete emotions. Given the subjectivity inherent to emotions, this could have restrained the results since it did not allow people to express positive emotions for "negative" images, and vice-versa. For example, Yoon *et al.* concluded that some of images did not have agreement between the tags assigned by the image creators and the ones given by image viewers [31].

Machajdik datasets (Art Photo and Abstract Paintings) [32], Mikels, and NAPS-BE discriminate the emotions elicited by images. However, Abstract Paintings is focused in a very specific type of images that are not usually found in personal collections, while the ratings for images of the Art Photo were only done by the artists. IAPS, Emotional Picture Set (EmoPicS) [33], and NAPS do not provide any information about the emotional content of their images, offering only valence and arousal information or physical characteristics of the images. Karolinska Directed Emotional Faces (KDEF) [34], NimStim Face Stimulus Set (NimStim) [35], and Pictures of Facial Affect (POFA)[1] were only labeled with facial expressions and corresponding emotions.

Some datasets have Valence and Arousal (VA) information, but no emotional data; others have emotional information, but no VA; and finally, only GAPED, NAPS-BE, and Mikels have both, but they are restrained and limited.

1. http://www.paulekman.com/product/pictures-of-facial-affect-pofa/

2. The emotional polarity (negative or positive) for each image was defined by the authors, not collected from the participants.

TABLE 2. DESCRIPTION OF THE MOST COMMONLY USED DATASETS OF IMAGES TO ELICIT EMOTIONS.

Dataset	Description
IAPS	It contains 1182 images, and provides a set of normative emotional stimuli for experimental investigations of emotion and attention. The authors rely on a dimensional view, in which emotions are defined by a coincidence of values on a number of VAD dimensions. Each picture is characterized in terms of their valence and arousal ratings. They were made by males, females and children using Self-Assessment Manikin (SAM) questionnaires during 10 years [36].
EmoPicS	It contains 378 standardized color images with different semantic contents, such as social situations, animals, and plants, selected from public online photo libraries and archives. Each image of the database was rated with their corresponding dimensional information: valence and arousal, and also with some physical characteristics of the given image: color composition, contrast, and luminance.
GAPED	It contains 730 pictures: 121 representing positive emotions using human and animal babies as well as natural sceneries, 89 for the neutral, mainly using inanimate objects, and 520 for the negative, using spiders, snakes, human rights violation, and animal mistreatment. The pictures were rated according to valence, arousal, and the congruence of the represented scene with moral and legal norms regarding Swiss legislation, since the study was conducted in Switzerland. These ratings were made by 60 subjects, where each subject rated 182 images.
NAPS	It contains 1356 realistic, high-quality images divided into five categories: animals, faces, landscapes, objects, and people. Besides valence, arousal and motivational direction (avoidance-approach) ratings, each image was annotated with some physical characteristics, namely color composition, contrast, and luminance. 204 subjects made the ratings, where each one rated 362 images, pseudo-randomly chosen from all the categories with the constraint that no more than three stimuli of the same category were presented in succession.
POFA	This dataset consists of 110 photographs of facial expressions that have been widely used in cross-cultural studies, and more recently, in neuropsychological research. All images are in black and white, and each image has a set of norms associated. It is important to note that the images are not identical in intensity or facial configuration.
KDEF	It is a set of 4900 pictures of human facial expressions of emotion suitable for perception, attention, emotion, and memory. Thus, special attention was given to photograph expressions at different angles, with soft light, and using t-shirts with uniform colors. A grid was used to center the face of the users during shooting, as well as position the eyes and mouth in certain coordinates of the image during scanning. The set contains 70 individuals, each displaying seven different emotional expressions, which were photographed from five different angles.
NimStim	It consists of 646 facial expression stimuli. Images include fearful, happy, sad, angry, surprised, calm, neutral, and disgusted expressions displayed by a variety of models of various genders and races. Examples of facial expressions were shown to the actors, for them to get an idea of what was the aim, and then they posed for each facial expression. Muscles were adjusted until the desired expression was achieved.
Art Photo	It contains 807 artistic photographs that were obtained by using the emotion label as search terms in the deviantArt site. The emotion label was determined by the artist who uploaded the photo, that was trying to evoke a certain emotion in the viewer of the photograph through the conscious manipulation of the image composition, colors, etc.
Abstract	It contains 228 images with combinations of color and texture, without any recognizable objects. To obtain ground truth, images were peer rated in a web-survey where the users could select the emotional category from amusement, anger, awe, contentment, disgust, excitement, fear and sad, for 20 images per session. 230 people rated approximately 280 images, where each image was rated about 14 times.
Mikels	This dataset is composed of 330 images from the IAPS, annotated with positive (amusement, awe, contentment, and excitement) and negative (anger, disgust, fear, and sadness) emotions. Thirty males and 30 females made the emotional category ratings in two studies, using a subset of negative images and a subset of positive images, with a constrained set of categorical labels.
NAPS-BE	This dataset contains 510 images from the NAPS, annotated with the emotions anger, disgust, fear, happiness, sadness, and surprise. It has 98 images depicting animals, 161 faces, 49 landscapes, 102 objects, and 100 people. Sixty seven females and 57 males made the emotional ratings, where each subject rated around 170 images.

3. Emotional User Study

In this section, we describe the study carried out, in which participants identified both the emotional polarity and emotions they felt while visualizing each image.

3.1. Participants

Sixty participants completed the study: 26 females and 34 males, with 70% of them belonging to the 18-29 age group, and almost 60% having a BSc Degree. None of the participants had participated in any study using the IAPS or GAPED, and the overwhelming majority had no knowledge about these datasets.

Regarding their emotional state at the beginning of the study, 31 participants classified themselves as neutral, 25 as positive, and only 4 as negative. Considering the discrete emotions (anger, disgust, fear, happiness, neutral, sadness, and surprise), the majority of the participants were feeling moderately happy or moderately neutral, both with a median of 3 in a scale of 1-5, with 1 corresponding to a weak feeling, and 5 to a strong feeling.

3.2. Apparatus and Material

A MacBook Pro (13-inch) computer was used with an application for participants to see the images and rate the emotions and polarities elicited by each image.

The dataset used in the study was composed of 86 images from the IAPS, 76 images from the GAPED, and 7 images from Mikels' dataset. It contained images with animals (cats, dogs, horses, sharks, snakes, spiders, tigers, among others), car accidents, children, death situations, diseases, fire, mutilation, natural catastrophes, poverty, and war scenarios. We chose a set of images that we believed to represent in a balanced way the discrete emotions throughout the valence-arousal space (see Figure 1).

Since it was impractical and even unpleasant for participants to annotate all the images in our dataset, and also due to the time it would take, we randomly divided our dataset into four subsets: DS0 to DS3. DS0 contained 57 images (30 IAPS, 20 GAPED, 7 Mikels), DS1 contained 40 images (20 IAPS, 20 GAPED), while DS2 and DS3 contained 36 images each (18 IAPS, 18 GAPED).

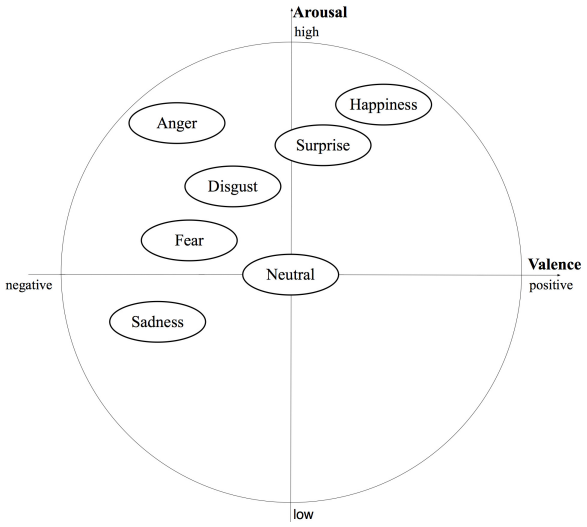


Figure 1. Adaptation of the Circumplex Model of Affect, mapping the discrete emotions into the Valence-Arousal plane [37].

All the participants rated each image of DS0, while images from DS1, DS2 and DS3 were rated by 20 participants. With this process, we managed to get a larger number of annotated images in the shortest time possible.

3.3. Design and Procedure

The experimental sessions took place in a room properly prepared for the task, aiming at providing comfort to participants, with adequate lighting and isolation from external noises. The option for the solo exhibition seeks to contribute to better control of external interference (e.g., comments from other participants, noise) that could interfere with emotional participant's experience [38].

We started by explaining the purposes of the study and how it would be held. To ensure the willingness of the subjects regarding negative images, we showed three images as examples of what could be expected. After that, the subjects could decide whether to continue or not the study. One participant (not included in the 60) decided not to continue the study due to medical issues. If they accepted, they should fill the participants' questionnaire with their personal information (age, gender, etc.), and the classification of their current emotional state (polarity and emotions).

The first screen of the application presented a summary of the most important aspects of the study. Then, seven blocks of images were presented sequentially, with about 14 images on each block. Each image (with a resolution of 640x480 pixels) was displayed randomly during 5 seconds, and after the visualization, participants evaluated their emotional state (regarding the polarity felt), and rated it for each of the emotions (see Figure 2). To obtain the participants' emotional reactions without practical limitations (e.g. specialized equipment for collecting physiological signals), we adopted a 5-point Likert scale for each emotion.

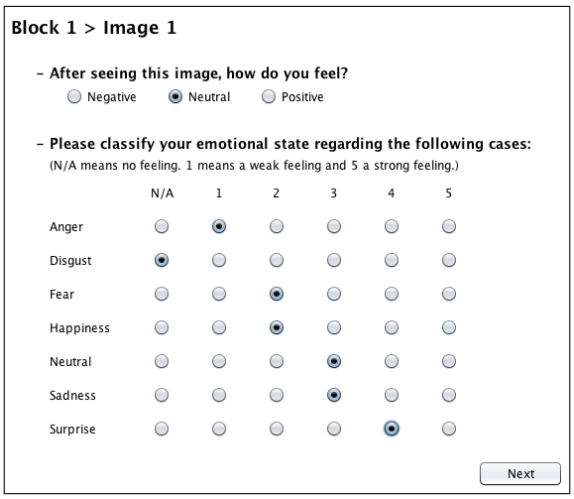


Figure 2. Rating screen of the application with the 5-point Likert scale.

This process was repeated for each image of the seven blocks of images of our study. Although in similar studies participants usually had a limited time to answer, we decided not to do it. This way, we allowed participants to spend the time they needed, without feeling pressured to respond or even stressed out. We also provided a 30 seconds interval between each block of images, during which only a black screen was displayed, to relax the user and avoid fatigue.

To verify and validate if our procedure had any error and if it was completely clear to the subjects, we performed a pilot test with a 27 years old male and a 18 years old female. With the exception of an image that was duplicated, none of the subjects had any doubt or detected any error in our study. An interesting aspect identified in this pilot test was the different sensitivities of the participants to the negative images. One subject considered the majority of the images very violent, while the other considered them almost neutral, and in some cases he enjoyed the consider negative content.

4. Emotional Classification Procedure

In this section, we describe the procedure used to classify each image based on the participants' ratings both in terms of the dominant polarity and discrete emotions.

To assign an emotional polarity to an image, we chose the polarity with the highest number of votes. In Table 3, we present examples of the distribution of votes across each polarity, while Figure 3 depicts the corresponding images.

TABLE 3. EXAMPLES OF THE DISTRIBUTION OF VOTES ACROSS EACH POLARITY.

Image	Negative	Neutral	Positive	Assigned Polarity
1460.jpg	0.0%	13.3%	86.7%	Positive
Sn087.jpg	20.0%	68.3%	11.7%	Neutral
9925.jpg	40.0%	50.0%	10.0%	Neutral
Sp044.jpg	40.0%	50.0%	10.0%	Neutral
3017.jpg	75.0%	20.0%	5.00%	Negative

(a) 1460.jpg (b) Sn087.jpg (c) 9925.jpg (d) Sp044.jpg (e) 3017.jpg

Figure 3. Examples of images from our dataset depicting: (a) kitten, (b) snake, (c) fire, (d) spider, and (e) mutilation.

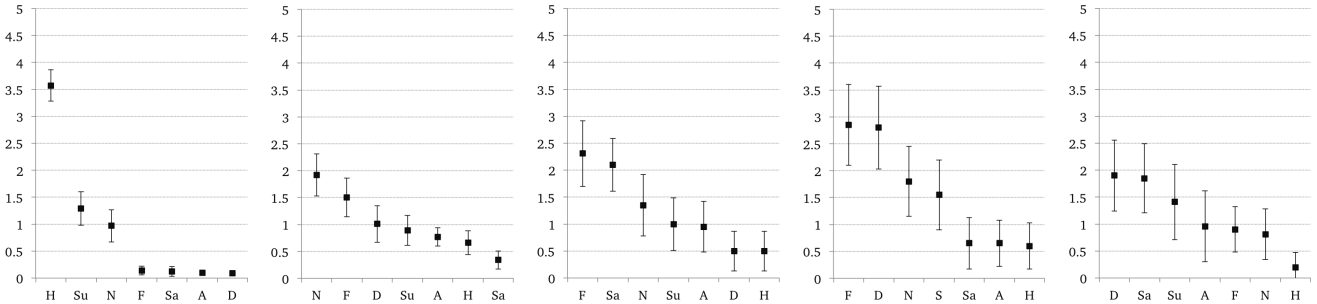


Figure 4. Examples of Confidence Intervals of images from our dataset, and how they are classified according to our procedure: (a) happiness emotion, (b) neutral and fear emotions, (c) fear, sadness and neutral emotions, (d) fear, disgust, neutral, and sadness emotions, and (e) undifferentiated.

We considered that an image could transmit up to four emotions, with no constraints about their polarity. We made this decision because Posner *et al.* stated that "*individuals do not experience, or recognize, emotions as isolated, discrete entities, but that they rather recognize emotions as ambiguous and overlapping experiences*" [30].

To identify the dominant emotions for each image, we followed the procedure from Mikels *et al.* [15]. However, since we are considering more emotions per image than Mikels (four vs three), our procedure is slightly different. For each image, we computed the mean of the ratings assigned by participants to each emotion, and a 90% t-based Confidence Interval (CI) around each mean. Then, the emotions' label was determined according to the overlap of the CIs for each emotion. If the mean for one emotion is higher than the means of all the other emotions, and if the CI for that emotion does not overlap with the CIs for the other emotional labels, it is classified as a single emotion (see Figure 4a). If two, three or four means are higher than the rest, and the intersection between their CIs is not empty, the image is categorized as blended (see Figures 4b - 4d). If more than four CIs overlap, the image is classified as undifferentiated (see Figure 4e).

In our study, and contrary to what Mikels did, we could have images with a mix of negative and positive emotions.

5. Results

In this section, we present the polarities agreement and emotional labels assigned to each image. We also present the most elicited emotions together. Finally, we present observations made by our participants during the study.

5.1. Agreement of Polarity Among Users

In Figures 5 and 6 we can observe, in detail, the votes of the users for each image in our dataset. From the 82 images classified as negative, 77 images had more than 50% of negative votes. The remaining votes were mainly neutral (45 images were rated with at most 30% of neutral votes, while 47 images had at most 5% of positive votes).

Regarding the 66 images classified as neutral, 62 of images had more than 50% of neutral votes. The remaining votes were usually rated more often as negative than positive (37 images with at most 30% of negative votes, while 41 had at most 15% of positive votes). Finally, all the 21 images classified as positive had more than 50% of positive votes. Eighteen images had at most 5% of negative votes, while 10 were rated with at most 30% of neutral votes.

In summary, all polarities were very well identified. When there was some mixing with either the positive or negative polarity, they were mixed with the neutral polarity. For the neutral polarity, it was mainly mixed with the negative polarity.

5.2. Agreement of Polarity Among Datasets

We compared our results only with GAPED because IAPS does not provide polarity information, and although Mikels provides information about the polarity, it was classified by the authors not by the participants.

We analyzed 76 images (33 negative, 9 positive, and 34 neutral) from the GAPED. For the neutral and positive polarities, we achieved an agreement of 100% for each. For the negative, the achieved agreement was 69%.

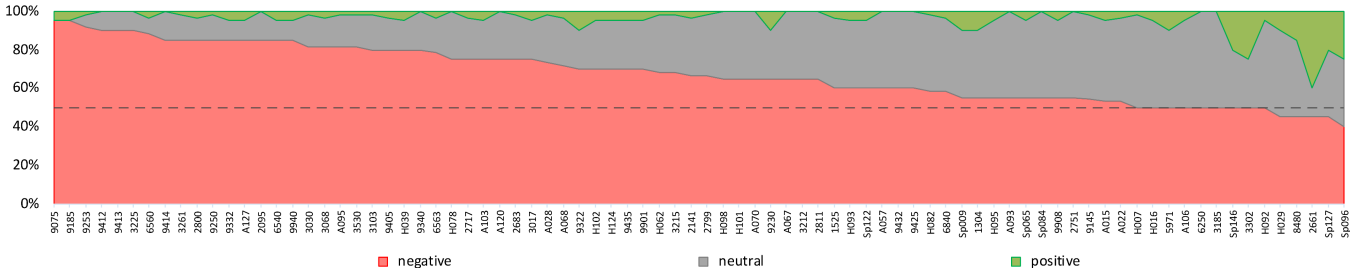


Figure 5. Images classified as negative in our dataset. We show the percentage of votes that users assigned to each category. (best seen in color)

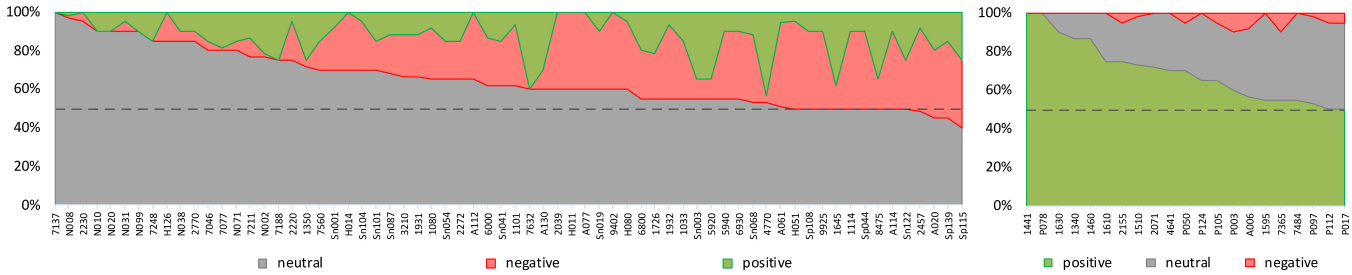


Figure 6. Images classified as neutral (left) and positive (right) in our dataset. We show the percentage of votes that users assigned to each category. (best seen in color).

The biggest mixed was with the neutral polarity (28%), while the mix with the positive polarity was very small (3%). Dan-Glauser *et al.* also reported that their results in GAPED had a high percentage of negative valence ratings overlapping with the neutral for animal mistreatment, spider, human concern, and snake pictures [12].

5.3. Emotional Labels

From the 169 images of our dataset, we obtained 60 images annotated with a single emotion (35.5%), 87 classified as blended (51.5%), with 29 referring to the combination of two emotions (17.2%), 31 to three emotions (18.4%), and 27 for four emotions (16.0%). Finally, we only had 22 images classified as undifferentiated (13.0%).

If we compare our results with those presented in Mikels dataset, we obtained more 6% of images classified with a single emotion, and less 8% undifferentiated images (29.0% vs 36.9% in Mikels considering only three emotions [15]). If we consider up to four emotions in an image, we have less 24% undifferentiated images (13.0% vs 36.9% in Mikels). In the case of blended images, we have around 21% more images considering up to four emotions in an image.

On the whole, we have a larger number of images annotated with emotional data. Thus, our dataset is more informative about the emotional labels assigned to images.

In Figure 7, we can see the different emotional labels resulting from the classification process. For the blended, we have for example DFSu, i.e., an image that contains the emotions disgust, fear, and surprise. The resulting label does not take into account the weight of each of the emotions present in an image, i.e., label DFSu includes the following combinations: DFSu, DSuF, FDSu, FSuD, SuDF, and SuFD.

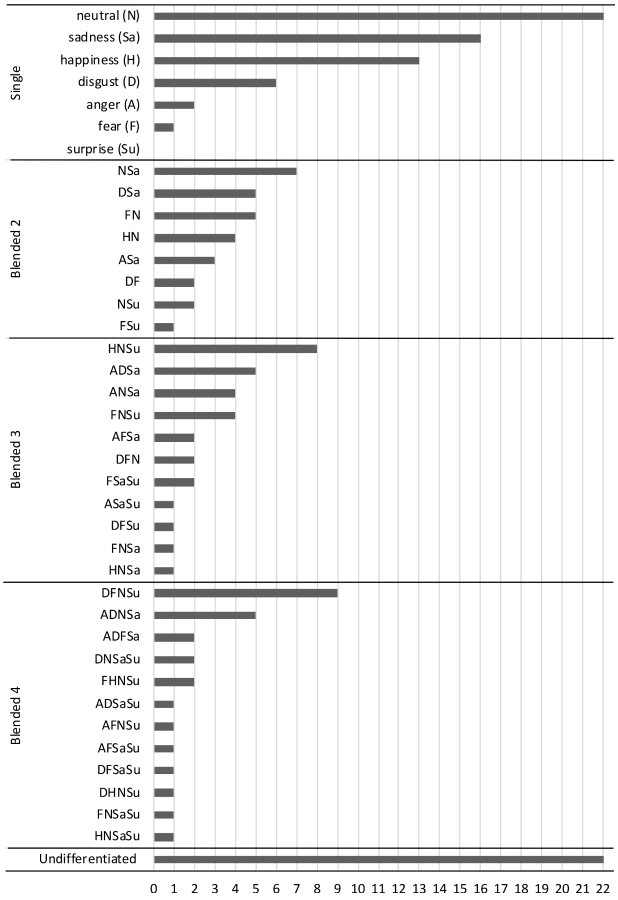


Figure 7. Emotional labels that result from the classification process.

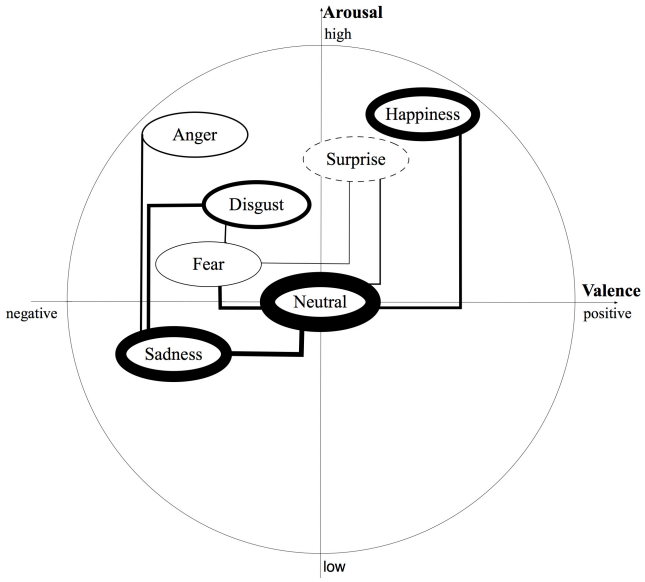


Figure 8. Relationship between single emotions, and between the blending of two emotions.

5.4. Relationship Between Emotions

In Figure 8, it is possible to analyze the most elicited single emotions, and the relationship between two emotions. For that, we considered the frequency of occurrence of each emotion. The thicker the line, the bigger is the number of images that elicited that emotion (single) or the greater is the relationship between two emotions (blended). A dashed line indicates that there were no images that elicited that emotion.

The most elicited single emotions were neutral, happiness, sadness, and disgust. The most obvious relations are between the emotions neutral and sadness, neutral and fear, neutral and happiness, and sadness and disgust. Anger, fear, and surprise are the emotions less elicited alone. However, surprise tends to appear in conjunction with fear and neutral emotions. In the case of anger, there is some relation with the elicitation of sadness, while fear was elicited together with disgust, as well as with surprise and neutral.

Regarding the correlations between basic emotions, and considering Figures 7 and 8, we confirm the results reported in the literature. Happiness negatively correlates with all the other basic emotions. Anger shows correlation with fear and disgust. There is also correlation between sadness and fear, and between sadness and disgust. Finally, fear was also correlated with disgust. Overall, our results are in line with those reported in previous studies [25], [39], [40], [41].

5.5. Observations from Participants

During each session, participants were encouraged to share with us their opinions/comments about the experience. More than 40% of the participants mentioned some type of difficulty in understanding the content of some of the images, leading to confusion about their feelings.

The majority identified the lack of context as the main reason for this, e.g., some participants did not understand if an animal in front of a car will be hit by it or not. In this case there is confusion between feeling negative if the animal is hit, and neutral or positive otherwise.

Five participants claimed that surprise is subjective, difficult to understand, and also difficult to elicit from an image. There seemed to be some exceptions to this, such as a shark moving as it is attacking a person or images with unexpected content like a lamp or stairs. A couple of participants indicated us that none of the images was able to trigger anger.

Regarding the personal taste of the participants, some appreciated specific content such as snakes (4), spiders (3), or aquatic animals (1), while others did not appreciate it at all. However, some of them considered images with those animals "beautiful", mainly due to the colors in them. Three participants declared that they were not sensitive to some images, such as a children smiling, leading them to feel neutral, although they considered that they should feel "happy". Finally, some participants also mentioned that the emotional content of the previous visualized image may interfere in the way they were feeling at that moment.

6. Conclusion

We described an unrestrained study performed with 60 participants to annotate a dataset of images with the polarity and discrete emotions elicited by each image. During our study there were no restrictions in the selection of the emotions, being possible for a user to associate a positive and a negative emotion to the same image.

We presented the relationship between multiple emotions that arouse when visualizing an image, and we verified that they were in line with existing literature. Moreover, we also presented our experimental procedure designed to avoid stress and fatigue, providing more comfort to the users.

We made a more complete and realistic picture dataset composed of 169 images publicly available to the community[3], as a new contribution to complement the already existing datasets. Each image was annotated with the emotional polarities (positive, neutral, and negative), discrete emotions (anger, disgust, fear, happiness, neutral, sadness, and surprise), and the original valence and arousal information.

Having in mind all the inherent subjectivity of emotions, the different constraints that could affect the participants judgement (current emotional state of the user, user's ability to evaluate what they felt, among others), the overall good agreement among participants, and between our dataset and the GAPED dataset, we can consider that the results achieved by our study are reliable and useful for the elicitation of emotions.

As future work, we intend to use our procedure to annotate more images with polarities, discrete emotions, and the physiological signals collected from the users while viewing the images.

3. http://www.di.fc.ul.pt/~mjf/research/ul-eps/UL-EPS_2018.xlsx

Acknowledgments

This work was supported by national funds through Fundação para a Ciência e Tecnologia, under LASIGE Strategic Project - UID/CEC/00408/2013.

References

[1] A. R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*. Harper Perennial, 1995.

[2] R. Picard, "Affective computing," MIT Media Laboratory, Perceptual Computing Section, Tech. Rep. 321, 1995.

[3] P. Dunker, S. Nowak, and C. Lanz, "Content-based Mood Classification for Photos and Music," in *Multimedia Information Retrieval*, 2008.

[4] D. Joshi, R. Datta, E. Fedorovskaya, Q.-t. Luong, J. Z. Wang, L. Jia, and J. Luo, "Aesthetics and Emotions in Images [A computational perspective]," *Signal Processing Magazine*, 2011.

[5] D. O. Bos, "EEG-based Emotion Recognition: The Influence of Visual and Auditory Stimuli," *Capita Selecta Paper*, 2006.

[6] M. Tkalčič, A. Kosir, and J. Tasic, "Affective recommender systems: the role of emotions in recommender systems," in *Workshop on Human Decision Making in Recommender Systems*, 2011.

[7] K. C. Klauer, "Affective priming," *European Review of Social Psychology*, 1997.

[8] S. Wang and X. Wang, "Emotion semantics image retrieval: An brief overview," in *Affective Computing and Intelligent Interaction*, 2005.

[9] T. Li and M. Ogihara, "Detecting emotion in music," in *Music Information Retrieval*, 2003.

[10] K. Sun, J. Yu, Y. Huang, and X. Hu, "An improved valence-arousal emotion space for video affective content representation and recognition." Multimedia and Expo, 2009.

[11] V. L. Rubin, J. M. Stanton, and E. D. Liddy, "Discerning Emotions in Texts," in *Exploring Attitude and Affect in Text: Theories and Applications*, 2004.

[12] E. S. Dan-Glauser and K. R. Scherer, "The Geneva affective picture database (GAPED): A new 730-picture database focusing on valence and normative significance." *Behavior Research Methods*, 2011.

[13] P. Lang, M. Bradley, and B. Cuthbert, "International affective picture system (IAPS): Affective ratings of pictures and instruction manual." NIMH Center for the Study of Emotion and Attention, Tech. Rep., 1997.

[14] A. Marchewka, Ł. Zurawski, K. Jednoróg, and A. Grabowska, "The Nencki Affective Picture System (NAPS): introduction to a novel, standardized, wide-range, high-quality, realistic picture database." *Behavior Research Methods*, 2014.

[15] J. A. Mikels, B. L. Fredrickson, G. R. Larkin, C. M. Lindberg, S. J. Maglio, and P. A. Reuter-Lorenz, "Emotional category data on images from the International Affective Picture System." *Behavior Research Methods*, 2005.

[16] A. Smith, "A new set of norms." *Behavior Research Methods*, 2004.

[17] ——, "Smith2004norms.txt," *Retrieved October 2, 2004 from Psychonomic Society Web Archieve*, 2004.

[18] M. Riegel, Ł. Zurawski, M. Wierzba, A. Moslehi, Ł. Klocek, M. Horvat, A. Grabowska, J. Michałowski, K. Jednoróg, and A. Marchewka, "Characterization of the Nencki Affective Picture System by discrete emotional categories (NAPS BE)." *Behavior Research Methods*, 2016.

[19] E. Fox, *Emotion Science: Cognitive and Neuroscientific Approaches to Understanding Human Emotions*. Palgrave Macmillan, 2008.

[20] R. D. Lane, E. M. Reiman, G. L. Ahern, G. E. Schwartz, and R. J. Davidson, "Neuroanatomical correlates of happiness, sadness, and disgust." *The American journal of psychiatry*, 1997.

[21] Y. Choi and E. M. Rasmussen, "Searching for images: The analysis of users' queries for image retrieval in American history," *Journal of the American Society for Information Science and Technology*, 2003.

[22] K. A. Olkiewicz and U. Markowska-kaczmar, "Emotion-based image retrieval - An artificial neural network approach," in *Computer Science and Information Technology*, 2010.

[23] A. Hanjalic, "Extracting Moods from Pictures and Sounds: Towards Truly Personalized TV," *Signal Processing Magazine*, 2006.

[24] C. E. Izard, *The Psychology of Emotions*. Plenum Press, 1991.

[25] S. Schmidt and W. Stock, "Collective indexing of emotions in images. A study in emotional information retrieval," *Journal of the American Society for Information Science and Technology*, 2009.

[26] R. Plutchik, "The Nature of Emotions," *American Scientist*, 2001.

[27] P. Ekman and R. J. Davidson, *The nature of emotion : fundamental questions.* New York : Oxford University Press, 1994.

[28] O. da Pos and P. Green-Armytage, "Facial expressions, colours and basic emotions," *Journal of the International Colour Association*, 2007.

[29] Y. Liu, O. Sourina, and M. K. Nguyen, "Real-Time EEG-Based Emotion Recognition and Its Applications," *Transactions on Computational Science*, 2011.

[30] J. Posner, J. a. Russell, and B. S. Peterson, "The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology." *Development and psychopathology*, 2005.

[31] J. Yoon, "Utilizing quantitative users' reactions to represent affective meanings of an image," *Journal of the American Society for Information Science and Technology*, 2010.

[32] J. Machajdik and A. Hanbury, "Affective image classification using features inspired by psychology and art theory," *International Conference on Multimedia*, 2010.

[33] M. Wessa, P. Kanshe, P. Neumeister, K. Bode, J. Heissler, and S. Schoenfelder, "EmoPics: Subjektive und psychophysiologische Evaluationen neuen Bildmaterials für die klinisch-bio-psychologische Forschung," *Zeitschrift für Klinischer Psychologie und Psychotherapie, Supplement, 1/11, 77*, 2010.

[34] D. Lundqvist, A. Flykt, and A. hman, *The Karolinska Directed Emotional Faces - KDEF.* CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, 1998.

[35] N. Tottenham, A. Borscheid, K. Elertsen, D. Marcus, and C. Nelson, "Categorization of facial expressions in children and adults: Establishing a larger stimulus set." *Cognitive Neuroscience Society*, 2002.

[36] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *Journal of Behavior Therapy and Experimental Psychiatry*, 1994.

[37] M. Tkalčič, U. Burnik, and A. Košir, "Using affective parameters in a content-based recommender system for images," *User Modeling and User-Adapted Interaction*, 2010.

[38] P. Arriaga and G. Almeida, "Fábrica de emoções : A eficácia da exposição a excertos de filmes na indução de emoções," Instituto Superior de Psicologia Aplicada, Tech. Rep., 2010.

[39] I. Bretherton and M. Beeghly, "Talking about internal states: The acquisition of an explicit theory of mind," *Developmental Psychology*, 1982.

[40] B. Fehr and J. Russell, "Concept of emotion viewed from a prototype perspective," *Journal of Experimental Psychology: General*, 1984.

[41] P. Shaver, J. Schwartz, D. Kirson, and C. O'Connor, "Emotion knowledge: Further exploration of a prototype approach." *Journal of Personality and Social Psychology*, 1987.

A Mobile Dietary and Emotional Diary System for Eating Disorder Care on the Smart Phone

In Jung Kim, HanZhong Zheng, Shi-Kuo Chang
Department of Computer Science
University of Pittsburgh, Pittsburgh, PA 15260, USA
{ink20, haz78, schang}@pitt.edu

***Abstract*—The advancement of the Internet of Things (IoT) and wireless sensors have paved the way for the development of new services for next-generation healthcare systems. Eating disorders are real, complex medical, psychiatric illnesses that can have serious consequences for health, productivity and personal relationships. We designed and implemented a mobile dietary and emotional diary system on the smart phone in order to detect eating disorder based on patients eating history and facial status detection. Through an analysis of normal and abnormal dietary input and emotional states, this system is intended for providing eating disorder healthcare service as a service in the cloud. The initial experimental results are presented and further research topics are discussed.**

Keywords—IoT, Eating disorder care, mobile dietary system, emotional diary system, slow intelligence system.

I. INTRODUCTION

The advancement of the Internet of Things (IoT) and wireless sensors has paved the way for the development of new services for next-generation healthcare systems to enable superior communication between healthcare professionals. However, in order to add value to raw sensor data we need to understand it by context aware mechanisms.

Context awareness from an IoT perspective introduces IoT paradigm and context-aware fundamentals with in-depth analysis of context life cycle. It evaluates 50 projects over last decade (2001-2011) [4]. There is a survey about techniques, methods, models, functionalities, systems, applications, middleware solutions related to context awareness and IoT. Context life cycle has 3 techniques which are context modelling techniques, context reasoning decision models, and context reasoning techniques. Context modelling techniques are key-value, markup schemes, graphical, object based, logic based, and ontology based modelling. Context reasoning decision models are decision tree, naive Bayes, hidden Markov models, support vector machines, k-nearest neighbor, and artificial neural networks. Lastly, context reasoning techniques are supervised learning, unsupervised learning, rules, fuzzy logic, ontological reasoning and probabilistic reasoning. From this survey, we have learned that development aids and practices has toolkits in general are suitable for limited scale application, middleware provide more functionality towards [1]managing data, standardization makes it easier to learn and use, and intelligibility toolkit helps faster adaptation of the users. Mobility, validity, and sharing are that IoT solutions need to track user movements, facilitate context-aware functionalities over different forms of devices and has different platforms, devices have different resource limitations, and different versions should be built on different devices. On Demand Data Modelling is data models need to be extensible on demand, and the ability to add knowledge when necessary is critical for wider adaptation. It also stores different types of context to help in a variety of situations. Hybrid Reasoning is multiple modelling and reasoning techniques can mitigate individual weaknesses using each other's strengths. For the hardware layer support, there are context awareness allows sensors to act more intelligently and save energy and significant amounts of energy can be saved by following fairly simple optimization. Dynamic configuration and extensions are pluggable rules that allow insertions when necessary, it is a major requirement as in IoT middleware applications, where domains and required knowledge cannot be predicted during the development stage. Distributed processing is real time processing and significant in the IoT, and cross domain context is queried to answer complex requirements.

There is the IoT for health care survey [5] which is about IoT based health care technologies, reviews network architecture and platforms, applications, industrial trends, and analyzes distinct IoT security and privacy features. From this survey, we can learn about what kind of IoT healthcare services and applications are possible, what kind of IoT healthcare products and prototypes are out there, and what kind of security requirements we can consider. Healthcare trends are ease of cost-effective interactions through seamless and secure connectivity across individual patients, clinics, and healthcare organizations is an important trend, and up-to-date healthcare networks driven by wireless technologies are expected to support chronic diseases, early diagnosis, real-time monitoring, and medical emergencies.

Enabling health monitoring as a service in the cloud [6] is wiki-health analysis framework that enables an ecosystem to support scientists, developers, and professionals to publish their data analysis models as utilities in the cloud and allow users to access those services and utilize their collected sensor data without any expert knowledge. Their ECG-based health monitoring service application deployed and wiki-health platform. There is Adaptive Learning Approach (ALA) which

[1] DOI reference number: 10.18293/DMSVIVA2018-022

reduces the training time while showing improved performance over existing methods.

IoT has a variety of application domains, including health care. However, enormous amount of raw sensor data need to be processed. As we describe above, there are many survey and researches about what kind of context modelling/reasoning, what health care application/services can be a target, what products/prototypes are out there, what security aspect we can consider, and ECG-based health monitoring service application trained by minimized data analysis.

We focus on treating eating disorder based on daily eating habit and knowledge base. However, it can be enhanced to other health disease care system such as asthma, cancer depression, anxiety, eating disorders, diabetes self-management, autism learning disorders, medication abuse, epilepsy or other seizure disorders, food allergies inflammatory bowel disease, obesity, organ transplant, sickle cell disease, and stuttering.

Eating disorders which are our primary focus are real, complex medical, and psychiatric illnesses that can have serious consequences for health, productivity, relationships. In the United States, 20 million women and 10 million men suffer from a clinically significant eating disorder at some time in their life. It includes anorexia nervosa, bulimia nervosa, binge eating disorder or EDNOS. Anyone can develop an eating disorder regardless of gender, age, race, ethnicity, culture, size, socioeconomic status or sexual orientation. Also, enabling eating disorders health care is a one of the behavioral health care we need to take care of. The treatment strategy is determined by the severity of illness and the specific eating disorder diagnosis, and cognitive-behavioral therapy and interpersonal therapy produce substantial and long-lasting changes and pharmacological treatment has often a useful role [8]. Therefore our mobile dietary and emotional diary system for eating disorder care will provide the long-lasting tracking system by writing the diary from the patient and checking the facial emotion to analyze the cognitive-behavioral.

On the other hands, to build E-health care system, it can be collaborated with healthcare professionals, regulators, pharmacies, insurance companies, vendors, hospitals, and patients [7]. Also, there are many challenges in E-health such as how the medical and healthcare information has been collected and stored, lack of technologies and potential cost to digitize the existing processes and tasks. Since one of the major challenge is to convert all of the medical and healthcare information to electronic format, we provide the healthcare system that the patients and professionals can easily write or update the information by using the smart phone.

In this paper, enabling health care service that could detect the eating disorders by 4 aspects. First, enable eating disorders health care as a service through the cloud and smartphone. Second, dynamic diet information from the cloud database. Third, real-time diet diary system on the smartphone to take care of behavioral health care. Lastly, emotion state recognizing can be confirm the patient facial status as needed. Section 2 describes the eating disorder diary system, and section 3 shows the emotion state recognizing. Then, abnormal state monitoring algorithm and equations are shown in section 4. Experiment and results explain in section 5 and we conclude our work with possible future work.

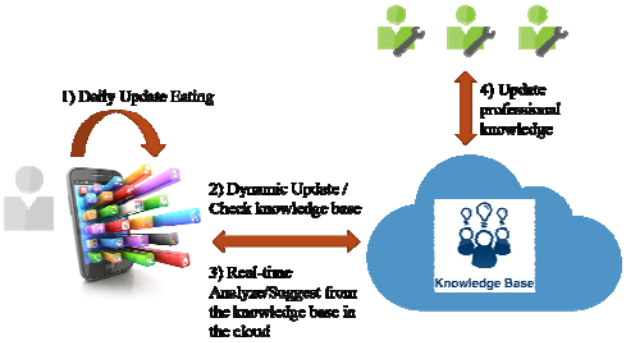


Fig. 1. Eating disorder diary system scenario

II. Eating Disorder Diary System

For the mobile dietary system, the eating disorder diary system has 3 major parts which are prevention, diagnosis, and treatment. Prevention aims to promote a healthy development before the occurrence of eating disorders. It also intends early identification of an eating disorder before it is too late to treat. Children as young as ages 5–7 are aware of the cultural messages regarding body image and dieting. Internet and modern technologies provide new opportunities for prevention. On-line programs have the potential to increase the use of prevention programs. The development and practice of prevention programs via on-line sources make it possible to reach a wide range of people at minimal cost. Such an approach can also make prevention programs to be sustainable. For the prevention, we design the professional suggestion part toward individual patents by building communication platforms for sharing information.

The diagnostic workup typically includes complete medical and psychosocial history and follows a rational and formulaic approach to the diagnosis. There are multiple medical conditions which may be misdiagnosed as a primary psychiatric disorder, complicating or delaying treatment. These may have a synergistic effect on conditions which mimic an eating disorder or on a properly diagnosed eating disorder. This typically involves counselling, a proper diet, a normal amount of exercise, and the reduction of efforts to eliminate food. Hospitalization is occasionally needed. Medications may be used to help with some of the associated symptoms. At five years about 70% of people with anorexia and 50% of people with bulimia recover. Recovery from binge eating disorder is less clear and estimated at 20% to 60%. These diagnosis and treatment can be solved by depression recognition, monitoring psychological status, machine learning to make better diagnosis, real-time diary system, positive feedback to encourage having healthier eating habits, and so on.

Figure 1 shows the eating disorder diary system scenario. Whenever the patent writes the diary to update the eating history, the system will dynamically update and check knowledge base which is located in the cloud database.

Meanwhile, real-time analysis and suggestion mechanism from the knowledge base in the cloud come to the eating

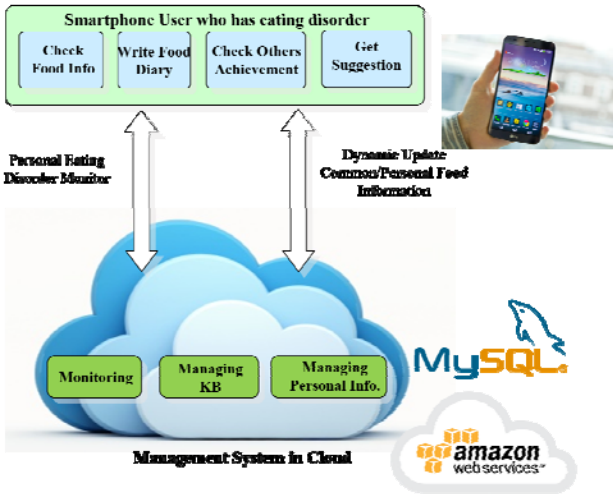


Fig. 2. Eating disorder architecture on Android platform and MySQL database on the cloud environment

disorder diary system on the smart phone. On the other hand, all the patients' information will be uploaded to the professional people to get the individual feedback.

Figure 2 explains the eating disorder diary architecture. On the smart phone side, there are 4 components which are checking food information, writing food diary, checking others achievement, and getting professional suggestion for the mobile dietary app which is implemented on Android platform. These components are used to monitor the personal eating disorder as well as dynamically update the common and personal food information to the management system in the cloud. In the cloud server side, there are 3 components which are monitoring, managing knowledge base, and managing personal information. All these information is implemented on the top of MySQL and Amason Web Service cloud servers.

III. Emotion State Recognizing

The user's emotional information can be collected through the emotional diary system. The emotional diary system is a smart phone app implemented on iOS platform, which utilizes the smartphone camera to detect user's facial expression. Human facial expression can be sensed and analyzed through "Affdex SDK", which is distributed as CocoaPod. "Affdex SDK" provides open source API support for Swift projects in the task of detecting different emotional expressions such as joy, disgust, surprise, etc. [1]. Through the camera, the emotional diary system is able to localize the key facial landmarks and capture the user's facial expressions.

The emotional diary system is able to recognize 7 different emotional expressions of the user. Each emotion is represented by a numerical value expressed in the progress bar. After the user confirms the most representative emotion, a detailed facial expression report will be generated and are available for user to

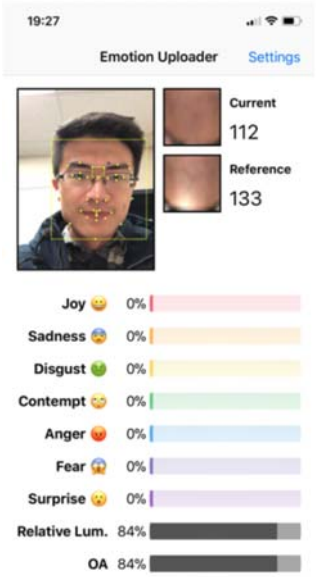


Fig. 3. A sample user-interface of emotion diary app on iOS platform.

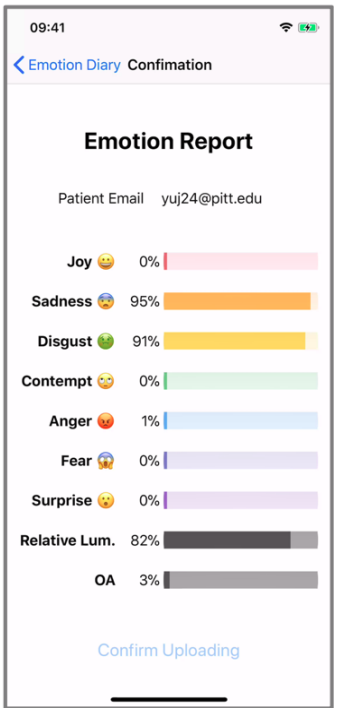


Fig. 4. A sample report generated by emotion diary app

read (shown in figure 3 and 4).

For this project, we are mainly interested in the most three representative different emotions: joy, disgust, sadness to

detect the abnormal amount of calories intake and track the care procedure for eating disorder.

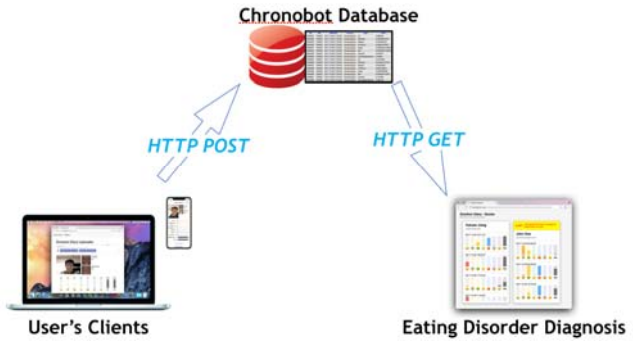


Fig. 5. Emotional diary system architecture

Input: D_i
for *Centroid* Φ_j *in all Centroids* **do**
  calculate the distance $d(D_i, \Phi_j)$
  if $(d(D_i, \Phi_j) == min)$ **then**
    $c(D_i) = \Phi_j$
  end
  add D_i into the data instances
  update the Φ_j
end
if $\Psi(c(D_i)) == e_1$ **then**
  return *completely normal event*
end
if $\Psi(c(D_i)) == e_2$ **then**
  return *normal event*
end
if $\Psi(c(D_i)) == e_3$ **then**
  return *slightly abnormal event*
end
if $\Psi(c(D_i)) == e_4$ **then**
  return *abnormal event*
end
Output:$\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$

Algorithm 1: algorithm shows the procedure of detecting the event types

Figure 5 demonstrates the architecture of the emotional diary system. The system can be divided into three components interacting with each other. A mobile component (smartphones, personal computer) collect user's emotions at different time stamps and uploads the emotions into Chronobot MySQL Database of a Mobile Slow Intelligence System [2]. Then, eating disorder diagnosis component sends the request to the database to continuously monitor the emotions for each user. The emotion state can be one of the criteria for evaluating the caring procedure of eating disorder.

IV. Abnormal State Monitoring

The goal of eating disorder care is to identify the abnormal events in the data. The system uses the data-driven model to identify the 3 different events in the data: {abnormal-event, almost-abnormal event, almost normal event, normal event}, also denoted as {e1, e2, e3, e4} The basic idea of abnormal event identification is to "customize" for each user, that is the definition of the abnormal event can be automatically adjusted with respect to users.

In the system, the problem of identifying the abnormal events can be seem as the problem of clustering with three centroids (Φ1, Φ2, Φ3, Φ4). Each centroid is equivalent to different type of event.

Definition 1: For the new data instance D_i, the Euclidean distance between the data instance D_i and the three centroids (Φ1, Φ2, Φ3) is denoted as $d(D_i, \Phi1)$, $d(D_i, \Phi2)$, $d(D_i, \Phi3)$; $C(D_i)$ is cluster of the data instance D_i: If $C(D_i) = \Phi1$, then $\Psi(D_i) = e1$. If $C(D_i) = \Phi2$, then $\Psi(D_i) = e2$. If $C(D_i) = \Phi3$, then $\Psi(D_i) = e_3$. If $C(D_i) = \Phi_4$, then $\Psi(D_i) = e_4$.

Algorithm 1 shows the process of the system detects the abnormal event with the data instance Di as the input and the types of the event as the output. The study of the clustering is data-driven. However, during the initial phase of the system, identifying the centroid may be a problem, due to the lack of data instances. We assume the initial three centroids are (μ-2σ, μ, μ+2σ), where μ = 2,200 calories, σ = 200 calories, following with the Gaussian distribution, based on the average amount of calories in taking on daily basis recommended by Dietary Guidelines Advisory Committee (DGAC) [3]. Clustering the new data instance D_i with d dimensions and updating the new centroids of the cluster are implemented through k-means algorithm illustrated in Equation 1 and 2, which enables the definition of the abnormal event gradually adjusted to each user.

$$argmin_{j \in \{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}} dis(\mathbf{D}_i^{|d|}, j) (1)$$

Equation 1: identifying the cluster for the new data instance Di

$$\Phi_j = \frac{1}{|D_j|} * \sum_{D_{i,j}}^{|D_j|} \mathbf{D}_{i,j}^{|d|} (2)$$

Equation 2: updating the centroid based on the new data instance Di

The attributes in the data instances consist of continuous values (e.g. calories amount) and categorical values (e.g. emotions). The distance calculation between the Di to the centroids cannot be purely the calculation of Euclidean distance, which does not work well for categorical values. We separate the attributes of the data instance into two group ε and η, where ε is the set of attributes with continuous values and η is the set of attributes with categorical values. The difference between two ε: ε1 and ε2 and be simply calculated with distance function (e.g. Euclidean distance, city block, etc.). For η, the difference of the two η: η1 and η2 can be computed through a kernel function that calculates the dissimilarity scores between η1 and η2.

$$dist(D_i, \Phi_j) = \alpha_\epsilon * \lambda(D_i, \Phi_j) + \beta_\eta * \kappa(D_i, \Phi_j) (3)$$

Equation 3: the distance measurement between the data instance D_i and the centroid Φj

The λ is the normal distance calculation function and κ is the kernel function that used to measure the dissimilarities. The α and β are the weights assigned to each distance measurement.

Using this approach, it can take all attributes into account for the distance measurement, while avoiding the inference caused

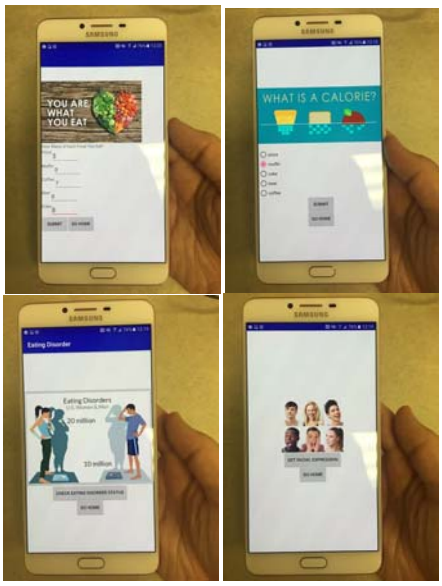


Fig. 6. Dietary System: updating dietary history (left-top), checking food information(right-top), checking eating disorder status(left-bottom), getting emotion status identification (right-bottom)

by categorical values. Besides, the value of α and β can dynamically change and be learned through continuous data stream collected from user in order to better detect the abnormalities.

V. Experimental Results

The application was implemented on iOS and Android platforms to collect users' emotional states and dietary information. All the records were stored in Chronobot MySQL database. We focus on identifying the abnormal events with incoming data stream collected from the applications. Table 1 shows the sample data stored and extracted from Chronobot that we used for analyzing the abnormal events. From the table, we can extract the total amount of calories in taking on daily basis associating with its corresponding emotional state. The total amount of calories and emotional state are the two criteria to evaluate the abnormality of the events.

Through the mobile dietary system on the top of Android platform, we can upload our dietary history several time per day, check food information such as calories, check eating disorder status among normal / almost normal / almost abnormal / abnormal, and get emotion status identification which is extracted from iOS platform as shown in figure 6.

We conducted a user-study experiment for a period of 15 days with 5 different patients. During the experiment, we kept track of users' daily food consumption and calculated the total amount of calories in taking based on the food types and amount, while recording user's emotional expression. Then to better projection the abnormalities of the events, we normalized the event abnormality values into the interval [0, 1] and classify the event based on its event abnormality values. The higher value indicates the higher degree of abnormality of the event.

User ID	Date	Emotion State	Total Calories	Food Types
11111	2018-03-21	Joy	2,013	Apple salad, milk, turkey sandwich, wheat bagel
11111	2018-03-25	Sadness	1,130	Apple, milk, salad
2222	2018-03-29	Disgust	2,514	Fried chicken, brownie, potato fries, burger
2222	2018-03-30	Disgust	2,201	Egg, ham, sea food, soup, fried rice

Table 1: A sample data stored in the Chronobot database

User ID	Date	Emotion State	Total Calories	Abnor-mality Value	Classifica-tion
11111	2018-04-21	Joy	2,013	0.0	Normal (blue)
22222	2018-04-25	Sadness	1,130	1.0	Abnormal (red)
33333	2018-04-29	Disgust	2,514	0.7	Almost abnormal (orange)
44444	2018-04-30	Disgust	2,201	0.3	Almost Normal (green)
55555	2018-04-21	Sadness	2197	0.3	Almost Normal (green)

Table 2: A simple table visualization of abnormal event identification

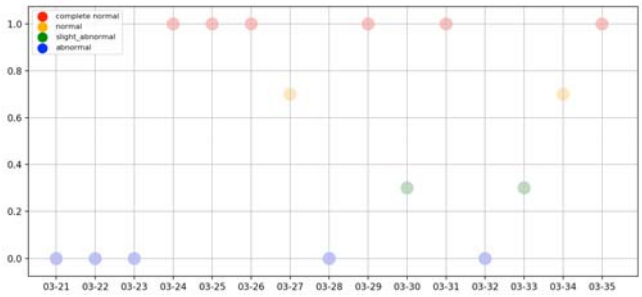


Fig. 7. A sample graph visualization of abnormal event identification

The types of the event also map to different color. Figure 7 and table 2 are the two different the visualizations of the types of events identified by our system. Event set {abnormal-event, almost-abnormal event, almost normal event, normal event} map to the color set (red, orange, green, blue) to allow users, physicians be aware of the abnormalities inside the data, or evaluate the caring process of eating disorder. In Table 2, it demonstrates a sample for 5 different users who accepted the

test. Figure 7 is a sample visualization of a patient over the testing period.

VI. Conclusion and Future Work

Eating disorders are real, complex medical, psychiatric illnesses that can have serious consequences for health, productivity and personal relationships. We designed and implemented a mobile dietary and emotional diary system on the smart phone in order to detect eating disorder based on patients eating history and facial status detection. Through an analysis of normal and abnormal dietary input and emotional state, this system is intended for providing eating disorder healthcare service as a service in the cloud. In the future, we can check how social media feature will impact the human behavior for their healthcare and do some benchmark depends on cloud resource location compared with smartphone location. We can also build software engineering modeling with cloud architecture for the slow intelligence system and professional medical users' interface to collect recommendation. The reason why we build the dietary system and emotional diary system on the Android and iOS separately is to show the diversity, however, we can build into the one platform as well. Also, sensor fusion feature is possible by combining food information with location information, detecting user lie, reducing user's behavior by detecting order voice, location, camera or so. Our cloud environment can be more cost effective cloud resource usage by bidding resource, and it will be more effective when the medical information is getting massive in the real world since it can be scalable on demand. In the future, we also plan to increase the number of criteria that were used to determine the abnormalities of the event. For example, we could consider the patient's daily exercise and the types of the food. Even though a patient's daily total calories are high, he/her has a very active daily exercises, the system will still consider it is normal. Although a patient's daily calories are low, the food he/she ate are most junk food, and then the system still consider it is an abnormal event. We also plan to use different decision making methods such as building a neural network and decision tree.

Acknowledgement

The Emotion Diary was first conceived and implemented by Yuhuan Jiang.

References

[1] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. 2016. AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16). ACM, New York, NY, USA, 3723-3726. DOI: https://doi.org/10.1145/2851581.2890247.

[2] Shi-Kuo Chang, Wei Guo, Duncan Yung, ZiNan Zhang, HaoRan Zhang and WenBin You, A Mobile TDR System for Smart Phones, DMSVLSS 2017, Wyndyam Pittsburgh University Center, Pittsburgh, PA, USA, 75-85.

[3] Technical tables for the "Typical" rather than "Nutrientdense" food choices analysis, completed for the 2010 Dietary Guidelines Advisory Committee. Addenda to Britten et al., Impact of typical rather than nutrient-dense food choices in the US Department of Agriculture Food Patterns, JAND 112(10): 1560-1569, 2012

[4] Context Aware Computing for The Internet of Things, IEEE Communication survey & tutorials, 2013

[5] The Internet of Things for Health Care, IEEE Access, 2015

[6] Enabling Health Monitoring as a Service in the Cloud, IEEE/ACM 7th International Conference on Utility & Cloud Computing, 2014

[7] Upkar Varshney, "Pervasive Healthcare Computing", Springer Publisher

[8] Halmi, KA, "The multimodal treatment of eating disorders", (2005), World Psychiatry. World Psychiatric Association

Event-Based Data Input, Modeling and Analysis for Meditation Tracking using TDR System

Shi-Kuo Chang[1], CuiLing Chen[2], Wei Guo[1] and NanNan Wen[1]
[1]**School of Computing and Information**
University of Pittsburgh, Pittsburgh, PA 15238, USA
{schang, weg21, naw66}@pitt.edu

[2]**College of Mathematics and Statistics, Guangxi Normal University, Guilin 541004, PR China**
mathchen@163.com

Abstract

In this paper we describe an experimental TDR system with continuous data input from devices such as smart phones and sensors such as brain wave headsets. We developed event-based data input, modeling and analysis techniques in order to analyze input data and track progress of meditation. Initial experimental results indicate that this approach is quite promising.

Keywords

Meditation tracking, event-based data input, modeling and analysis, slow intelligence system, TDR system.

1 Introduction

In our previous work we developed the TDR system, which is a multi-level slow intelligence system with interacting super-components each of which has its own computation cycle [1], as a platform to explore applications in personal health care, emergency management, social networks and so on. In this paper we apply the TDR system to event-based data analysis and visualization for meditation tracking.

Meditation, defined as "the attention inwards towards the subtler levels of a thought until the mind transcends the experience of the subtlest state of the thought and arrives at the source of the thought", has been proven to have positive effects on social skills, feeling of compassion, self-management, somatic awareness and mental flexibility. It has also been used in treatment of anxiety disorders, stress reduction, chronic pain, persistent pain, depression, autism spectrum disorders, traumatic experiences, acquired brain injury, and even eating disorder, psoriasis and substance abuse.

Nowadays many people are learning meditation. However there are still no adequate meditation monitoring systems to take continuous measurements from various sensors when a person is in meditation and to track its progress. In this paper we describe an experimental TDR system with continuous data input from devices such as smart phones and sensors such as brain wave headsets. We developed event-based data input, modeling and analysis techniques in order to analyze input data and track progress of meditation. Initial experimental results demonstrate that this approach is quite promising.

The paper is organized as follows. In Section 2 we describe the system architecture. The interface to support event-based data input is presented in Section 3. Event-based data modeling is described in Section 4, followed by a detailed example of data analysis presented in Section 5. Section 6 presents user scenarios for the experimental system. Discussion and conclusion are presented in Section 7.

2 System Architecture

Figure 1 illustrates the experimental TDR system consisting of interacting super-components and the chronobot database. Each super-component has its own computational cycles. The super-components interact with one another through the SIS server. Based on requests from the administrator, the super-components process input data and upload them to

DOI reference number: 10.18293/DMSVIVA2018-005

the Chronobot database. In the TDR system there are at least three super-components: Tien (Heaven), Di (Earth) and Ren (Human). The Tien super-component handles sensors for the atmosphere, the Di super-component deals with sensors for the lithosphere and the Ren super-component manages sensors for the human body.

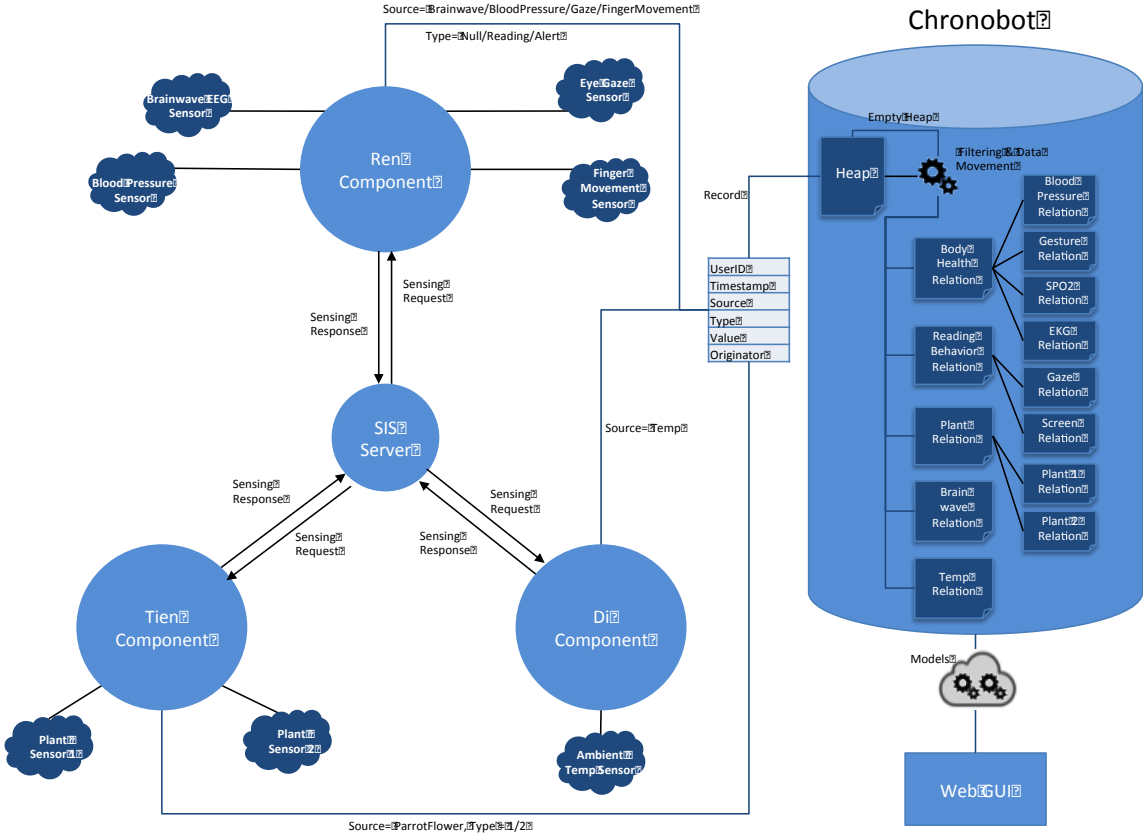


Figure 1. TDR System with super-components and Chronobot database.

In order to track meditation we proposed to use brain wave headset as well as eye gaze tracking by smart phone during meditation [2]. Data from brain wave sensor and eye gaze tracker are collected by their respective input processors in Ren super-component and uploaded to the Heap relation in the Chronobot database. The Heap is a collection of records each with a variable number of attributes for different types of sensor data, which are filtered and moved into different relations such as Gazing Behavior Relation, Brain Wave Relation and so on, by the request of the administrator through the Web GUI.

A more detailed view is shown in Figure 2. Records in the Heap are first filtered and then moved to the corresponding relations. In the filtering of data, the resultant data must conform to the model for the corresponding relation. We will first explain the conceptual framework. The detailed formal model will be presented in Section 4.

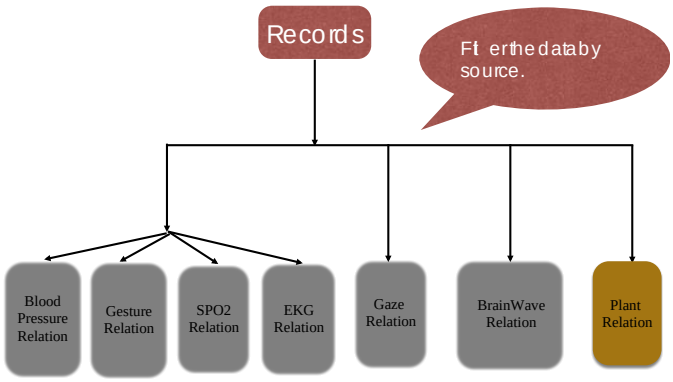


Figure 2. Records are filtered and moved to the corresponding relations.

3. Event-Based Data Input

The database for the TDR system is a time varying database. To make sense of the time varying database, we need to monitor the data streams and detect significant changes. For the best of our knowledge, there's few researches on designing user interface for time varying database. User interface design requires a good understanding of user needs [3], in our approach we need to be able to specify what is normal and what is not normal. In fact, a database is governed by a data model specifying what is the normal pattern. The computation cycles specify the collection, filtering and storage of data that conform to the normal pattern, so the end result is a **normal event**. The cycle then repeats itself. When the data deviates from the normal pattern, it is an **abnormal event** to take notice of. Our approach to user interface design is therefore based upon this concept of normal and abnormal events.

In recent years, visualization has become an important tool to support exploration and analysis of large volumes of data. Therefore, to shift the needs of users into the focus, we should pursue an event-based approach to visualization. This approach allows users to specify their interests as event types. The **normal event** is the data model. The **abnormal event** is what deviates from the data model.

During a computation cycle, the normal event is usually the end result, i.e. the processing and storage of data that conforms to the specified data model. When instances of the specified abnormal event types are detected, the user interface automatically adjusts visual representations according to the detected event instances. This approach results in visualizations that are adapted to the needs and interests of the users. Hence, users are supported in achieving their task at hand.

In terms of event-based visualization, the basic idea is to let users specify their interest by means of event types, to detect instances of these events in the data, and to create representations that can be automatically adjusted with respect to the detected event instances. Accordingly, three main aspects are investigated:.

1. Event specification,
2. Event detection,
3. Event representation.

To bridge the gap between informal user interests and the digital language of computers, a formalism for the event specification must be developed. Here, the difficulty is to build a formal basis that provides a suitable expressiveness while still allowing users to specify their interests as easily as possible. Especially when facing users who are not familiar with event-based visualization, it is essential to provide methods and tools that allow an intuitive specification of event types.

The task of the event specification is to compile event types that are or might be of interest to visualization users. The event specification necessitates a formal foundation to allow a later detection of event instances.

In our approach, we have two types of events: normal events that represent the data model, and abnormal events that represent deviation from the data model. Events are always specified for a certain relation. Before moving data from heap to relation, we first check if the tuple satisfies a certain event type.

(1) **Normal Event:** As an example, if every tuple has an error rate less than the threshold ε (for example ε = 0.1), then it is a normal event. This event can be described as:

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) \le \varepsilon, \quad \text{..... (1)}$$

Once a normal event is specified, we can create a computation cycle to get the TDR system started. For formal definition, see Section 4.

(2) **Abnormal Event 1:** As an example, for three consecutive tuples, if each tuple's error rate exceeds the threshold ε, then it is an abnormal event. This event can be described as:

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) > \varepsilon,$$
$$\Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})) > \varepsilon, \quad \text{..... .... (2)}$$
$$\Psi(\Delta T(\eta), X(v_{I+2}), Y(v_{I+2})) > \varepsilon,$$

For formal definition, see section 4. If condition (2) is met, user then can use the following steps to specify this event.:

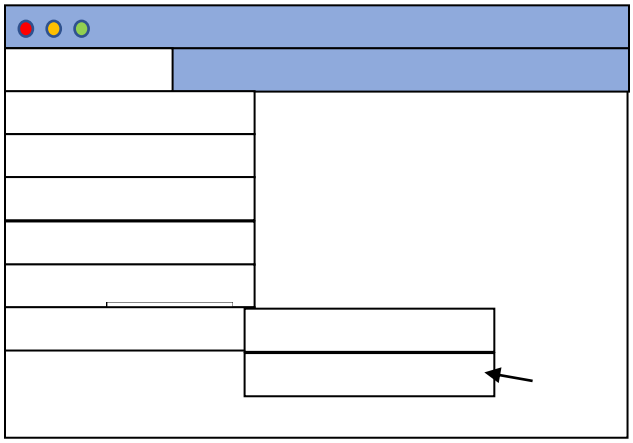


Figure 3. Choose Event Type for a certain relation.

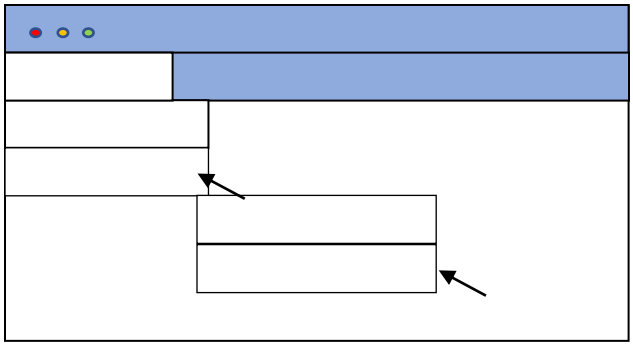


Figure 4. Click on Abnormal Event and then Independent Event.

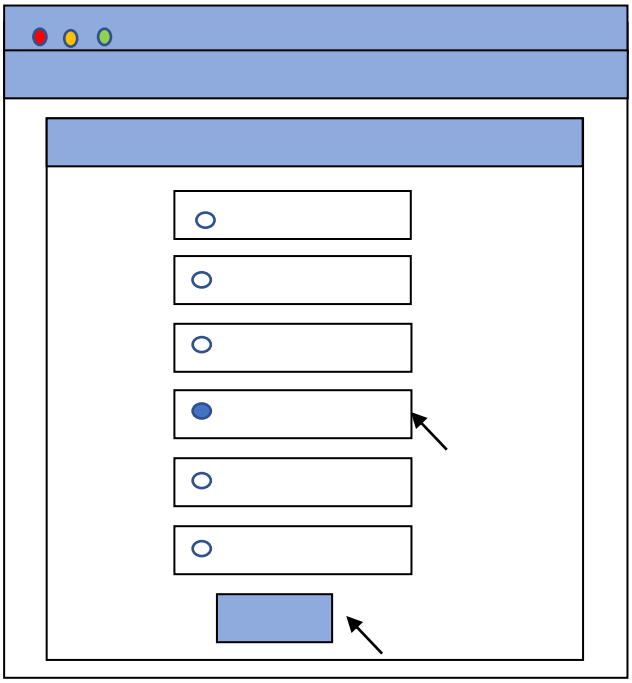


Figure 5. Choose tuple and submit event.

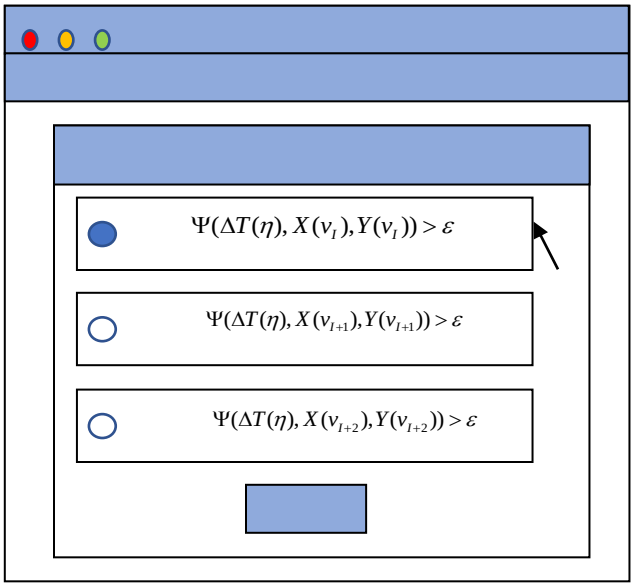


Figure 6. Choose event type.

(3) **Abnormal Event 2:** For three consecutive tuples, if each tuple's error rate is twice as much as the previous tuple, then it is an event. This event can be described as:

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) > \varepsilon \quad \text{..... (3)}$$

$$\Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})) > 2 \cdot \Psi(\Delta T(\eta), X(v_I), Y(v_I))$$

$$\Psi(\Delta T(\eta), X(v_{I+2}), Y(v_{I+2})) > 2 \cdot \Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1}))$$

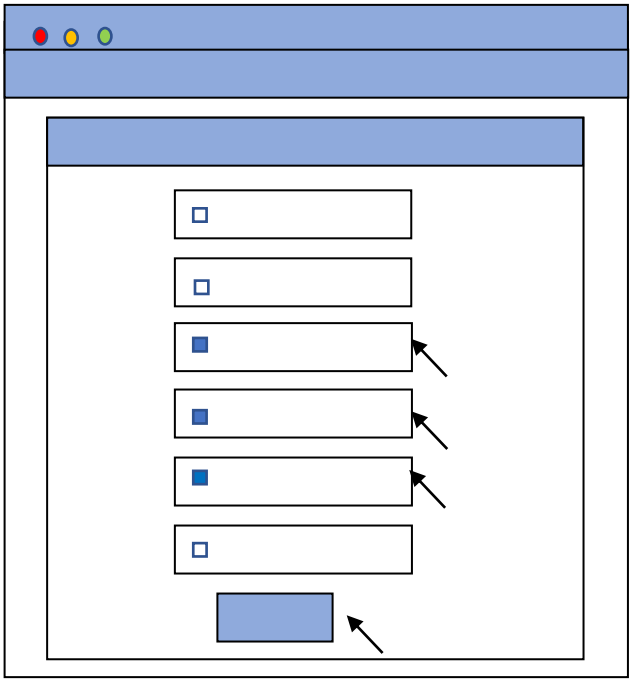


Figure 7. Choose 3 tuples and submit.

The user can click on Abnormal Event and then Dependent Event, similar to Figure 4. For formal definition, see section 4. If condition (3) is met, user then can use the steps in Figure 7 and Figure 8

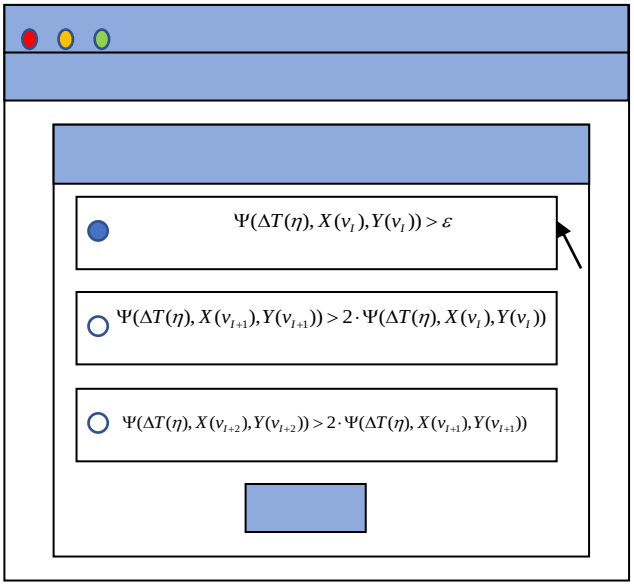


Figure 8. Choose event type.

to specify this event.

4. Event-Based Data Modeling

Inspired by [4] and [5], we consider a multimedia database with time-varying $R(T, A_1, \cdots, A_n)$, where T denotes time, $A_1, \cdots, A_n$ are other attributes, v_i is the tuple corresponding to the moment t_i, U is a set of the attributes $A_1, \cdots, A_n$, and $dom(A_k)$ is the domain of each A_k, $v_i[A_k]$ denotes the value of the tuple v_i in the attribute A_k. Thus for any two moments t_i and t_j of T in R, there are always a pair of corresponding tuples

$v_i = (t_i, v_i[A_1], \cdots, v_i[A_n])$, $v_j = (t_j, v_j[A_1], \cdots, v_j[A_n])$.

The similarity between any two attribute values $v_i[A_k]$ and $v_j[A_k]$ of A_k is based on a distance function of type $d : dom(A_k)^2 \to [0,1]$ [4]. For simplicity, we denote with $D(A_k)$ the set of the distance functions defined on A_k.

According to the distance function of type $d : dom(A_k)^2 \to [0,1]$, definition 1 is given as follows, then from which we get definition 2.

Definition 1 Given a relation $R(T, A_1, \cdots, A_n)$, for a pair of tuples v_i and v_j corresponding to any two moments t_i and t_j of T, we say that v_i is **similar** within τ to v_j with respect to d at the moments t_i and t_j, denoted with $\boldsymbol{v_i[A_k] \cong_{(d,\tau,t_i,t_j)} v_j[A_k]}$, iff $\boldsymbol{d(v_i[A_k], v_j[A_k]) \le \tau}$, where τ is a threshold.

Definition 2 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, we say the **type-M function dependency** during the time of T (**T-MFD**): $\boldsymbol{X_{(d_1,\tau')} \xrightarrow{T} Y_{(d_2,\tau'')}}$ holds, if and only if for a pair of tuples v_i and v_j corresponding to any two moments t_i and t_j of T, whenever $v_i[X] \cong_{(d_1,\tau',t_i,t_j)} v_j[X]$, then $v_i[Y] \cong_{(d_2,\tau'',t_i,t_j)} v_j[Y]$, where $d_1 \in D[X]$, $d_2 \in D[Y]$, $\tau', \tau'' \in [0,1]$ are thresholds.

Obviously, given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, sometimes there are a pair of tuples v_I and v_J corresponding to some two moments t_I and t_J of T such that $v_I[X] \cong_{(d_1,\tau',t_I,t_J)} v_J[X]$ holds, whereas $v_I[Y] \cong_{(d_2,\tau'',t_I,t_J)} v_J[Y]$ doesn't hold. Then the following definition is necessary.

Definition 3 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, if $v_I[X] \cong_{(d_1,\tau',t_I,t_J)} v_J[X]$ holds, whereas $v_I[Y] \cong_{(d_2,\tau'',t_I,t_J)} v_J[Y]$ doesn't hold, we say v_I, v_J at the moments t_I and t_J with respect to X,Y constitute a **dependency violation event** (**DVE**) of T, denote by **T-DVE -v_I, v_J[X, Y]**. The DVEs of all tuples during the time of T with respect to X,Y are denoted as **T-DVEs -X,Y**.

Thus the occurrence rate of T-DVEs - X,Y with respect to any two attributes X,Y is a very important problem that one concerns, which can be calculated by the following definition.

Definition 4 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, we define the **dependency**

violation rate (**DVR**) of X,Y during the time of T (**T-DVR -*X,Y***) as follows:

$$\Psi(T,X,Y)=\frac{r_1}{r_T},$$

where r_T, $r_1 \subseteq r_T$ denote the combinatorial number of any pair of attributes in X or Y during the time of T, the number of the T-DVEs -X,Y, respectively.

We know if there are T-DVEs -X,Y, and the T-DVR -X,Y is very small, even very close to zero, then $X_{(d_1,\tau')} \xrightarrow{T} Y_{(d_2,\tau'')}$ almost holds, from which the following generalized T-MFD definition is yielded.

Definition 5 Given a relation $R(T,A_1,\cdots,A_n)$ and $X,Y \subseteq U$, we say the **relaxed type-M function dependency** during the time of T (**T-RMFD**): $X_{(d_1,\tau')} \xrightarrow{\Psi(T,X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$ holds, if and only if for a pair of tuples v_i and v_j corresponding to any two moments t_i and t_j of T, whenever $v_i[X] \cong_{(d_1,\tau',t_i,t_j)} v_j[X]$, then almost $v_i[Y] \cong_{(d_2,\tau'',t_i,t_j)} v_j[Y]$ holds, and $\Psi(T,X,Y)\le\varepsilon$, where $\Psi(T,X,Y)$ is the T-DVR -X,Y, $d_1 \in D[X]$, $d_2 \in D[Y]$, and $\tau',\tau'',\varepsilon \in [0,1]$ are thresholds.

Remark 1: It is depended on the value of ε to a great degree whether $X_{(d_1,\tau')} \xrightarrow{\Psi(T,X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$ holds.

For a relation $R(T,A_1,\cdots,A_n)$ and $X,Y \subseteq U$, when T is too long and there are too many data during the whole time T, we can consider to investigate fewer data during a part time. If we use the symbol η to denote the duration of the part time, and get the following definition.

Definition 6 Given a relation $R(T,A_1,\cdots,A_n)$ and $X,Y \subseteq U$, we say the **relaxed type-M function dependency** during $\Delta T(\eta)$ ($\Delta T(\eta)$ - **RMFD**) :

$$X_{(d_1,\tau')} \xrightarrow{\Psi(\Delta T(\eta),X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$$

holds, if and only if for a pair of tuples v_i and v_j corresponding to any two moments t_i and t_j during $\Delta T(\eta)$ (i.e., $\Delta T = \max\limits_{M_1\le i,j\le M_2, i\ne j} |t_i - t_j| \le \eta$, $M_1, M_2 \in \{1,2,\cdots,m\}$), whenever $v_i[X] \cong_{(d_1,\tau',t_i,t_j)} v_j[X]$ holds, there is almost $v_i[Y] \cong_{(d_2,\tau'',t_i,t_j)} v_j[Y]$ holds, and

$$\Psi(\Delta T(\eta),X,Y)=\frac{r_2}{r_{\Delta T(\eta)}} \le \varepsilon,$$

where $r_{\Delta T(\eta)}$ is the combinatorial number of any pair of attributes in X or Y during $\Delta T(\eta)$, $r_2 \subseteq r_{\Delta T(\eta)}$ is the number of the DVEs $-X,Y$ during $\Delta T(\eta)$ (***ΔT(η)* - DVEs -*X,Y***), m is the number of the tuples during the whole time T, $d_1 \in D[X]$, $d_2 \in D[Y]$, and $\tau',\tau'',\varepsilon \in [0,1]$ are thresholds.

Remark 2: Similar to definition 4, we can say $\Psi(\Delta T(\eta),X,Y)$ in definition 6 is the dependency violation rate of X,Y during $\Delta T(\eta)$ (***ΔT(η)* - DVR -*X,Y***).

For a relation $R(T,A_1,\cdots,A_n)$ and $X,Y \subseteq U$, if we investigate the data during some part time of T and can get the relation between X and Y during the whole time T, then we only need to consider the relation $R(T,A_1,\cdots,A_n)$ during this part time. The following definition describes this case.

Definition 7 Given a relation $R(T,A_1,\cdots,A_n)$ and $X,Y \subseteq U$, if

$$T=\Delta T_1(\eta)\cup\cdots\cup\Delta T_L(\eta),\ \Delta T_i(\eta)\cap\Delta T_j(\eta)=\Phi$$
$$(i,j=1,2,\cdots,L,\ i\ne j),$$

where $\Delta T_i(\eta)$ denotes $\max\limits_{j\ne k}|t_j - t_k| \le \eta$ ($j,k \in \{1,2,\cdots,L\}$), and $\Psi(\Delta T_i(\eta),X,Y) \le \varepsilon$ $(i=1,2,\cdots,L)$ holds, we use $\Psi(\Delta T_I(\eta),X,Y)$ to denote $\min\limits_{1\le i\le L}\{\Psi(\Delta T_i(\eta),X,Y)\}$. Thus the **RMFD** of X, Y during $\Delta T_I(\eta)$ can be expressed as $X_{(d_1,\tau')} \xrightarrow{\Psi(\Delta T_I(\eta),X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$.

Remark 3: Under the conditions of definition 7, we can get $X_{(d_1,\tau')} \xrightarrow{\Psi(T,X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$ by $X_{(d_1,\tau')} \xrightarrow{\Psi(\Delta T_I(\eta),X,Y)\le\varepsilon} Y_{(d_2,\tau'')}$.

Based on definition 3 and definition 6, it is easy to know there are two classes of dependency

violation events during $\Delta T(\eta)$, so we summarize as follows.

Definition 8 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for a pair of tuples v_I and v_J corresponding to some two moments t_I and t_J during $\Delta T(\eta)$, if there is one of the following cases happening, we say v_I, v_J at the moments t_I and t_J with respect to X, Y constitute a **dependency violation event** during $\Delta T(\eta)$ ($\Delta T(\eta)$-**DVE**):

A $|t_I - t_J| > \eta$;

B $v_I[X] \cong_{(d_1, \tau', t_i, t_j)} v_J[X]$ holds, whereas $v_I[Y] \cong_{(d_2, \tau'', t_i, t_j)} v_J[Y]$ doesn't hold.

For simplicity, Case (1) is denoted as $\Delta T(\eta)$ - **DVE** -t_I, t_J, Case (2) is denoted as $\Delta T(\eta)$ - **DVE** -v_I, v_J **[X, Y]**.

For a tuple v_I corresponding to some moment t_I during $\Delta T(\eta)$, sometimes we need to know the **DVR** of v_I. To this end we need to introduce the definition of **DVE** of v_I. According to definition 3, we present the following definition.

Definition 9 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for some tuple v_I corresponding to some moment t_I and a series of tuples v_j corresponding to some moments t_j during $\Delta T(\eta)$, if $v_I[X] \cong_{(d_1, \tau', t_I, t_j)} v_j[X]$ holds, whereas $v_I[Y] \cong_{(d_2, \tau'', t_I, t_j)} v_j[Y]$ doesn't hold $(1 \le I, j \le m_1 \text{È} I \ne j)$, where m_1 is the number of the tuples during $\Delta T(\eta)$, we say v_I, v_j at the moments t_I and t_j with respect to X,Y constitute a **dependency violation event** during $\Delta T(\eta)$, denote by $\Delta T(\eta)$ - **DVE** -v_I, v_j **[X, Y]**. All of the DVEs of v_I with respect to X,Y during $\Delta T(\eta)$ are denoted as $\Delta T(\eta)$ - **DVEs** -v_I **[X, Y]**.

Based on definition 9, we can get definition 10.

Definition 10 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for the tuple v_I corresponding to some moment t_I during $\Delta T(\eta)$, we define the **dependency violation rate** of v_I during $\Delta T(\eta)$ ($\Delta T(\eta)$ - *DVR* - v_I**[X, Y]**) as follows:

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) = \frac{r_{v_I}}{m_1 - 1},$$

where $r_{v_I} \subseteq r_{\Delta T(\eta)}$ is the number of $\Delta T(\eta)$ - DVEs -v_I[X, Y], m_1 is the number of the tuples during $\Delta T(\eta)$.

For a tuple v_I during $\Delta T(\eta)$ and a given ε, if $\Psi(\Delta T(\eta), X(v_I), Y(v_I)) \le \varepsilon$, then we say v_I constitutes a **normal event** (**NE**) (see section 3). Otherwise we say it is an **abnormal event** (**ANE**). In particular, we study the following cases.

Definition 11 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for the three consecutive tuples v_I, v_{I+1}, v_{I+2} corresponding to some moments t_I, t_{I+1}, t_{I+2} during $\Delta T(\eta)$, if

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) > \varepsilon,$$

$$\Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})) > \varepsilon,$$

$$\Psi(\Delta T(\eta), X(v_{I+2}), Y(v_{I+2})) > \varepsilon,$$

we say the tuples v_I, v_{I+1}, v_{I+2} constitute an **abnormal event 1** during $\Delta T(\eta)$ ($\Delta T(\eta)$ - **ANE -1)** (see section 3) .

Definition 12 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for the three consecutive tuples v_I, v_{I+1}, v_{I+2} corresponding to some moments t_I, t_{I+1}, t_{I+2} during $\Delta T(\eta)$, if

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) > \varepsilon,$$

$$\Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})) > 2 \cdot \Psi(\Delta T(\eta), X(v_I), Y(v_I)),$$

$$\Psi(\Delta T(\eta), X(v_{I+2}), Y(v_{I+2})) > 2 \cdot \Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})),$$

we say the tuples v_I, v_{I+1}, v_{I+2} constitute an **abnormal event 2** during $\Delta T(\eta)$ ($\Delta T(\eta)$ - **ANE-2)** (see section 3) .

More generally, we have the following case.

Definition 13 Given a relation $R(T, A_1, \cdots, A_n)$ and $X, Y \subseteq U$, for the three consecutive tuples v_I, v_{I+1}, v_{I+2} corresponding to some moments t_I, t_{I+1}, t_{I+2} during $\Delta T(\eta)$, if

$$\Psi(\Delta T(\eta), X(v_I), Y(v_I)) > \varepsilon,$$

$$\Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1})) > n \cdot \Psi(\Delta T(\eta), X(v_I), Y(v_I)),$$

$\Psi(\Delta T(\eta), X(v_{I+2}), Y(v_{I+2})) > n \cdot \Psi(\Delta T(\eta), X(v_{I+1}), Y(v_{I+1}))$, where $n > 2$, we say the tuples v_I, v_{I+1}, v_{I+2} constitute an **abnormal event N** during $\Delta T(\eta)$ **($\Delta T(\eta)$ - ANE-N)**.

Sometimes X and Y don't satisfy definition 6 during $\Delta T(\eta)$ because there are abnormal events. However, the subsets X_I, Y_I of X and Y getting by deleting the abnormal events, maybe satisfy definition 6. The following definition describes this case.

Definition 14 Given a relation $R(T, A_1, \cdots, A_n)$ and

$$X = (v_1[X], \cdots, v_n[X]) \subseteq U,$$

$$Y = (v_1[Y], \cdots, v_n[Y]) \subseteq U,$$

if there is v_k corresponding to some moment t_k during $\Delta T(\eta)$ such that

$$\Psi(\Delta T(\eta), X(v_k), Y(v_k)) > \varepsilon$$

$$(k = K_1, K_2, \cdots, K_M \in \{1, 2, \cdots, m_1\}),$$

whereas for

$$X_I = X - \{v_k[X] \mid k \in \{K_1, K_2, \cdots, K_M\}\},$$

$$Y_I = Y - \{v_k[Y] \mid k \in \{K_1, K_2, \cdots, K_M\}\},$$

and any $v_i[X] \grave{E} v_j[X] \in X_I$, $v_i[Y] \grave{E} v_j[Y] \in Y_I$, whenever $v_i[X] \cong_{(d_1, \tau', t_i, t_j)} v_j[X]$ holds, there is almost $v_i[Y] \cong_{(d_2, \tau'', t_i, t_j)} v_j[Y]$ holds, and

$$\Psi(\Delta T(\eta), X_I, Y_I) = \frac{r_3}{r_{I_{\Delta T(\eta)}}} \le \varepsilon,$$

then the **relaxed type-M function dependency** during $\Delta T(\eta)$:

$$X_{I(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), X_I, Y_I) \le \varepsilon} Y_{I(d_2, \tau'')}$$

holds, where $\Psi(\Delta T(\eta), X_I(v_k), Y_I(v_k))$ is **$\Delta T(\eta)$ - $DVR\text{-}v_k[X_I, Y_I]$**, $r_{I_{\Delta T(\eta)}}$ is the combinatorial number of any pair of attributes in X_I or Y_I during $\Delta T(\eta)$, $r_3 \subseteq r_{I_{\Delta T(\eta)}}$ is the number of the DVEs $-X_I, Y_I$ during $\Delta T(\eta)$, m_1 is the number of the tuples during $\Delta T(\eta)$, $d_1 \in D[X]$, $d_2 \in D[Y]$, and $\tau', \tau'', \varepsilon \in [0,1]$ are thresholds.

Remark 4: If $X_{(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), X, Y) \le \varepsilon} Y_{(d_2, \tau'')}$ doesn't hold, and there are ANEs in X and Y, we can delete some $v_I[X]$s and $v_I[Y]$s corresponding to them from X and Y, then we get X_I and Y_I, and we have $X_{I(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), X_I, Y_I) \le \varepsilon} Y_{I(d_2, \tau'')}$ holds. This means the case of definition 14 is happening.

5. Event-Based Data Analysis Example

The following records represent a person's meditation input data including EEG from brainwave headset and GazeX and GazeY from the smart phone:

Time		EEG	GazeX	GazeY
2018-2-20	16:57:00	53	0.02884405	0.36825011
2018-2-20	16:57:01	57	-0.0057313	0.39013446
2018-2-20	16:57:02	74	0.00372011	0.33091585
2018-2-20	16:57:03	84	0.07300814	0.36468598
2018-2-20	16:57:04	90	0.06822054	0.39343803
2018-2-20	16:57:05	84	0.01829791	0.35769521
2018-2-20	16:57:06	74	0.07686714	0.4012554
2018-2-20	16:57:07	43	0.05864623	0.40079645
2018-2-20	16: 57:08	27	0.08833459	0.41172976
2018-2-20	16: 57:09	43	0.02981886	0.40139946
2018-2-20	16: 57:10	43	0.08068578	0.3896068
2018-2-20	16: 57:11	67	0.07305756	0.37007838
2018-2-20	16: 57:12	77	0.05570461	0.44665981
2018-2-20	16: 57:13	70	0.05092989	0.44977627
2018-2-20	16: 57:14	67	0.03441077	0.41223145
2018-2-20	16: 57:15	69	0.03749303	0.49343493
2018-2-20	16: 57:16	67	0.03365155	0.42283732
2018-2-20	16: 57:17	61	0.0471089	0.47274698
2018-2-20	16: 57:18	54	0.04033958	0.48874432
2018-2-20	16: 57:19	56	0.04615196	0.45340732
2018-2-20	16: 57:20	60	0.08277113	0.43117775
2018-2-20	16: 57:21	75	0.12389434	0.4264601
2018-2-20	16: 57:22	90	0.02021705	0.47553028
2018-2-20	16: 57:23	90	0.04613996	0.37573326

Firstly we define: for any attribute X,

$$d_{\max\_X_{ij}} = \max_{1 \le i, j \le m} |v_i[X] - v_j[X]|$$

denotes the maximum of the distance between the values of any two tuples v_i, v_j in the attribute X, where m is the number of the tuples during the whole time T, and

$$d(v_i[X], v_j[X]) = \frac{|v_i[X] - v_j[X]|}{d_{\max\_X_{ij}}}$$

is the distance function.

Then according to the above distance function, for the attribute EEG, for simplicity we denote it as E, we have

$$d\left(v_i[E], v_j[E]\right) = \frac{|v_i[E] - v_j[E]|}{d_{\max\_E_{ij}}},$$

$$d_{\max\_E_{ij}} = \max_{1 \le i, j \le m} |v_i[E] - v_j[E]|.$$

It is easy to see from the table that $d_{\max\_E_{ij}} = 90 - 43 = 47$.

Assuming we can choose the time from 16:57:00 to 16:57:07 on February 20th, 2018. For any pair of tuples during this time, we calculate their distance functions as follows:

$$d\left(v_1[E], v_2[E]\right) = \frac{|53 - 57|}{47} = \frac{4}{47} \approx 0.0851.$$

Similarly, we can get

$d\left(v_2[E], v_5[E]\right) \approx 0.7021$, $d\left(v_3[E], v_6[E]\right) \approx 0.2128$,

$d\left(v_4[E], v_7[E]\right) \approx 0.2128$, $d\left(v_5[E], v_8[E]\right) = 1$,

$\vdots$

Obviously, if $\tau' = 0.7$, then except that

$$d\left(v_1[E], v_5[E]\right) \approx 0.7872 > \tau',$$
$$d\left(v_2[E], v_5[E]\right) \approx 0.7021 > \tau',$$
$$d\left(v_4[E], v_8[E]\right) \approx 0.8723 > \tau',$$
$$d\left(v_5[E], v_8[E]\right) = 1 > \tau',$$
$$d\left(v_6[E], v_8[E]\right) \approx 0.8723 > \tau',$$

for the other pair of tuples, $d\left(v_i[E], v_j[E]\right) \le \tau'$ $(i, j = 1, 2, \cdots, 8,\ i \ne j)$ always holds.

And for the attribute GazeY, for simplicity we denote it as GY, then

$$d\left(v_i[GY], v_j[GY]\right) = \frac{|v_i[GY] - v_j[GY]|}{d_{\max\_GY_{ij}}},$$

$$d_{\max\_GY_{ij}} = \max_{1 \le i, j \le m} |v_i[GY] - v_j[GY]|,$$

and

$$d_{\max\_GY_{ij}} = 0.4012554 - 0.33091585 = 0.07033955.$$

Thus we can similarly get their distance functions:

$$d\left(v_1[GY], v_2[GY]\right) = \frac{|0.36825011 - 0.39013446|}{0.07033955} \approx 0.3111,$$

and

$$d\left(v_2[GY], v_3[GY]\right) \approx 0.8419,$$
$$d\left(v_3[GY], v_5[GY]\right) \approx 0.9017,$$
$$d\left(v_4[GY], v_6[GY]\right) \approx 0.0994,$$
$$d\left(v_5[GY], v_8[GY]\right) \approx 0.0918,$$

$\vdots$

If $\tau'' = 0.6$, then except that

$$d\left(v_2[GY], v_3[GY]\right) \approx 0.8419 > \tau'',$$
$$d\left(v_3[GY], v_5[GY]\right) \approx 0.9017 > \tau'',$$
$$d\left(v_3[GY], v_7[GY]\right) = 1 > \tau'',$$
$$d\left(v_3[GY], v_8[GY]\right) \approx 0.9935 > \tau'',$$
$$d\left(v_6[GY], v_7[GY]\right) \approx 0.6193 > \tau'',$$
$$d\left(v_6[GY], v_8[GY]\right) \approx 0.6128 > \tau'',$$

for the other pair of tuples, $d\left(v_i[GY], v_j[GY]\right) \le \tau''$ $(i, j = 1, 2, \cdots, 8,\ i \ne j)$ always holds.

It is clear that $v_2[E] \cong_{(d_1, \tau', t_i, t_j)} v_3[E]$ holds, whereas $v_2[GY] \cong_{(d_2, \tau'', t_i, t_j)} v_3[GY]$ doesn't holds. By the definition 3, this is a dependency violation event (DVE). In fact, $\Delta T(\eta)$- DVEs -[E, GY] are as follows:

$$\Delta T(\eta)\text{- DVEs -}v_2, v_3\text{[E, GY]},$$
$$\Delta T(\eta)\text{- DVEs -}v_3, v_5\text{[E, GY]},$$
$$\Delta T(\eta)\text{- DVEs -}v_3, v_7\text{[E, GY]},$$
$$\Delta T(\eta)\text{- DVEs -}v_3, v_8\text{[E, GY]},$$
$$\Delta T(\eta)\text{- DVEs -}v_6, v_7\text{[E, GY]}.$$

Therefore

$$\Psi(\Delta T(\eta), E, GY) = \frac{5}{28} \approx 0.1786.$$

If $\varepsilon = 0.18$, then

$$\Psi(\Delta T(\eta), E, GY) \le \varepsilon.$$

And according to the definition 6, as long as $v_i[E] \cong_{(d_1, \tau', t_i, t_j)} v_j[E]$ holds, there is almost $v_i[GY] \cong_{(d_2, \tau'', t_i, t_j)} v_j[GY]$ holds. So

$$E_{(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), E, GY) \le \varepsilon} GY_{(d_2, \tau'')}$$

holds.

We note that for the tuple v_3, according to definition 10, we have

$$\Psi(\Delta T(\eta), E(v_3), GY(v_3)) = \frac{4}{7} \approx 0.5714 > \varepsilon,$$

i.e., v_3 constitutes an abnormal event (ANE).

At the same time, we can get

$$\Psi(\Delta T(\eta), E(v_1), GY(v_1)) = \frac{0}{7} = 0 \le \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_2), GY(v_2)) = \frac{1}{7} \approx 0.1429 \le \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_4), GY(v_4)) = \frac{0}{7} = 0 \le \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_5), GY(v_5)) = \frac{1}{7} \approx 0.1429 \le \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_6), GY(v_6)) = \frac{1}{7} \approx 0.1429 \le \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_7), GY(v_7)) = \frac{2}{7} \approx 0.2857 > \varepsilon,$$

$$\Psi(\Delta T(\eta), E(v_8), GY(v_8)) = \frac{1}{7} \approx 0.1429 \le \varepsilon.$$

Therefore there is no ANE-1 happening during the time from 16:57:00 to 16:57:07 on February 20th, 2018. Obviously, there is also ANE-2 appearing.

It is clear during the time from 16:57:00 to 16:57:07 on February 20th that $E = (v_1[E], \cdots, v_8[E])$, $GY = (v_1[GY], \cdots, v_8[GY])$. According to the above calculation process, we know if $\varepsilon = 0.05$, then

$$E_{(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), E, GY) \le \varepsilon} GY_{(d_2, \tau'')}$$

doesn't hold. However, for

$E_1 = (v_1[E], v_2[E], v_4[E], v_5[E], v_6[E], v_7[E], v_8[E]) \subseteq E$,

$GY_1 = (v_1[GY], v_2[GY], v_4[GY], v_5[GY], v_6[GY], v_7[GY], v_8[GY]) \subseteq GY$,

there is only one dependency violation event (DVE): $\Delta T(\eta)$ - DVEs -v_2, v_3[E, GY].

Then we can get

$$\Psi(\Delta T(\eta), E_1, GY_1) = \frac{1}{21} \approx 0.0476 < \varepsilon.$$

So $E_{1(d_1, \tau')} \xrightarrow{\Psi(\Delta T(\eta), E_1, GY_1) \le \varepsilon} GY_{1(d_2, \tau'')}$. This is the case of Definition 14.

6. User Scenarios

In TDR system, sensor data from different devices, devices like temperature, humid, gaze, and etc, will be stored in a heap. In order for the administrator to better organize those data into separate relations, we have developed some tools to facilitate the process. The following are the steps how an admin can manage the system.

6.1. Scenario One: Organize Records

Upon login as an admin, you can see the following:

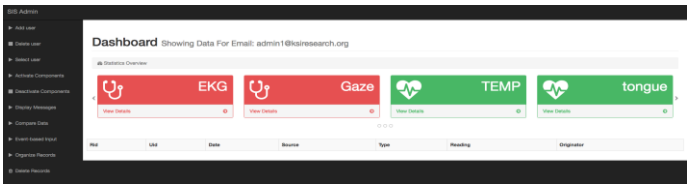


Figure 9. The Dashboard.

To write data into different relations, click organize records, then you can choose which relation you wish to write the data to.

Figure 10. Choose relations.

Upon selecting which relation the admin prefer to write data to, the system will show how many records are available in heap. The admin may type in the number of records he/she wants to write into the specific relation, but the number has to be no greater than maximum records in heap, after click on submit, the system will remind the admin whether his/her action was preformed successfully.

6.2. Scenario Two: Event-based Input

From the main page, if admin wish to move data to relations subject to certain restrict,ion he/she may choose to use event-based input tool.

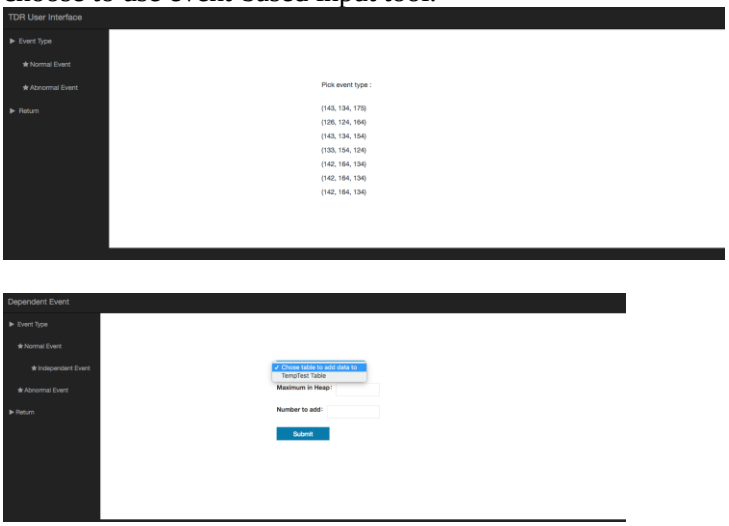


Figure 11. Normal event.

After admin has chosen normal event, admin can then pick which relation he/she wish to choose. If a tuple is a normal event for the relation, then the tool can add the tuple into the relation. Similarly, after chose abnormal event, the tool will prompt admin to pick which event (aka: dependent event or independent event) he/she want to add records to.

We will give an example upon picking dependent event, but the flow will be the same if admin chose independent event.

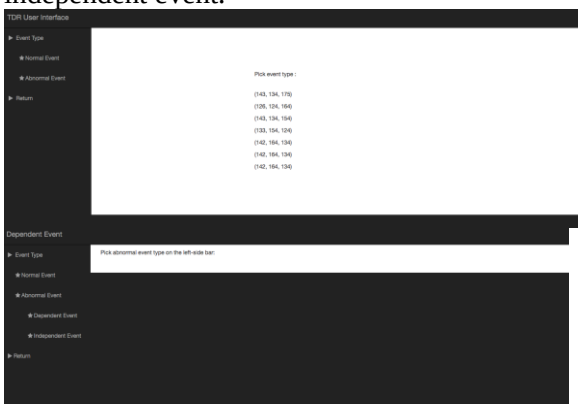


Figure 12. Independent event.

After chose which relation data admin wish to apply algorithm on, the tool will select data records and apply (2) on it, if data records satisfy (2), then move it to the correspond relation.

7. Discussion

In this paper we describe an experimental TDR system with the following features: 1) the experimental system can run on a smart phone and therefore portable; 2) a meditation validation channel to check the consistency between the predictions via gaze features vs. features to increase the accuracy of meditation prediction; 3) through event-based data input, modeling and analysis, a user can access the brainwave from a one-channel NeuroSky Mindwave headset and gaze data from a Samsung phone and the consistency check graph via a web GUI; 4) QA and rating, where a user can provide feedback right after his/her meditation process, master/teacher will rating the meditation quality based on such feedback and previous measurement data. We can also track user's typing movements when providing feedback to measure the users' muscle change during and after meditation.

An initial experiment was designed and conducted to test the ability of monitoring meditation state via brainwave and gaze tracking techniques, as well as observe the relationship between the two sources of signals. Preliminary results indicated a trend of positive relationship (correlation coefficient = 0.982) between gaze y-axis signals and brainwave signals (Figure 13), which indicates the validity of our approach in meditation detection as well as inspired us to further investigate their degrees of correlations.

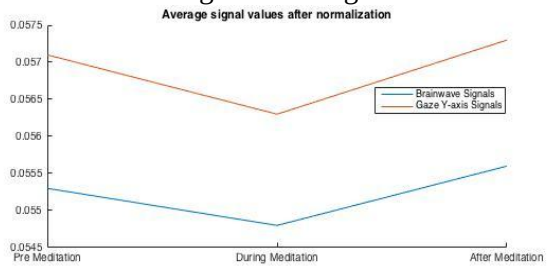


Figure 13. Preliminary results on meditation states tracking via brainwave signals and gaze signals.

The current system has certain limitations: 1) headset requires a precise wearing process to extract sensor data, otherwise a portion of the data may be missing. Users who are not professional enough or without external support, will only have partial data, which is less accurate; 2) Gaze tracking via front facing camera of smart phone is portable and maneuverable, but lack of accuracy due to the noisy luminance effect in real environment as well as the user's meditating habit.

For future work, we need to develop techniques to overcome the above mentioned limitations, as well as to design approaches to help people better understand their meditation state without too much manual intervention. More experiments need to be designed and carried out to validate the proposed approach.

Acknowledgement

This research was supported in part by Knowledge Systems Institute, USA. The research of CuiLing Chen was supported by the Visiting Scholarship Fund of Education, Department of Guangxi Zhuang Autonomous Region, P R China.

References

[1] Shi-Kuo Chang, JunHui Chen, Wei Gao and Qui Zhang, TDR System - A Multi-Level Slow Intelligence System for Personal Health Care, SEKE2016, Hotel Sofitel, Redwood City, CA, USA, 183-190.

[2] Shi-Kuo Chang, Wei Guo, Duncan Yung, ZiNan Zhang, HaoRan Zhang and WenBin You, A Mobile TDR System for Smart Phones, DMSVLSS 2017, Wyndyam Pittsburgh University Center, Pittsburgh, PA, USA, 75-85.

[3] https://en.wikipedia.org/wiki/User_interface_design

[4] S.-K. Chang, V. Deufemia, G. Polese, and M. Vacca, A normalization framework for multimedia databases, IEEE Trans. Knowl. Data Eng., vol. 19, no. 12, pp. 1666–1679, Dec. 2007.

[5] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese, Relaxed Functional Dependencies - A Survey of Approaches, IEEE Transactions on knowledge and data engineering, VOL. 28, NO. 1, JANUARY 2016.

A Markov-Model-Based Framework for Supporting Real-Time Generation of Synthetic Memory References Effectively and Efficiently

Alfredo Cuzzocrea
DIA Dept., University of Trieste and ICAR-CNR, Italy
alfredo.cuzzocrea@dia.units.it

Enzo Mumolo
DIA Dept., University of Trieste, Italy
mumolo@units.it

Marwan Hassani
MCS Dept., Eindhoven University of Technology, The Netherlands
m.hassani@tue.nl

Giorgio Mario Grasso
COSPECS Dept., University of Messina, Italy
gmgrasso@unime.it

Abstract

Driven by several real-life case studies and in-lab developments, synthetic memory reference generation has a long tradition in computer science research. The goal is that of reproducing the running of an arbitrary program, whose generated traces can later be used for simulations and experiments. In this paper we investigate this research context and provide principles and algorithms of a Markov-Model-based framework for supporting real-time generation of synthetic memory references effectively and efficiently. Specifically, our approach is based on a novel Machine Learning algorithm we called Hierarchical Hidden/non Hidden Markov Model *(HHnHMM). Experimental results conclude this paper.*.

1 Introduction

One of the problems with trace driven simulation is that trace collection and storage are time and space consuming procedures. To collect a trace, hardware or software monitors are used. The amount of data to be saved is of the order of hundreds or thousands of megabytes for some minutes the program executions. This is necessary to produce reliable results [21]. Due to the large amount of data to be processed the computer time is also very long. Several techniques have been proposed to reduce the cache simulation time: trace stripping, trace sampling, simulation of several cache configurations in one pass of the trace [35] and parallel simulation [19, 31]. Synthetic traces have been proposed as an alternative to secondary-storage based traces since they are faster and do not demand disk space. They are also attractive since and they could be controlled by a reduced set of parameters which regulate the workload behavior. The problem of Synthetic traces is that it is difficult to exactly mimic the real behavior of the addressed program, thus limiting the use of the traces to early evaluation stages. Many studies, for example [14, 36], have highlighted the difficulty to exactly describe original characteristics of the memory references, such as locality, with analytic models.

On the other hand, driven by several real-life case studies and in-lab developments, synthetic memory reference generation has a long tradition in computer science research, as confirmed by several recent research initiatives (e.g., [6, 5, 24]).

In this paper we use a machine learning approach for describing collected traces. In particular, a specific type of Markov Model (MM), the Hierarchical Hidden/non Hidden HHnMM, where each state of MM is linked to an HMM for producing sequences of labels, not just labels as in standard HMM, has been worked out. This approach is attractive because on one side the behavior of the execution is learned by the model to ensure by machine learning that the artificial sequence will mimic the behavior of the original execution and on the other side, making use of the generation characteristic of the Ergodic Hidden Markov Models, sequences of any lengths can be generated. The machine learning framework requires that a suitable feature representation of the executions is provided, as we will describe shortly. Our approach consists of a learning phase, where a real trace is analyzed with the aim to derive the features for training the HHnMM, and a generation phase, where a synthetic trace is generated from HHnMM. A preliminary version of this

DOI reference number: 10.18293/DMSVIVA2018-018

paper appears in the workshop paper [13].

2 Methodology Overview

For performance analysis of the memory subsystem of a new computer system, we may generate a long sequence of memory references from some given testing application. Generally this require to store the long sequences on a disk, which may occupy many gigabytes of disk space. This may lead to disk space unavailability or data transfer delay problems. The alternative approach is to artificially generate a sequence of memory references similar to that required by the same given software application. In Figure 1, we summarize the trace analysis algorithm described in this paper.

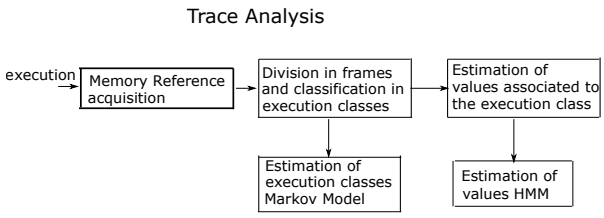


Figure 1. Trace analysis algorithm

First of all we must reduce the memory references produced by an application to a simpler representation. Thus, we divide the memory references sequence in frames, and each frame is classified as belonging to some execution classes, for example *Sequential* or *Periodic*. The execution classes are easily estimated from the reference traces. The sequence of memory references is thus transformed into a sequence of execution classes, which is a sequence with very few labels. This sequence is modeled with an Ergodic Markov Model, which is the Non Hidden part of the model. To each state of this MM, an Ergodic Hidden Markov Model is linked, for modeling the sequence of values associated with each execution class. For example the *Periodic* execution class is associated to the value of Period, or Loop width, which may change for each periodic frame. Once the non Hidden and the Hidden Markov Models are trained, artificial memory reference sequences can be generated by using the generation characteristic of the Markov Models. Namely, starting from an initial node of the MM which describe the execution classes, we generate a sequence of values associated to the execution class by visiting the associated HMM. The generation of synthetic memory references is summarized in Figure 2.

For example one may want to generate the memory references generated by the C compiler, *gcc*. The key of our algorithm is that the compiler produces somehow different memory references when used to compile different C

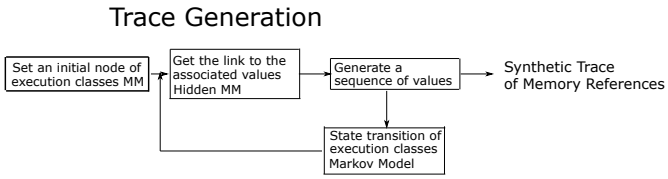


Figure 2. Synthetic trace generation algorithm

sources. The different memory references sequence anyway should contain a common structure because the same compiler *gcc* is used.

3 Describing Executions from Memory References

In this Section, we deal with the identification of the type of execution starting from the sequence of the memory reference patterns captured from the running programs. Memory reference patterns have been studied in the past by many authors with the goal to improve program execution on high performance computers or to improve memory performance. Tools to understand memory access patterns of memory operations performed by running programs are described also by Choudhury *et al.* in [8]. Such studies are directed towards the optimization of data intensive programs such as those found in scientific computing.

Other works, for example [16, 23, 27, 32], have the goal to improve memory performance, since memory systems are still the major performance and power bottleneck in computing systems. In particular, Harrison *et al.* describe in [16] the application of a simple classification of memory access patterns they developed earlier to data prefetch mechanism for pointer intensive and numerical applications. Lee *et al.* exploit the regular access patterns of multimedia applications to build hardware prefetching technique that is assisted by static analysis of data access pattern with stream caches.

Several papers by Choi *et al.*, namely [26, 7] , analyze streams of disk block requests. Choi *et al.* describe algorithms for detecting block reference patterns of applications and applies different replacement policies to different applications depending on the detected reference pattern. The block reference patterns are classified as Sequential, Looping, Temporally clustered and Probabilistic.

We remark that while the works reported above in this Section studied the way data is read or written into memory, in this paper we are interested to know how the instructions are fetched in memory during execution. Data memory reference patterns are important for memory or computation performance reasons. For us, instruction memory reference patterns are important for the generation of synthetic mem-

ory reference traces.

3.1 Instruction Memory Reference Patterns

Memory reference patterns generated by instructions have been studied in the past by several researcher, for example by Abraham and Rau [1], who studied the profiling of load instructions using the Spec89 benchmarks. Their goal was to construct more effective instruction scheduling algorithms, and to improve compile-time cache management. Austin *et al.* [2] profiled load instructions while developing software support for their fast address calculation mechanism. They reported aggregate results from their experiments, not individual instruction profiles.

We recall that our approach for generating artificial traces of memory reference consists in the analysis of the real memory reference patterns generated by an application, and in building a stochastic model of the memory reference patterns. For this purpose the memory reference sequence must be described appropriately. Therefore we divide the original sequence in short frames, and we detect the type of the underlying execution. It is worth observing that the detection of loops, and the measure of the related period, highly depends on the frame size, because if the frame size is shorter than the period, it is impossible to detect that the address stream is periodic. However, in this case we still can capture the locality of the original memory reference sequence during the generation of synthetic memory references phase, as we will describe shortly.

Clearly, the first type of execution one can think about is *Sequential*. Thus we first use a sequentiality test, described shortly. If the frame is not sequential, we apply a periodicity test to see if the sequence is *Periodic*, which means that the instructions which generate the memory addresses is of type looping. For example, consider the following matrix multiplication code, which is of course made of nested loops.

```
// Multiplying matrices a (4x3) by b (3x4)
// storing result in 'result' matrix
  for(i=0; i<4; ++i)
     for(j=0; j<4; ++j)
        for(k=0; k<3; ++k)
           result[i][j]+=a[i][k]*b[k][j];
```

Figure 3. Matrix multiplication example

The instructions of this example make memory accesses that we capture with the PIN binary instrumentation framework [30]. To this purpose we use the *itrace* Pintool, that prints the address of every instruction that is executed. In Figure 4 we show a part of the memory reference pattern generated by the matrix multiplication code.

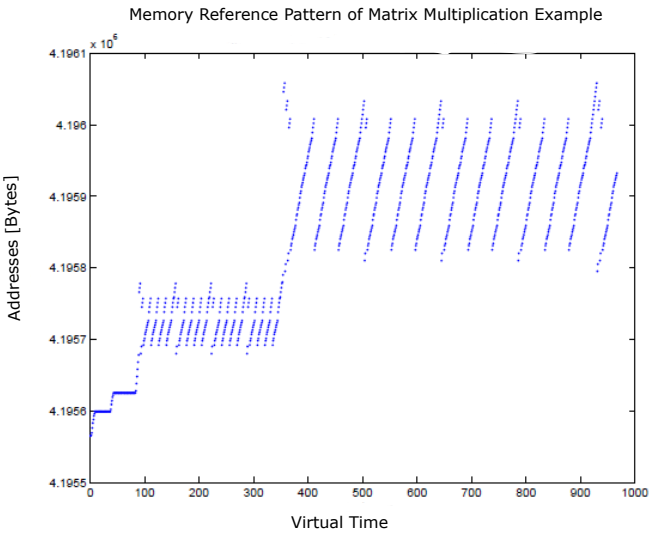


Figure 4. Memory references generated by the matrix multiplication example

We see a first burst of periodic addresses, from virtual time 100 to virtual time 350 approximately. This is the code which resets the $(4x4)$ *result* matrix. The actual matrix multiplication starts from virtual time 375 approximately.

The second example we discuss in this paper is related to indirect addressing used to access numeric vector. In the code included below, $c[]$ is a sparse array and $d[]$ is its index array. This example is taken from [16].

```
//access to sparse array
//c[]=sparse array. d[]=index array
  i=head;
  x=c[i];
  while(i){
       x += c[d[i]];
       i = d[i];
 }
```

Figure 5. Indirect address access example

In Figure 6 we show a part of the memory reference pattern generated by the numeric vector accessed with indirect addressing code. The pattern shows a periodic behavior, due to the *while* instruction. The access parts are only variable accesses.

Another aspect of this example we want to highlight is the following. We performed the generation of the index array in two ways, a deterministic and a random one. The deterministic generation code produces the memory references shown in Figure 6 while the memory references produced by random generation are shown in Figure 7.

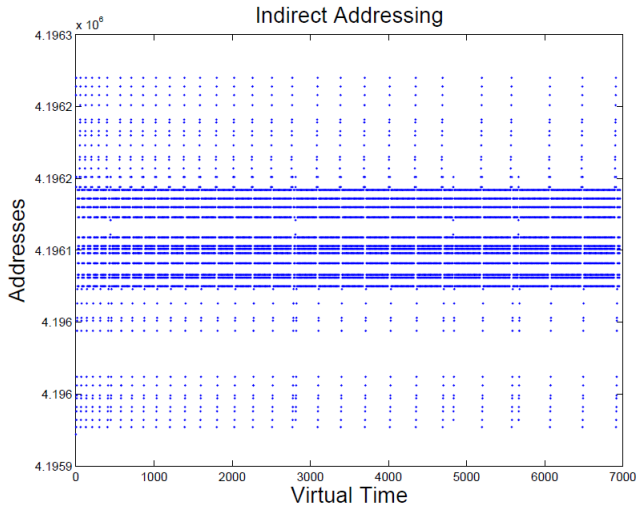


Figure 6. Memory references of the indirect access example via deterministic generation

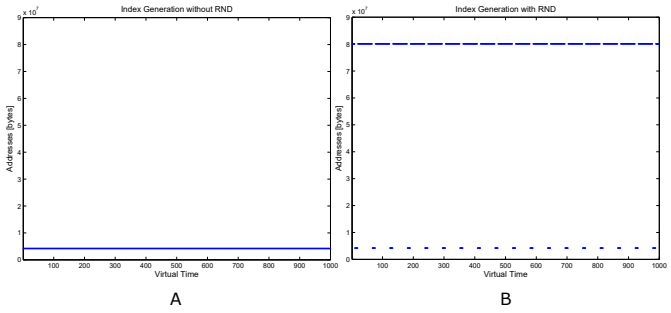


Figure 7. Memory references of the indirect access example via random generation

The difference among the two patterns is that in the random case we note that the addresses change abrutly at several virtual time instants. This is due to the calls to the subroutine *rnd*, which is a routine running at the user level. The amount of address change would have been much greater if a system call like an I/O routine had been made. Therefore, the third type of instruction is *Jump*. It is detected when the value of the addresses change greatly during a frame, which can be detected using a threshold. A *Jump* can be due to a branch in the code, or to a subroutine call or return.

The fourth type of instruction sequence is *Random*. It can be detected using a test for randomness. If no test gives a reliable output, then the frame is established to come from a sequence of instructions called *Other*, which is the fifth execution type.

3.2 Automatic Classification of Memory Reference Patterns

In this Section we describe the algorithms we used for the four tests.

1. *Sequential Pattern.* We classify the frame execution as *Sequential* as explained in the following. The values x_i of the memory reference addresses in a frame of length N are represented by the array $Frame$ reported in (1).

$$Frame = \left[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+N-1}, x_{i+N}\right] \quad (1)$$

The differences between adjacent memory address reference values are represented by the array Δ reported in (2).

$$\Delta = \left[(x_{i+1} - x_i), (x_{i+2} - x_{i+1}), \ldots, (x_{i+N} - x_{i+N-1})\right] \quad (2)$$

If all the values contained in the array Δ are positive, then $Frame$ is monotonic ascendent and it is classified as *Sequential*. Note that in this way a sequential frame can contain also ascending jumps. We assume that the monotonic test is performed by a software routine whose input argument is $Frame$. Our routine is called $Sequential(Frame)$, and has a boolean output, namely $true$ if the frame is sequential, $false$ otherwise. The $Slope$ value of sequential frames is easily found as the inclination angle of sequential patterns. $Slope$ sequences are then used to incrementally train the Hidden Markov Model $HmmS$ using a routine $HmmS = Inc\_Train(HmmS, Slope)$.

2. *Periodic Pattern.* The frame periodicity, and hence its *period*, is determined with standard spectral techniques used in signal processing for looking for signal harmonicity [28]. More precisely, given a frame of memory reference addresses as reported in (1), we weight its values with an *Hamming Window* [34], and we compute a Fast Fourier Transform [9] on it. The periodicity is detected by finding the relevant peaks in the spectrum amplitude and by looking for an harmonic structure of the peaks. For example, in Figure 8 we show a frame of size equal to 100 virtual ticks, taken from a memory reference sequence, with a clear periodic pattern. In the right pattern of Figure 8 we show the spectral amplitude of the windowed frame. In this case, the harmonicity can be easily detected. The first harmonic is the fundamental frequency of the frame is called f_0.

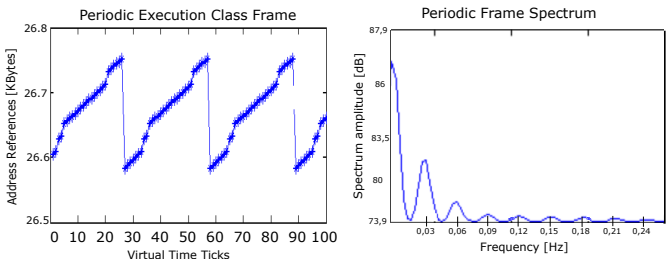


Figure 8. Spectral analysys of a periodic frame.

Since the harmonic structure shows a fundamental frequency equal to $f_0 = 0.03$ *Hz*, the period of the periodic pattern is evaluated as $1/f_0 = 33,3$. In our case the frame periodicity is computed by a software routine we call $Bounce(Frame)$. Also this routine has a Boolean output, namely $true$ if the frame is periodic and $false$ otherwise. The value of the period is estimated by the routine called $FindPeriod(Frame)$. The values of periods are used to incrementally train the Hidden Markov Model $HmmP$ as follows $HmmP = Inc\_Train(HmmP, Period)$.

3. *Random Pattern.* Many tests have been devised to verify the hypothesis of randomness of a series of observations, i.e. the hypothesis that N independent random variables have the same continuous distribution function [25].

   Our randomness test belongs the class of quick tests of the randomness of a time-series based on the sign test and variants [3]. This class of tests considers a series of N memory reference observations as that reported in (1) and the difference array reported in (2). If the observations are in random order, the expected number of plus or minus signs in (2) is $(N-1)/2$. The variance of (2) is $(N+1)/12$ and the distribution rapidly approaches normality as N increases. We then compute mean and variance of the sequence shown in (2) and we infer the frame randomness based on the similarity of the computed mean and variance with the expected ones. More precisely, we estimate the frame randomness with the routine called $Random(Frame)$.

   Once the randomness of a frame is established, its statistical distribution should be estimated for synthetic generation purposes. The discrete statistical distributions of the random variables $x_i$of the i-th frame are estimated by computing the Histograms of the frame itself. In a first step the original trace is analyzed until enough random frames are collected. For each random frame, its Histogram is computed. These Histograms divide the minimum – maximum range of the frame values in sixteen Bins, whose size is clearly $(max - min)/16$. Each Bin contains the number of values occurring in each interval divided by the frame size, say N, in order that the cumulative sum of Histogram is one. Complete information about the i-th random frame is contained in the $Stat(i)$ array reported in (3), which concatenates the Histogram values with the max and min values of the frame. In (3), $h_k(i)$ is the k-th Histogram value out of sixteen, i is the random frame index and $max(i), min(i)$ are the maximum and minimum of the i-th random frame.

$$Stat(i) = \big[h_1(i), h_2(i), \ldots, h_{16}(i), max(i), min(i)\big] \quad (3)$$

   All the obtained $Stat(i)$ arrays are combined in a $Codebook$ structure using standard clustering techniques [15]. In this way, all the random frames in the trace can be represented. We use a routine called $CodeBook = CB(H, max, min)$ for that purpose. Vector quantization of $Stat(i)$ means that each random frame is represented by an the index that corresponds to a minimum Euclidean distance between $Stat(i)$ in the trace and the $Codeword$ corresponding to the index. We code Random frames by the routine $Code = VQ(CodeBook, H, max, min)$. The code sequence is used to incrementally train the Hidden Markov Model $HmmR$ with the routine $HmmR = Inc\_Train(HmmR, Code)$.

4. *Jump Pattern.* The determination of $Jump$ patterns is straightforward. The difference between the values of the last and the beginning memory reference addresses of the frame is computed. If the difference is greater than a pre-established threshold the frame is classified as *Jump*. The Jump values are used to incrementally train the Hidden Markov Model $HmmJ$. We decide if the frame contains a jump or not with the routine $Bounce(Frame)$. The jump value is used to incrementally train the Hidden Markov Model $HmmJ$ with a routine $HmmJ = Inc\_Train(HmmJ, Code)$.

4 Experimental Results

We want to study if the algorithm is able to capture enough locality from the original traces. The simplest way to do that is to compare cache miss rate curves. We performed such experiments using a cache simulator, in particular the *Dinero IV* [22] and the benchmark suite SPEC2000 [20, 33]. Even if this benchmark suite has been officially discontinued by SPEC, still it is well suited to our purposes, as it is less demanding than more recent benchmarks, like SPEC2006. In Figure 9, Figure 10, Figure 11, and Figure

12, we report the miss-rate results for the `crafty`, `gzip`, `twolf`, `vortex` SPE2000 benchmarks.

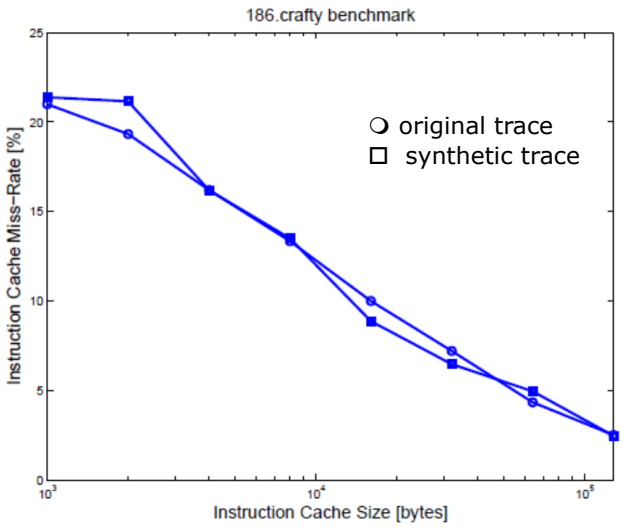


Figure 9. Original vs. synthetic instruction cache miss-rates for `crafty` benchmark

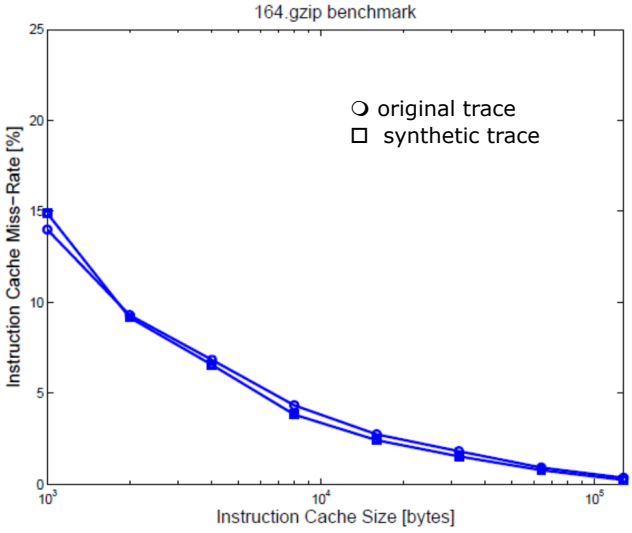


Figure 10. Original vs. synthetic instruction cache miss-rates for `gzip` benchmark

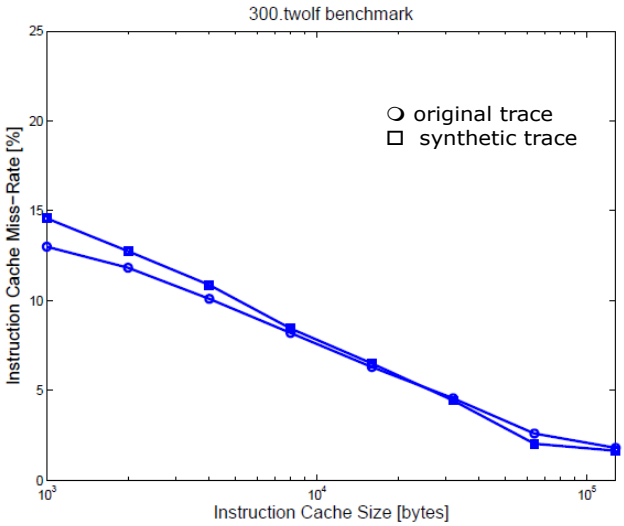


Figure 11. Original vs. synthetic instruction cache miss-rates for `twolf` benchmark

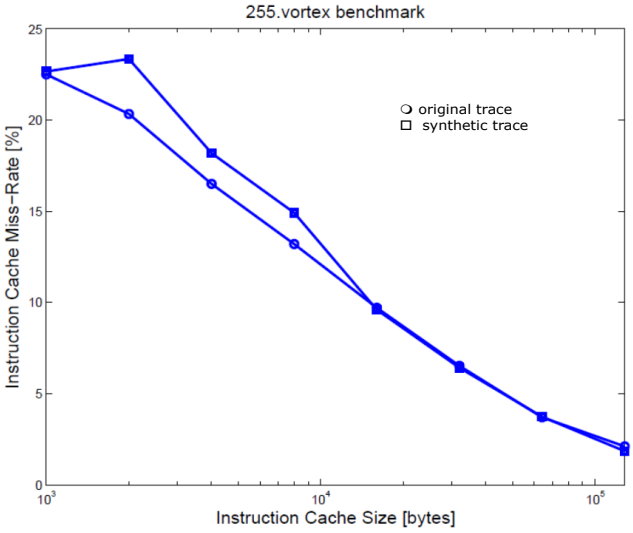


Figure 12. Original vs. synthetic instruction cache miss-rates for `vortex` benchmark

5 Final Remarks and Future Work

In this paper we describe an approach for workload characterization using ergodic hidden Markov models. The page references sequences produced by a running application are divided into short virtual time segments and used to train an HMM which models the sequence and is then used for run-

time classification of the application type and for synthetic traces generation. The main contribution of our approach are on one hand that a run-time classification of the running application type can be performed and on the other hand that the applications behavior are modeled in such a way that synthetic benchmarks can be generated.

As an extension, one can substitute *HnHMM* with stream classification methods, e.g. [29], or streaming sequential pattern mining approaches, e.g. [18], to allow for a batch-free adaptation to the sequences produced by programs during the execution, by also considering Cloud infrastructures (e.g., [11]) and big data performance issues (e.g., [12, 10]), for instance. A promising further direction that we want to additionally investigate is to improve the synthetic trace generation by considering the end-to-end context of the process that generates sequences out of program execution. In this case, application of online stream process mining will help in discovering the underlying process and adapt to it in real time [17], also following conventional approaches as regards the main adaptive model (e.g., [4]).

References

[1] S. G. Abraham and B. R. Rau. Predicting load latencies using cache profiles. *Internal Report HPL94110, Compiler and Architecture Research*, pages 1–44, 1996.

[2] T. M. Austin, D. N. Pnevmatikatos, and G. S. Sohi. Streamlining data cache access with fast address calculation. In *Proceedings of 22nd ISCA, Santa Margherita Ligure, Italy, June 22-24, 1995*, pages 369–380, 1995.

[3] C. Cammarota. The difference-sign runs length distribution in testing for serial independence. *Journal of Applied Statistics*, pages 1–11, 2010.

[4] M. Cannataro, A. Cuzzocrea, and A. Pugliese. A probabilistic approach to model adaptive hypermedia systems. In *Proceedings of the International Workshop for Web Dynamics*, pages 12–30, 2001.

[5] A. Chaurasia and V. K. Sehgal. Optimal buffer-size by synthetic self-similar traces for different traffics for noc. *SIGBED Review*, 12(3):6–12, 2015.

[6] J. Chen and R. M. Clapp. Astro: Auto-generation of synthetic traces using scaling pattern recognition for MPI workloads. *IEEE Trans. Parallel Distrib. Syst.*, 28(8):2159–2171, 2017.

[7] J. Choi, S. H. Noh, S. L. Min, E. Ha, and Y. Cho. Design, implementation, and performance evaluation of a detection-based adaptive block replacement scheme. *IEEE Trans. Computers*, 51(7):793–800, 2002.

[8] A. N. M. I. Choudhury, K. C. Potter, and S. G. Parker. Interactive visualization for memory reference traces. *Comput. Graph. Forum*, 27(3):815–822, 2008.

[9] W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling, D. Nelson, C. Rader, and P. Welch. What is the fast fourier transform? *Proceedings of the IEEE*, pages 1664 – 1674, 1967.

[10] A. Cuzzocrea. Accuracy control in compressed multidimensional data cubes for quality of answer-based OLAP tools. In *18th International Conference on Scientific and Statistical Database Management, SSDBM 2006, 3-5 July 2006, Vienna, Austria, Proceedings*, pages 301–310, 2006.

[11] A. Cuzzocrea, G. Fortino, and O. F. Rana. Managing data and processes in cloud-enabled large-scale sensor networks: State-of-the-art and future research directions. In *13th IEEE/ACM CCGrid, Delft, Netherlands, May 13-16, 2013*, pages 583–588, 2013.

[12] A. Cuzzocrea, F. Furfaro, and D. Saccà. Enabling OLAP in mobile environments via intelligent data cube compression techniques. *J. Intell. Inf. Syst.*, 33(2):95–143, 2009.

[13] A. Cuzzocrea, E. Mumolo, M. Hassani, and G. M. Grasso. Towards effective generation of synthetic memory references via markovian models. In *Proceedings of the 42nd IEEE Computer Society Signature Conference on Computers, Software and Applications, COMPSAC 2018, Tokyo, Japan, July 23-27, 2018*, 2018.

[14] D. Ferrari. On the foundations of artificial workload design. In *Proceedings of 1984 ACM SIGMETRICS, Cambridge, Massachusetts, USA, August 21-24, 1984*, pages 8–14, 1984.

[15] R. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4 – 29, 1984.

[16] L. Harrison. Examination of a memory access classification scheme for pointer-intensive and numeric programs. In *Proceedings of 10th ICS, Philadelphia, PA, USA, May 25-28, 1996*, pages 133–140, 1996.

[17] M. Hassani, S. Siccha, F. Richter, and T. Seidl. Efficient process discovery from event streams using sequential pattern mining. In *IEEE SSCI, Cape Town, South Africa, December 7-10, 2015*, pages 1366–1373, 2015.

[18] M. Hassani, D. Töws, A. Cuzzocrea, and T. Seidl. BFSP-Miner: an effective and efficient batch-free algorithm for mining sequential patterns over data streams. *International Journal of Data Science and Analytics*, Dec 2017.

[19] P. Heildelberger and H. Stone. Parallel trace-driven simulation by time partitioning. In *Proceedings of the Winter Simulation Conference*, pages 734–737, 1990.

[20] J. Henning. Spec cpu2000: measuring cpu performance in the new millennium. *Computer*, pages 1–44, 2000.

[21] M. A. Holliday. Techniques for cache and memory simulation using address reference traces. *Int. Journal in Computer Simulation*, 1(2), 1991.

[22] M. D. H. Jan Elder. Dinero iv trace-driven uniprocessor cache simulator, 2003.

[23] B. Jang, D. Schaa, P. Mistry, and D. R. Kaeli. Exploiting memory access patterns to improve memory performance in data-parallel architectures. *IEEE Trans. Parallel Distrib. Syst.*, 22(1):105–118, 2011.

[24] N. E. Jivan and M. R. Dagenais. A stateful approach to generate synthetic events from kernel traces. *Adv. Software Engineering*, 2012:140368:1–140368:12, 2012.

[25] D. E. Knuth. *The art of computer programming, Volume II: Seminumerical Algorithms*. Addison-Wesley, 1998.

[26] D. Lee, J. Choi, J. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. Kim. LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies. *IEEE Trans. Computers*, 50(12):1352–1361, 2001.

[27] J. Lee, C. Park, and S. Ha. Memory access pattern analysis and stream cache design for multimedia applications. In *Proceedings of 2003 ASP-DAC, Kitakyushu, Japan, January 21-24, 2003*, pages 22–27, 2003.

[28] T. Lobos, Z. Leonowicz, and J. Rezmer. Harmonics and interharmonics estimation using advanced signal processing methods. In *Harmonics and Quality of Power, 2000. Proceedings. Ninth International Conference on*, pages 335–340, 2000.

[29] Y. Lu, M. Hassani, and T. Seidl. Incremental temporal pattern mining using efficient batch-free stream clustering. In *Proceedings of 29th SSDBM, Chicago, IL, USA, June 27-29, 2017*, pages 7:1–7:12, 2017.

[30] C. Luk, R. S. Cohn, R. Muth, H. Patil, A. Klauser, P. G. Lowney, S. Wallace, V. J. Reddi, and K. M. Hazelwood. Pin: building customized program analysis tools with dynamic instrumentation. In *Proceedings of 2005 ACM SIGPLAN PLDI, Chicago, IL, USA, June 12-15, 2005*.

[31] D. Nicol, A. Greenberg, and B. Lubachevsky. Massively parallel algorithms for trace-driven cache simulation. In *Proceedings of the 6th workshop on Parallel and Distributed Simulation (1992)*, pages 3–11, 1992.

[32] A. J. Niessen and H. A. G. Wijshoff. Address reference generation in a memory hierarchy simulator environment, 1995.

[33] C. Niki, J. Thornock, and K. Flanagan. Using the bach trace collection mechanism to characterize the spec2000 integer benchmarks. In *Proceedings of the Third IEEE Annual Workshop on Workload Characterization*, pages 369–380, 2000.

[34] P. Podder, T. Z. Khan, M. H. Khan, and M. M. Rahman. Comparative performance analysis of hamming,hanning and blackman window. *International Journal of Computer Applications (0975 8887)*, 96(18):1–7, 2014.

[35] H. S. Stone. *High-performance computer architecture (3. ed.)*. Addison-Wesley, 1993.

[36] D. Thiébaut, J. L. Wolf, and H. S. Stone. Synthetic traces for trace-driven simulation of cache memories. *IEEE Trans. Computers*, 41(4):388–410, 1992.

Spider Diagrams with Absence

Gem Stapleton[1] and Lopamudra Choudhury[2] and Mihir Chakraborty[2]
[1] Centre for Secure, Intelligent and Usable Systems,
University of Brighton, UK
[2] Jadavpur University, India
g.e.stapleton@brighton.ac.uk, choudhuryl@yahoo.com, mihirc4@gmail.com

Abstract

Spider diagrams have been developed as a visual logic for representing information about sets, their cardinalities and, sometimes, the specific individuals within those sets. They are expressively equivalent to monadic first-order logic with equality. Existing diagrammatic logics with this level of expressiveness are not capable of directly *expressing the* absence *of a particular individual from a set. Instead, individuals must be asserted to be* present *in some particular set and, thus, absent from the set's complement. The first work to allow absence to be directly asserted was seen in the Venn-i system. It has since been shown that, when expressiveness is limited to monadic first-order logic without equality, the inclusion of absence information can significantly reduce diagram clutter. In this paper, we extend spider diagrams to include direct representation of the absence of individuals from sets. After formalizing the syntax and semantics, we give conditions for satisfiability. Building on that, we introduce inference rules specifically related to spiders (which represent elements, individuals or their absence) that alter the levels of clutter in consistent diagrams. In the context of these rules, we explore the implications of including absence information for reducing clutter.*

1. Introduction

The ability to negate statements plays a crucial role in all logics. The notion of absence is closely related to that of negation: $a \notin P$ (i.e. the individual a is not in the set P) indicates, informally speaking, that the individual a is *absent* from P. Indeed, the importance of negation should not be underestimated, "The capacity to negate is the capacity to refuse, to contradict, to lie, to speak ironically, to distinguish truth from falsity – in short, the capacity to be human" [7]. In diagrams research, though, it has long been believed that diagrams are not well equipped to make negated statements directly. Indeed, even simple statements like $a \notin P$ cannot be made explicitly in most Euler diagram-based logics, such as [10, 14, 17, 18]. Instead, these types of diagrams tend to assert $a \in \overline{P}$ (the complement of P).

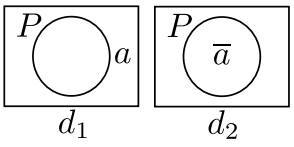


Figure 1. Asserting presence and absence.

There is an exception to this: Choudhury and Chakraborty developed a diagrammatic logic named Venn-i that allows $a \notin P$ to be *directly* expressed [4]. The Venn-i logic builds on Shin's Venn-I system [15], which exploits Peirce's $\otimes$-sequences to indicate the non-emptiness of sets [13]. Venn-i also uses i-sequences and $\bar{i}$-sequences to represent individuals and, respectively, their absence. Choudhury and Chakraborty adopt a classical interpretation, meaning that the absence of an individual from one set implies its presence in the complement[1]. An inspiration for Choudhury's and Chakraborty's work came from the notion of *abhāva* (absence). Abhāva, an important feature of ancient Indian knowledge systems, allocates a first class status to the absence of individuals.

Examples can be seen in figure 1. The diagram d_1 directly expresses that a is in $\overline{P}$, since the location of the symbol a is outside the curve P. From this, we can deduce that a is not in P, that is, a is absent from P. By contrast, the diagram d_2 directly expresses that a is absent P by placing $\overline{a}$ inside P. Thus, the use of $^-$ depicts the absence of individ-

DOI reference number: 10.18293/DMSVIVA2018-014.

[1] A related system, developed by Bhattacharjee et al. focuses on a non-classical interpretation of absence [2]. In that system, the absence of an individual from one set *does not* imply its presence in the complement. They devised a sound and complete set of inference rules which allow diagrammatic proofs to be written when absence information is given.

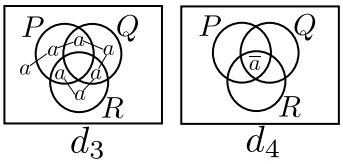


Figure 2. Clutter reduction.

uals from a set. Whilst this example is sufficient to illustrate the use of absence, it does not demonstrate the role that absence can play in reducing diagram clutter. We explore this more fully later in the paper. For now, suppose we want to express that a is absent from $P \cap Q \cap R$. Referring to figure 2, this is achieved indirectly by d_3 which expresses that $a \in \overline{P \cap Q \cap R}$ (i.e. the complement of $P \cap Q \cap R$). The diagram d_4 is semantically equivalent but instead expresses the required statement, $a \notin P \cap Q \cap R$, directly using absence. The diagram d_4 is arguably less cluttered than d_3.

Clutter in Euler diagrams was studied by John et al. [12]: they devised a theoretical measure of clutter. Alqadah et al. empirically found that increasing levels of clutter in Euler diagrams negatively impacts user task performance [1]. Whilst empirically studying concept diagrams [11] (which extend Euler diagrams with syntax for individuals amongst other things), Hou et al. found that the diagrams where people had trouble performing tasks were those with the higher levels of clutter [8, 9]. Hence, there is clearly a need to be able to measure the level clutter in diagrams generally and its impact on end-user task performance.

In previous work we, with Burton, demonstrated that explicitly representing the absence of individuals allows information to be presented in a less cluttered way [3]. This was in an Euler diagram system, called Venn-i^e, that incorporated $\otimes$-sequences, i-sequences (like the a-sequence in d_3)and $\overline{i}$-sequences (like $\overline{a}$ in d_4). We note here that, just as in Venn-i, distinct i-sequences need not represent distinct individuals. As such, Venn-i^e is a monadic first-order logic (without equality). An empirical evaluation suggested, for Venn-i^e, high levels of clutter arising from individuals is detrimental to human cognition [16].

There are two key points: (a) rising levels of diagram clutter negatively impacts human cognition and (b) representing absence directly in systems with the expressiveness of monadic first-order logic (without equality) can bring reductions in clutter. An obvious question arises: does the inclusion of absence in more expressive systems still bring with it possibility of reducing clutter to the same extent? This paper takes the first step towards understanding clutter arising from spiders (akin to sequences) in an extended version of spider diagrams, which we call enhanced spider diagrams. This increases the level of expressiveness, over which we can explore the role of absence, to monadic first-order logic with equality. In particular, we make the following contributions:

- identify how to extend spider diagrams to include absence information (section 2).
- formalize the syntax and semantics of enhanced spider diagrams (section 3),
- identify necessary and sufficient conditions for enhanced spider diagrams to be unsatisfiable (section 4),
- introduce inference rules for enhanced spider diagrams that apply specifically to spiders (section 5), and
- discuss how the inference rules can be used to alter the level of clutter present in enhanced spider diagrams (section 6).

We conclude and discuss future work in section 7.

2. Representing Absence in Spider Diagrams

We now proceed to, briefly, show how absence can be incorporated into spider diagrams [5], which typically include *existential* spiders for denoting the existence of elements in sets. In addition, they have been studied with the inclusion of *constant spiders* [17] which, in this paper, we refer to as *positive spiders*. These spiders represent the presence of specific individuals in particular sets.

An example can be seen in figure 3. The spider diagram d_5 expresses the following: (i) due to the spatial relationships between the curves, R is disjoint from P and from Q; (ii) due to the inclusion of two spiders, there are at least two elements, one of which (denoted by the existential spider comprising two nodes) is in P and the other of which (denoted by the positive spider a) is the individual a and is in the set $P \backslash Q$, and (iii) due to the shading, combined with the existential spider, there is at most one element in $P \cap Q$.

The diagram d_6 augments d_5 with additional information expressed by four *negative* spiders: (iv) the individual b is not in $\overline{P}$ and the individual c is not in R. Therefore, from (iv), b is in the set P. A natural question then arises: should the individual b be necessarily different from the two elements represented by the existential spider and the positive spider a? In our view, the most *diagrammatic* interpretation is that b is *not* necessarily different. It seems natural to say that two existential or positive spiders placed in a common region represent distinct individuals since they are represented by distinct syntactic devices. However, it does not seem very diagrammatic for d_6 to force P to contain at least *three* elements when we can see only *two* spiders inside P. This observation suggests that the diagram would not be *well-matched* to its semantics [6] if we forced b to denote the presence of an *additional* element in P. Therefore, in our extension of spider diagrams to include absence information directly, via *negative* spiders, we do not interpret negative spiders as providing distinctness information about individuals, unlike existential and positive spiders.

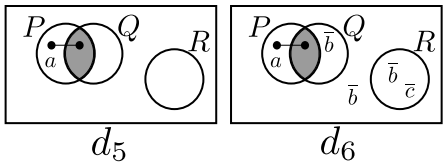


Figure 3. Incorporating absence.

3. Syntax and Semantics

Having introduced how absence can be incorporated into spider diagrams, we now formally define the syntax and semantics of the enhanced system.

It is helpful for us to have a countably infinite set of labels from which all labels used on the curves in any diagram are drawn; we call this set $\mathcal{L}$. A **zone** is a pair, (in, out), where in and out are finite, disjoint subsets of $\mathcal{L}$. Given $\mathcal{L}$, the set of all zones is denoted $\mathcal{Z}$. In figure 3, each diagram has five zones, such as the one inside both P and Q but outside R; this zone is $(\{P, Q\}, \{R\})$. We also have a countably infinite set of constant symbols, denoted $\mathcal{C}$, which are used as labels for *positive* and *negative* spiders. So, in figure 3, spider labels a, b and c appear. Using these predefined sets, we can now formally define an *enhanced spider diagram* at the abstract syntax level:

Definition 1 *An **enhanced spider diagram**, d, is a tuple, $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ such that:*

1. *L is a finite set of labels chosen from $\mathcal{L}$.*
2. *Z is a set of zones where $(\emptyset, L) \in Z$ and for all (in, out) in Z, $in \cup out = L$.*
3. *ShZ is a subset of Z whose elements are called **shaded** zones.*
4. *ES, PS, NS are finite pairwise disjoint sets whose elements are called **existential spiders**, **positive spiders**, and **negative spiders** respectively.*
5. *η: $ES \cup PS \cup NS \to \mathbb{P}Z\backslash\{\emptyset\}$ returns the location of each spider.*
6. *ρ: $PS \cup NS \to \mathcal{C}$ returns the label of each positive and negative spider.*

*We further define $S(d) = ES \cup PS \cup NS$ to be the set of **spiders** in d.*

In figure 3, d_6 has the following abstract syntax:

1. $L = \{P, Q, R\}$.
2. $Z = \{(\{P\}, \{Q, R\}), (\{Q\}, \{P, R\}), (\{P, Q\}, \{R\}), (\{R\}, \{P, Q\}), (\emptyset, \{P, Q, R\})\}$.
3. $ShZ = \{(\{P, Q\}, \{R\})\}$.
4. $ES = \{\sigma_1\}$, $PS = \{\sigma_a\}$, $NS = \{\sigma_{b,1}, \sigma_{b,2}, \sigma_{b,3}, \sigma_c\}$.
5. $\eta(\sigma_1) = \{(\{P\}, \{Q, R\}), (\{P, Q\}, \{R\})\}$, $\eta(\sigma_a) = \{(\{P\}, \{Q, R\}), \quad \eta(\sigma_{b,1}) = \{(\{Q\}, \{P, R\})\}$, $\eta(\sigma_{b,2}) = \{(\emptyset, \{P, Q, R\})\}$, $\eta(\sigma_{b,3}) = \{(\{R\}, \{P, Q\})\}$, and $\eta(\sigma_c) = \{(\{R\}, \{P, Q\})\}$.
6. $\rho(\sigma_a) = a$, $\rho(\sigma_{b,1}) = \rho(\sigma_{b,2}) = \rho(\sigma_{b,3}) = b$, and $\rho(\sigma_c) = c$.

We now proceed to define some useful syntactic notions. For example, in figure 3, the zone $(\{P, R\}, \{Q\})$ is *missing* from d_6 (and d_5). Since d_6 is taken to assert that R is disjoint from both P and Q, the set $P \cap R \cap \overline{Q}$ represented by this zone must be empty.

Definition 2 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. The **missing zones** of d are elements of*

$$MZ(d) = \{(in, out) \in \mathcal{Z} : in \cup out = L\}\backslash Z.$$

So, d_6 has three missing zones, giving

$$MZ(d) = \{(\{P, R\}, \{Q\}), (\{Q, R\}, \{P\}), (\{P, Q, R\}, \emptyset)\}.$$

Missing zones need not be the only zones that represent empty sets. In particular, shading placed in zones enforces an upper bound on set cardinality: in a shaded region, all elements must be represented by existential or positive spiders. So, a shaded region containing no part of any such spider represents the empty set. There are no *necessarily* empty zones in d_6 but, in figure 4, the diagram d_7 only contains empty zones (all zones are shaded and there are no existential or positive spiders).

Definition 3 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. The **empty zones** of d are elements of*

$$EZ(d) = \{z \in ShZ : \forall \sigma \in ES \cup PS \quad z \notin \eta(\sigma)\}.$$

It is also useful to identify zones that represent sets in which an individual cannot lie, due to the information provided by negative spiders. For instance, in d_6 of figure 3, b is not in (the sets represented by) the three zones which contain $\bar{b}$.

Definition 4 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. Let c be a constant from $\mathcal{C}$. The **negative zones** for c in d are elements of $NZ(c, d)$ where*

$$NZ(c, d) = \{z \in Z : \exists \sigma \in NS \quad \eta(\sigma) = \{z\} \wedge \rho(\sigma) = c\}.$$

So, in d_6, we have:

- no negative zone for a: $NZ(a, d_6) = \emptyset$,
- three negative zones for b since there are three $\bar{b}$s placed in single zones: $NZ(b, d_6) = \{(\{Q\}, \{P, R\}), (\emptyset, \{P, Q, R\}), (\{P\}, \{P, Q\})\}$ and

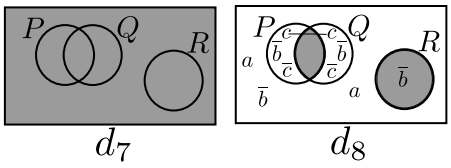


Figure 4. Inconsistency and clutter.

- one negative zone for c, since there is one $\overline{c}$ placed in a single zone: $NZ(c, d_6) = \{(\{R\}, \{P, Q\})\}$.

Our attention now turns to semantics. As is standard, we interpret the curve labels as subset of some (non-empty) universal set and constant symbols as elements.

Definition 5 *An **interpretation**, $\mathcal{I}$, is a triple, $\mathcal{I} = (U, \psi, \Psi)$, such that*

1. *U is a non-empty set, called the **universal set**,*
2. *$\psi: \mathcal{C} \to U$ maps constants to elements in U, and*
3. *$\Psi: \mathcal{L} \to \mathbb{P}U$ maps curve labels to subsets of U.*

The function Ψ is extended to interpret zones and sets of zones (regions) as follows: for each zone, (in, out),

$$\Psi(z) = \bigcap_{l \in in} \Psi(l) \cap \bigcap_{l \in out} (U \backslash \Psi(l))$$

and for each region, r, $\Psi(r) = \bigcup_{z \in r} \Psi(z)$.

Given an interpretation, it can either agree with the intended intuitive meaning of a diagram or not. For instance, in figure 3, any interpretation where $\Psi(P) \cap \Psi(R) \neq \emptyset$ does not agree with the intended intuitive meaning of d_6: because the curves P and R do not overlap, d_6 tells us that $\Psi(P) \cap \Psi(R) = \emptyset$. The semantics of diagrams are given by the set of interpretations that match our expectations of what the diagram expresses. We therefore give precise conditions under which an interpretation *satisfies* an enhanced spider diagram; satisfying interpretations are called *models*.

Definition 6 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram and let $\mathcal{I} = (U, \psi, \Psi)$ be an interpretation. Then $\mathcal{I}$ is a **model** for d provided there exists a function, $\psi': ES \cup PS \cup NS \to U$, where, for each positive and negative spider, σ, in $PS \cup NS$ we have $\psi'(\sigma) = \psi(\rho(c))$ and the following conditions hold.*

1. ***Missing Zones Condition:** missing zones represent empty sets, that is for all $z \in MZ(d)$, $\Psi(z) = \emptyset$.*
2. ***Shaded Zones Condition:** shaded zones represent sets containing only elements represented by existential or positive spiders, that is for all $z \in ShZ$,*

$$\Psi(z) \subseteq \{\psi'(\sigma) : \sigma \in ES \cup PS\}.$$

3. ***Spider Distinctness Condition** no two existential or positive spiders represent the same element, that is for all σ_1 and σ_2 in $ES \cup PS$,*

$$\psi'(\sigma_1) = \psi'(\sigma_2) \Rightarrow \sigma_1 = \sigma_2.$$

4. ***Existential Spiders Condition** each existential spider represents an element in the region in which it is placed, that is for all σ in ES, $\psi'(\sigma) \in \Psi(\eta(\sigma))$.*
5. ***Positive Spiders Condition** each positive spider represents an element in the region in which it is placed, that is for all σ in PS, $\psi'(\sigma) \in \Psi(\eta(\sigma))$.*
6. ***Negative Spiders Condition** each negative spider represents an element that is not some zone in the region in which it is placed, that is for all σ in NS, there exists z in $\eta(\sigma)$ where $\psi'(\sigma) \notin \Psi(z)$.*

*Such a function ψ' that makes the above conditions true for d is called **valid**. If $\mathcal{I}$ models d then $\mathcal{I}$ **satisfies** d. Diagrams with no models are **unsatisfiable**.*

The interpretation with $U = \{1, 2, 3, 4\}$, $\psi(a) = 1$, $\psi(b) = 2$, $\psi(c) = 3$, $\Psi(P) = \{1, 2\}$, $\Psi(Q) = \{2, 3\}$ and $\Psi(R) = \{4\}$ is a model for d_6. However, if instead $\Psi(Q) = \{1, 2, 3\}$ then the resulting interpretation fails to model d_6; for instance, $\psi(a) = 1$ but, since the positive spider labelled a is in the zone $(\{P\}, \{Q, R\})$ and we have

$$\begin{aligned} \Psi(\{P\}, \{Q, R\}) &= \Psi(P) \cap (U \backslash \Psi(Q)) \cap (U \backslash \Psi(R)) \\ &= \emptyset, \end{aligned}$$

the positive spiders condition fails under any ψ' that agrees with ψ.

We now present two results relating to, respectively, empty zones and negative zones. Firstly, we establish that empty zones represent empty sets in models:

Lemma 1 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. In all models, $I = (U, \psi, \Psi)$, for d, $\Psi(EZ(d)) = \emptyset$.*

Secondly, we show that, for any given spider label, c, its associated negative zones do not contain the individual represented by c. For instance, in d_6 we already saw that $NZ(b, d_6)$ includes $(\{Q\}, \{P, R\})$. This zone is the location for a negative spider, $\sigma_{b,1}$, labelled b, that is $\eta(\sigma_{b,1}) = \{(\{Q\}, \{P, R\})\}$. In any model for d_6, the negative spiders condition tells us that $\psi'(b) = \psi(b) \notin \Psi(\{Q\}, \{P, R\})$. Importantly, the negative zones arise precisely from the negative spiders whose locations are single zones, from which the proof of the following lemma readily follows:

Lemma 2 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. In all models, $I = (U, \psi, \Psi)$, for d, for all $c \in \mathcal{C}$, it is the case that $\psi(c) \notin \Psi(NZ(c, d))$.*

Thus, the definitions of empty zones and negative zones have the expected properties in models.

4 Inconsistency and Satisfiability

A key motivation for this work is to explore the role of absence in clutter reduction. We begin by observing that every unsatisfiable diagram is semantically equivalent to a diagram containing no spiders. For example, in figure 4, d_7 is inconsistent: every interpretation has a non-empty universal set yet, since d_7 is entirely shaded and contains no spiders, the shading and missing zones conditions can never both be satisfied (a non-empty universal set implies at least one zone represents a non-empty set). The diagram d_8 is also unsatisfiable, for any one of the following reasons:

1. There are two positive spiders both with the same label, a, meaning that the spider distinctness condition can never hold.
2. The negative $\overline{b}$ spiders together imply that b must lie in $P \cap Q \cap \overline{R}$, yet this region is entirely shaded and contains no part of an existential or positive spider. Therefore, d_8 implies two contradictory statements: $b \in P \cap Q \cap \overline{R}$ and $P \cap Q \cap \overline{R} = \emptyset$.
3. The negative $\overline{c}$ spiders tell us that c cannot lie in $P \cap \overline{Q} \cap \overline{R}$ or in $Q \cap \overline{P} \cap \overline{R}$, yet the positive spider c expresses $c \in (P \cap \overline{Q} \cap \overline{R}) \cup (Q \cap \overline{P} \cap \overline{R})$. Clearly both these assertions cannot be true at the same time.

Since d_8 is inconsistent, it is semantically equivalent to d_7, which is visually less cluttered. Since every inconsistent diagram is semantically equivalent to a diagram that is entirely shaded and contains no spiders, identifying necessary and sufficient conditions for unsatisfiability provides some insight into how negative spiders can lead to clutter reduction.

One important feature of the last example was that it was not possible to find zones that represent sets containing certain individuals. For instance, there was no zone for the individual c since it was taken to be present in $(\{P\}, \{Q, R\})$ or $(\{Q\}, \{P, R\})$ yet absent from both $(\{P\}, \{Q, R\})$ and $(\{Q\}, \{P, R\})$. For a diagram to be satisfiable, for each constant, c_i, we must be able to select a zone that, in some model, represents a set containing the individual represented by c_i.

Definition 7 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. A **zone selection function** for d is a mapping, $f\colon ES \cup PS \cup NS \to Z$ which ensures the following hold:*

1. *the zone selected for each existential and positive spider is one of the zones in its location: for all $\sigma \in ES \cup PS$, $f(\sigma) \in \eta(\sigma)$,*
2. *the zone selected for a negative spider cannot be a negative zone for its label: for all $\sigma \in NS$, $f(\sigma) \notin NZ(\rho(\sigma), d)$ and*
3. *if a shaded zone is selected for a negative spider then it must also be selected for an existential or positive spider: for all $\sigma \in NS$, if $f(\sigma) \in ShZ$ then there exists $\sigma' \in ES \cup PS$ such that $f(\sigma') = f(\sigma)$, and*
4. *spiders with the same label have the same zone selected: for all σ_1, σ_2 in $PS \cup NS$ if $\rho(\sigma_1) = \rho(\sigma_2)$ then $f(\sigma_1) = f(\sigma_2)$.*

The zone selection function identifies, for each spider, a specific zone in the diagram, d. Any given zone selection function can be used to define a model for d, where the individuals represented by the spiders are in the sets represented by the selected zones. For the purposes of intuition, we consider each of the conditions of definition 7. Condition 1 arises from the need for each existential and positive spider to represent an element in (the set represented by) one of the zones of its location. Condition 2 captures the fact that negative zones cannot contain, in a model for d, the individual represented by $\rho(\sigma)$. Condition 3 considers the interaction between negative spiders and shading. The zone selected for a negative spider, if shaded, cannot represent the empty set in a model. This is enforced by the requirement that some existential or positive spider has been assigned to that shaded zone. The last condition requires the same zone to be selected for spiders with a common label because such spiders represent the same individual.

Using d_6 in figure 3 as an example, adopting the previously given abstract syntax, we can define $f(\sigma_1) = \{(\{P, Q\}, \{R\})\}$, $f(\sigma_a) = \{(\{P\}, \{Q, R\})\}$, $f(\sigma_{b,1}) = f(\sigma_{b,2}) = f(\sigma_{b,3}) = \{(\{P, Q\}, \{R\})\}$, and $f(\sigma_c) = \{(\{P, Q\}, \{R\})\}$. Under this zone selection function, a model can be generated for d_6 where b and c represent the same individual and that individual is in the set represented by the zone $\{(\{P, Q\}, \{R\})\}$. Under any valid ψ', this individual is also represented by σ_1, due to the presence of shading.

We are now in a position to define the notion of (in)consistency:

Definition 8 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. Whenever the following conditions all hold d is **consistent**.*

1. *If all of the zones in Z are shaded then there is at least one existential or positive spider.*
2. *No two positive spiders have the same label, that is, the function ρ is injective when its domain is restricted to PS.*
3. *There exists a zone selection function, f, for d.*

*If d is not consistent then d is **inconsistent**.*

So, d_6 in figure 3 is consistent whereas d_7 and d_8 in figure 4 are inconsistent.

Theorem 1 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. Then d is consistent if and only if d is satisfiable.*

5 Inference Rules

The goal of this section is to introduce inference rules that can later be used to reduce clutter in enhanced spider diagrams. These inference rules focus on spiders only. It is therefore helpful to introduce transformations on diagrams that remove and add spiders. In what follows we use $|$ to indicate a domain restriction.

Transformation 1 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. Let σ be a spider in $S(d)$. We define a spider removal operation on d:*

$$d - \sigma = (L, Z, ShZ, ES\backslash\{\sigma\}, PS\backslash\{\sigma\}, NS\backslash\{\sigma\}, \eta', \rho')$$

where $\eta' = \eta|_{S\backslash\{\sigma\}}$, and $\rho' = \rho|_{(PS\cup NS)\backslash\{\sigma\}}$.

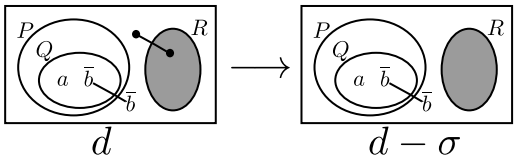


Figure 5. Removing a spider.

For example, in figure 5, the existential spider, σ, is removed from d to give $d - \sigma$. This removal transformation only needs to 'know' which spider to remove. However, when adding a spider, we need to know the location in which it is to be placed and, if it is positive or negative, its label must be supplied. Figure 6 shows three applications of the spider addition transformation. In each case, a spider is added to the region $r = \{(\{P\}, \{Q, R\}), (\emptyset, \{P, Q, R\})\}$. In the first case, the existential spider σ_e is added. In the second and third cases, a positive and, respectively, negative spider (σ_a and $\sigma_{\overline{a}}$ resp.) is added with the label a.

Transformation 2 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. Let σ be an element that is not in $S(d)$ (i.e. a fresh spider), let c be a constant in $\mathcal{C}$, and let r be a subset of Z. We define three spider addition operations on d:*

1. $d +_e (\sigma, r) = (L, Z, ShZ, ES \cup \{\sigma\}, PS, NS, \eta \cup \{(\sigma, r)\}, \rho)$
2. $d +_p (\sigma, r, c) = (L, Z, ShZ, ES, PS \cup \{\sigma\}, NS, \eta \cup \{(\sigma, r)\}, \rho \cup \{(\sigma, c)\})$
3. $d +_n (\sigma, r, c) = (L, Z, ShZ, ES, PS, NS \cup \{\sigma\}, \eta \cup \{(\sigma, r)\}, \rho \cup \{(\sigma, c)\})$.

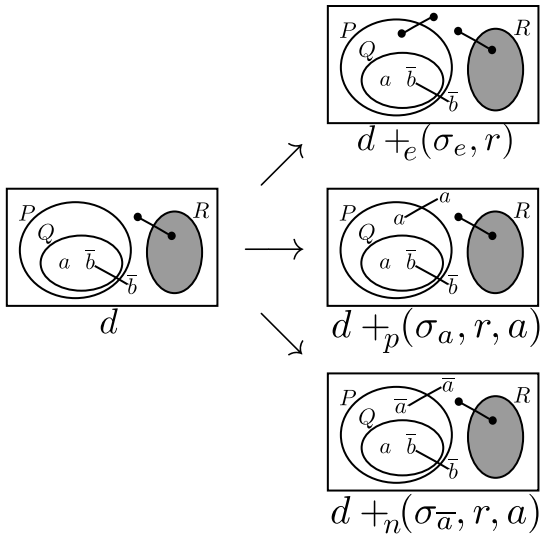


Figure 6. Adding spiders.

These transformations will now be used to define inference rules that delete spiders, shrink spiders, and swap spiders; we do not provide rules for adding spiders since such a transformation increases visual clutter and our focus is on reducing clutter. Importantly, all these rules are equivalences: the diagram to which the rule is applied has the same models as the resulting diagram. The fact that the rules are equivalences means we can explore different representations of information using different types of spider. In addition, the rules are only defined for consistent diagrams; when applied to inconsistent diagrams they need not be equivalences. In section 6 we will discuss the impact of negative spiders on clutter reduction.

Firstly we introduce three inference rules that allow spiders to be deleted. Clearly, deleting spiders allows clutter to be reduced. We start by observing that, in any diagram d, and for constant, c, in $\mathcal{C}$, the individual represented by c lies in the set represented by $Z\backslash EZ$. Moreover, the individual cannot lie in the set represented by $NZ(c, d)$, so we can make the stronger assertion that the individual lies in the set represented by $(Z\backslash EZ)\backslash NZ(c, d)$. We use this insight in the first rule, which focuses on existential spiders and exploits absence information.

For example, in figure 7, the existential spider, σ, in the zone $(\emptyset, \{P, Q\})$ can be deleted. It is the only spider in this non-shaded location and, moreover, the negative zones for c are precisely all of the zones except $(\emptyset, \{P, Q\})$. Therefore, on deleting σ the information that $(\emptyset, \{P, Q\})$ does not represent the empty set, which is directly asserted by σ, can be deduced from the three negative spiders labelled c. Neither of the other two existential spiders can be deleted. Deleting the existential spider in the shaded zone $(\{P\}, \{Q\})$ would not be sound: in models for d, this zone contains exactly

two elements, deleting this spider results in a diagram in which all models require this zone to contain just one element. The other existential spider, which is placed in two zones, tells us that there is some element in the respective set that is different from the individual b. Thus, deleting this other existential spider, whilst sound, weakens information and does not result in a semantically equivalent diagram.

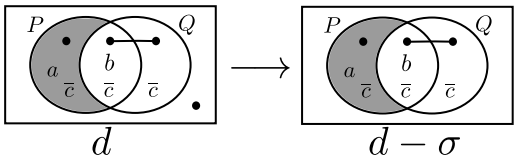


Figure 7. Delete existential spider.

Inference Rule 1 (Delete Existential Spider) *Let* $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ *be a consistent enhanced spider diagram. Let* σ *be an existential spider in* d*. If*

1. *there are no other existential or positive spiders in* d*, whose location overlaps with* σ*'s, that is for all* σ' *in* $(ES \cup PS) \backslash \{\sigma\}$*,* $\eta(\sigma) \cap \eta(\sigma') = \emptyset$*,*
2. σ*'s location does not include shaded zones, that is*

$$\eta(\sigma) \cap ShZ = \emptyset,$$

and

3. *some constant,* c*, must represent an individual in the set denoted by* $\eta(\sigma)$*, that is*

$$(Z \backslash EZ) \backslash NZ(c, d) \subseteq \eta(\sigma).$$

then d *may be replaced by* $d - \sigma$*.*

Intuitively, the delete existential spider rule can be applied when we know the element represented by c is in the set represented by $\eta(\sigma)$. However, it is important that no other existential or positive spiders have a location that overlaps with σ, essentially because negative spiders provide no distinctness information. The next rule, which allows the deletion of a positive spider is similar: a positive spider can be deleted when the information it provides is given by negative spiders:

Inference Rule 2 (Delete Positive Spider) *Let* $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ *be a consistent enhanced spider diagram. Let* σ *be a positive spider in* d*. If*

1. *there are no other existential or positive spiders in* d*, whose location overlaps with* σ*'s, that is for all* σ' *in* $(ES \cup PS) \backslash \{\sigma\}$*,* $\eta(\sigma) \cap \eta(\sigma') = \emptyset$*,*
2. *the only shaded zones in* σ*'s location are in* $NZ(\rho(\sigma), d)$*, that is*

$$\eta(\sigma) \cap ShZ \subseteq NZ(\rho(\sigma), d),$$

and

3. *the negative zones for* $\rho(\sigma)$ *indicate that* σ *must represent an individual in the set denoted by* $\eta(\sigma)$*:*

$$(Z \backslash EZ) \backslash NZ(\rho(\sigma), d) \subseteq \eta(\sigma).$$

then d *may be replaced by* $d - \sigma$*.*

Interestingly, negative spiders can always be deleted when they have multi-zone locations: if a negative spider, σ, has location $\{z_1, z_2\}$ for example, then this spider expresses that $\psi(\rho(\sigma)) \notin \Psi(z_1)$ or $\psi(\rho(\sigma)) \notin \Psi(z_2)$ which is trivially true in any interpretation. Also, just as positive spiders could be deleted when their informational content was represented by negative spiders, negative spiders can be deleted when their information is provided by positive spiders, by shading or, even, by other negative spiders.

To illustrate, in figure 8 the negative spider, σ, labelled b is deleted. This deletion does not weaken information since the positive spider labelled b provides the same absence information, albeit in a different form: from the positive spider, we can deduce that b is absent from the set represented by $(\{Q\}, \{P\})$. In fact, from d, any one of the negative spiders can be deleted. In the case of $\overline{a}$, the zone in which it is located is shaded and contains no part of any other spider: it is an empty zone. Therefore, the shading alone tells us that a does not lie in the set represented by $(\{P\}, \{Q\})$, so deleting $\overline{a}$ loses no information. In the remaining case, there are two $\overline{c}$s occupying the same zone and either one of them (but not both) can be deleted whilst preserving the informational content of d.

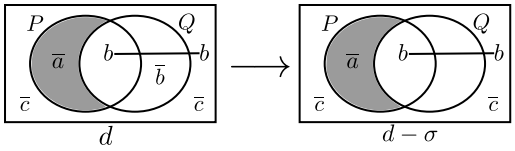


Figure 8. Delete negative spider.

Inference Rule 3 (Delete Negative Spider) *Let* $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ *be a consistent enhanced spider diagram. Let* σ *be a negative spider in* d*. If*

1. *there is not a unique zone in* $\eta(\sigma)$*, that is*

$$|\eta(\sigma)| \neq 1$$

or

2. *there is a unique zone,* z*, in* $\eta(\sigma)$*, so* $|\eta(\sigma)| = 1$*, where either*
   (a) *the zone* z *is empty, that is* $z \in EZ(d)$*,*
   (b) *there exists a positive spider,* σ'*, where* $\rho(\sigma') = \rho(\sigma)$*, whose location does not include* z*, that is* $z \notin \eta(\sigma)$*, or*

(c) there exists a negative spider, σ', where $\rho(\sigma') = \rho(\sigma)$, whose location is z, that is $\{z\} = \eta(\sigma')$

then d may be replaced by $d - \sigma$.

Another way to reduce clutter arising from spiders is to remove zones from their locations. Focusing first on existential spiders, sometimes their locations can be shrunk when we have information provided by negative spiders. However, we can never remove shaded zones from their locations, as this would reduce the upper bound placed on the cardinality of the associated set and, thus, not be sound. Again, when defining this rule we must be mindful of the fact that negative spiders do not provide distinctness information: carelessly removing a zone from an existential spider's location could reduce the associated lower bound on set cardinality and would not result in an equivalent diagram.

Figure 9 illustrates how we can shrink an existential spider. Here, in d we can see that c must lie in the set represented by $(\{P, Q\}, \emptyset)$, due to the negative zones for c and the fact that c cannot be in $(\{P\}, \{Q\})$ due to the shading (this shaded zone is empty). Therefore, we can shrink the existential spider, removing the zone $(\{Q\}, \{P\})$ from its location without weakening information.

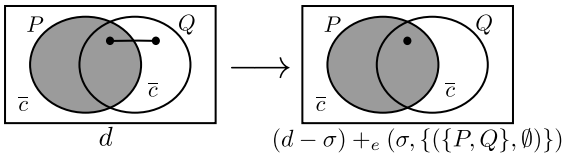


Figure 9. Shrink existential spider.

Inference Rule 4 (Shrink Existential Spider) *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be a consistent enhanced spider diagram. Let σ be an existential spider in d occupying at least two zones of which one, z, is not shaded. If*

1. *there are no other existential or positive spiders in d, whose location overlaps with σ's, that is for all σ' in $(ES \cup PS) \backslash \{\sigma\}$, $\eta(\sigma) \cap \eta(\sigma') = \emptyset$, and*
2. *some constant, c, must represent an individual in the set denoted by $\eta(\sigma) \backslash \{z\}$, that is*

$$(Z \backslash EZ) \backslash NZ(c, d) \subseteq \eta(\sigma) \backslash \{z\}.$$

Then d may be replaced by $(d - \sigma) +_e (\sigma, \eta(\sigma) \backslash \{z\})$.

In the above rule, we know that the set represented by $\eta(\sigma) \backslash \{z\}$ must contain $\psi(c)$ so it is not empty. It is therefore possible to shrink σ, removing z, without weakening information, in part since z is not shaded and in part since no other existential or positive spider has a location that overlaps with σ. We can also shrink positive spiders, when their locations include a negative zone.

Inference Rule 5 (Shrink Positive Spider) *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be a consistent enhanced spider diagram. Let σ be a positive spider in d occupying at least two zones where $\eta(\sigma) \cap NZ(\rho(\sigma), d) \neq \emptyset$. Let $z \in \eta(\sigma) \cap NZ(\rho(\sigma), d)$. Then d may be replaced by $(d - \sigma) +_p (\sigma, \eta(\sigma) \backslash \{z\}, \rho(\sigma))$ and vice versa.*

It is also possible to define a rule for shrinking negative spiders (when they include at least three zones in their locations). However, we have already seen that negative spiders with multiple zone locations can be deleted without losing information. Therefore we do not need a shrink negative spider inference rule in order to explore clutter reduction.

Lastly, we consider when it is possible to swap between different types of spider. Sometimes it is sound to swap an existential spider for a positive spider with the same location but this has no material impact on clutter so we omit this case. It is not sound to swap an existential spider, σ, for negative spiders as this would either reduce the lower bound on the set denoted by the location of σ or introduce new information about some specific individual. Therefore, there is only one interesting case where we can swap between types of spider: swapping between positive and negative spiders can alter the visual clutter in enhanced spider diagrams.

Swapping spiders is illustrated in figure 10, where the positive spider, σ, namely $a - a - a$, is swapped for two negative spiders, σ_1 and σ_2. These negative spiders occupy the two non-empty (single zone) regions $r_1 = \{(\{Q, R\}, \{P\})\}$ and $r_2 = \{(\{Q\}, \{P, R\})\}$ that did not previously contain negative a spiders. It is clear to see the information that a is in the set $U \backslash (Q \cup R)$ is not lost when this swap is performed.

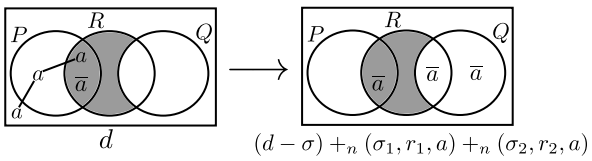


Figure 10. Swapping spiders.

Inference Rule 6 (Swap Positive and Negative Spiders) *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be a consistent enhanced spider diagram. Let σ be a positive spider in d where*

1. *there are no other existential or positive spiders in d, whose location overlaps with σ's, that is for all σ' in $(ES \cup PS) \backslash \{\sigma\}$, $\eta(\sigma) \cap \eta(\sigma') = \emptyset$, and*
2. *the only shaded zones in σ's location are in $NZ(\rho(\sigma), d)$, that is*

$$\eta(\sigma) \cap ShZ \subseteq NZ(\rho(\sigma), d).$$

Let σ_1, ..., σ_m be m fresh spiders, one for each zone in

$$Z\backslash(\eta(\sigma) \cup NZ(\rho(\sigma), d) \cup EZ) = \{z_1, ..., z_m\}.$$

Then d may be replaced by

$$(d - \sigma) +_n (\sigma_1, \{z_1\}, \rho(\sigma)) +_n .... +_n (\sigma_m, \{z_m\}, \rho(\sigma))$$

and vice versa.

6. Measuring and Reducing Clutter

The clutter measure given in [3] readily generalizes to enhanced spider diagrams. At the drawn diagram level, the measure counts the number of nodes and the lines used to connect spider nodes[2]. For each spider, the number of lines is one less than the number of zones in its location. For example, in figure 2, d_3 has a clutter score of 13, since there are seven nodes and six connecting lines, whereas d_4 has a score of 1. In figure 4, d_7 has a score of 0 but the semantically equivalent diagram, d_8, has a score of 11.

Definition 9 *Let $d = (L, Z, ShZ, ES, PS, NS, \eta, \rho)$ be an enhanced spider diagram. The **spider clutter score** for d, denoted $\mathcal{SCS}(d)$, is*

$$\mathcal{SCS}(d) = \sum_{\sigma \in S(d)} (2|\eta(\sigma)| - 1).$$

We now demonstrate how the rules can impact the clutter score. In figure 11, the first three rows illustrate the three deletion rules. Deleting the existential spider located in two zones in d_9 to give d_{10} reduces the score by 3. This spider can only be deleted, without losing information, since the negative spiders express that a is absent from Q. The region outside Q therefore 'contains' a and it is precisely this region that contains the existential spider that is deleted to give d_{10}. Deleting the positive spider, a, from d_{11} to give d_{12} reduces the clutter score by 5. This time, the information provided about the individual a by the positive spider can be inferred from the negative spiders, $\overline{a}$, and so a is redundant. These two rules clearly show that absence information, provided by negative spiders, leads to a reduction in visual clutter. When it comes to deleting negative spiders, there are various cases illustrated in d_{13} and d_{14}. Deleting $\overline{a} - \overline{a}$ reduces the score by 3, deleting two $\overline{b}$s by 2 and one $\overline{c}$ by 1 (total: 6).

The next two rows show applications of shrinking rules. Without absence information, it is not possible to shrink spiders and maintain the semantic information. In d_{15}, we can infer that $c \in P\backslash Q$, since $c \notin Q\backslash P$ and $c \notin \overline{P \cup Q}$. Thus,

[2]Recall that this measure of clutter was empirically evaluated in [16] where it was found that high levels of clutter resulted in worse task performance.

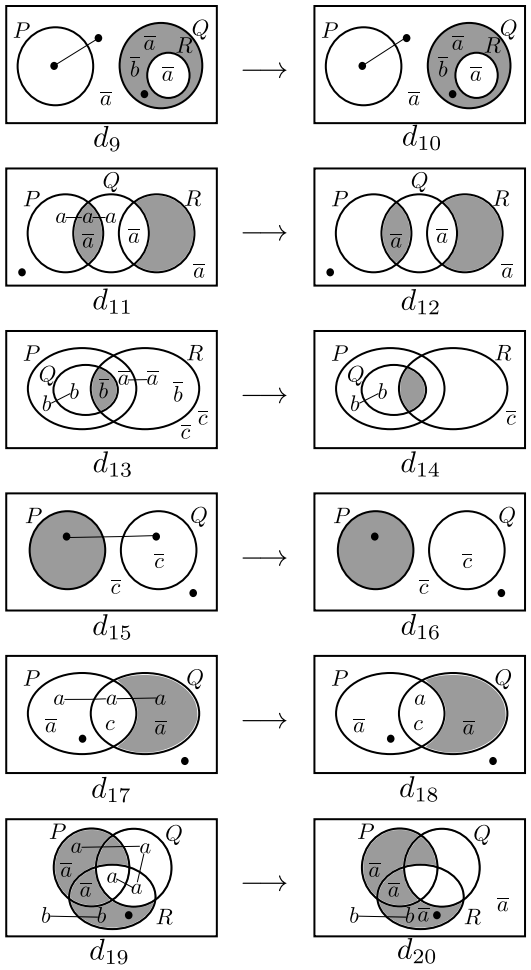


Figure 11. Rules' impact the clutter score.

the existential spider in d_{15} must represent the same element as c, and be in P. Therefore we can remove a zone from its location, as shown in d_{16}, reducing the score by 2. The case for d_{17} is more straightforward: the absence information about a allows us to reduce the location of the $a - a - a$ spider, lowering the clutter score by 4 after two applications of the shrink positive spider rule.

Lemma 3 *A single application of any one of the three deletion rules reduces the clutter score by $2|\eta(\sigma)| - 1$ where σ is the deleted spider.*

Lemma 4 *A single application of any one of the two shrinking rules reduces the clutter score by 2 where σ is the shrunk spider.*

The last row of figure 11 shows an application of the swap rule. The $a - a - a - a$ spider is swapped for two $\overline{a}$ spiders, reducing the clutter score by 5.

Lemma 5 *A single application of the swap rule reduces clutter whenever*

$$2|\eta(\sigma)| - 1 > |Z \backslash (\eta(\sigma) \cup NZ(\rho(\sigma), d) \cup EZ)|,$$

where σ is the positive spider to be swapped.

We now discuss how clutter reduction using absence, conveyed via negative spiders, contrasts with the Venn-i^e case. In enhanced spider diagrams, many of the rules required the subject existential or positive spider, σ, to be the only one in its location. This is because to delete, shrink or swap such a spider, absence information must be used, yet negative spiders do not provide distinctness information. Considering an absence spider in isolation, it only indicates the set in which the represented individual does not lie and, therefore, in which set it does lie. Of course, this is identical for $\overline{i}$-sequences in Venn-i^e. However, a significant difference is that, in Venn-i^e, $\otimes$-sequences and i-sequences do not represent distinct elements, so Venn-i^e's inference rules are much less restrictive. As a consequence, there are many more situations in which information about the absence of individuals can be used to reduce clutter. A less diagrammatic interpretation of absence, whereby negative spiders *do* assert that the represented individual is distinct from those represented by the other spiders, would lead to the ability to reduce clutter further.

7. Conclusion

We have introduced enhanced spider diagrams, which include syntax for visually representing the absence of individuals from particular sets directly. Necessary and sufficient conditions for diagram satisfiability were given, since unsatisfiable diagrams are semantically equivalent to diagrams with a zero clutter score. Following from this, we defined inference rules that permitted clutter to be reduced in satisfiable diagrams. Interestingly, the level of clutter reduction that is possible in this system is not as dramatic as was seen for the Venn-i^e system, where i-sequences and $\overline{i}$-sequences could readily be swapped to alter clutter. As indicated, an alternative, less well-matched, interpretation of negative spiders could lead to larger reductions in visual clutter. However, it is unclear whether lower clutter, with a less well-matched syntax, or higher clutter with a well-matched syntax is most effective for human cognition. Such a trade-off should be explored as it could provide important insight into choices that must be made when designing diagrammatic systems.

Acknowledgement This research was conducted whilst Gem Stapleton was at Jadavpur University as Visiting Professor.

References

[1] M. Alqadah, G. Stapleton, J. Howse, and P. Chapman. Evaluating the impact of clutter in Euler diagrams. In *8th Int. Conf. on Diagrams*, pages 109–123. Springer, 2014.

[2] R. Bhattacharjee, M. Chakraborty, and L. Choudhury. Venn diagram with names of individuals and their absence: A non-classical diagram logic. *Logica Universalis*, pages 1–66, 2018.

[3] J. Burton, M. Chakraborty, L. Choudhury, and G. Stapleton. Minimizing clutter using absence in Venn-i^e. In *9th Int. Conf. on Diagrams*, pages 107-122. Springer, 2016.

[4] L. Choudhury and M. K. Chakraborty. On extending Venn diagrams by augmenting names of individuals. In *3rd Int. Conf. on Diagrams*, pages 142–146. Springer, 2004.

[5] J. Gil, J. Howse, and S. Kent. Formalising spider diagrams. In *IEEE Symposium on Visual Languages (VL99), Tokyo*, pages 130–137. 1999.

[6] C. Gurr. Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *J. Visual Languages and Computing*, 10(4):317–342, 1999.

[7] L. Horn. *A Natural History of Negation*. CSLI Lecture Notes. Center for the Study of Language and Information, 2001.

[8] T. Hou, P. Chapman, and A. Blake. Antipattern comprehension: An empirical evaluation. In *9th Int. Conf. on Formal Ontology in Information Systems*, pages 211–224, 2016.

[9] T. Hou, P. Chapman, and I. Oliver. Measuring perceived clutter in concept diagrams. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 31–39. 2016.

[10] J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS J. Computation and Mathematics*, 8:145–194, 2005.

[11] J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: A case study. In *International Semantic Web Conference*, pages 257–272. Springer, 2011.

[12] C. John, A. Fish, J. Howse, and J. Taylor. Exploring the notion of clutter in Euler diagrams. In *4th Int. Conf. on Diagrams*, pages 267–282, 2006. Springer.

[13] C. Peirce. *Collected Papers*, volume 4. Harvard University Press, 1933.

[14] Y. Sato, K. Mineshima, and R. Takemura. The Efficacy of Euler and Venn Diagrams in Deductive Reasoning: Empirical Findings. In *6th Int. Conf. on Diagrams*, pages 6–22. Springer, 2010.

[15] S.-J. Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1994.

[16] G. Stapleton, A. Blake, J. Burton, and A. Touloumis. Presence and absence of individuals in diagrammatic logics: An empirical comparison. *Studia Logica*, 105(4):787815, 2017.

[17] G. Stapleton, J. Taylor, J. Howse, and S. Thompson. The expressiveness of spider diagrams augmented with constants. *J. Visual Languages and Computing*, 20:30–49, 2009.

[18] N. Swoboda and G. Allwein. Using DAG transformations to verify Euler/Venn homogeneous and Euler/Venn FOL heterogeneous rules of inference. *J. Software and System Modeling*, 3(2):136–149, 2004.

An Edge-based Graph Grammar Formalism and Its Support System

Xiaoqin Zeng[1] Yufeng Liu[1] Zhan Shi[1] Yingfeng Wang[2] Yang Zou[1] Jun Kong[3] Kang Zhang[4]

[1]Institute of Intelligence Science and Technology, Hohai University, Nanjing, Jiangsu, China

[2]School of Information Technology, Middle Georgia State University, Macon, GA 31206, USA

[3]Department of Computer Science, North Dakota State University, Fargo, ND 58102, USA

[4]Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA

***Abstract*—As a useful formal tool, graph grammar provides a rigorous but intuitive way for defining graphical languages and analyzing graphs. This paper presents a new context-sensitive graph grammar formalism called Edge-based Graph Grammar or EGG, in which a new methodology is proposed to tackle issues, such as the embedding problem, the membership problem and the parsing algorithm. It presents the formal definitions of EGG and its language. Then, a new parsing algorithm is given for checking the structural correctness or validity of a given host graph. The paper finally describes the development of an EGG support system with friendly GUI.**

Keywords-component; graph grammar; graphical language; embedding problem; parsing; production rule

I. INTRODUCTION

With the development of human-computer interaction techniques, graphical languages have been applied to various application domains, such as modeling visual interaction processes [1, 2], designing graphical user interface in multimedia applications [3], visual queries to databases [4], and defining the layout of a GUI in multimedia applications [3]. Conceptually, objects described by graphical languages can be abstracted as graphs consisting of nodes and edges. For the specification and analysis of these types of graphs, graph grammars [5, 6] are an ideal formal and intuitive tool.

It is well-known that formal string grammar lays a solid theoretical foundation for the definition and parsing of programming languages. For the same reason, graphical languages also need the corresponding formal graph grammars. In view of the theoretical role played to string languages by string grammars, graph grammars set a theoretical basis to visual languages [7]. However, the implementation of a graphical language is usually not as easy as implementing string languages [8]. This is mostly due to the fact that the extension from one-dimensional string grammars to two-dimensional graph grammars raises new issues [9] such as the embedding problem, the membership problem, high parsing complexity.

There have been a number of graph grammars and their applications in the literature [10-27]. According to the type of grammatical productions, graph grammars could be mainly divided into two categories: context-free and context-sensitive. The main differences between the two are the production formation and the expressive power. On the one hand, a context-free grammar requires that only a single non-terminal node be allowed on the left-hand side of a production [16]. In early years, many context-free grammars were proposed [17-21]. Since the productions of these graph grammars are quite simple, their expressive power is limited, which hinders the scope of their applications. On the other hand, in response to the increasing demands of intricate graph-oriented applications, researchers have developed several context-sensitive graph grammars, such as PLC (picture layout grammar) [21], CMG (constrain multiset grammar) [22], LGG (layered graph grammar) [23], RGG (reserved graph grammar) [8], SGG (spatial graph grammar) [24, 25]. These context-sensitive graph grammars allow the left-hand side of a production to be a graph rather than a node, so bring more expressive power. LGG and RGG are the most representatives of context-sensitive graph grammars.

Rekers and Schürr [23] proposed a context-sensitive graph grammar formalism called *Layered Graph Grammar* (LGG) for defining and parsing graphical visual languages. To solve the embedding problem, LGG puts a restriction on the definition of a redex in a host graph by requiring its nodes that are isomorphic to non-context nodes in productions can only link to other nodes in the host graph that are isomorphic to the context nodes in the productions. This restriction ensures no creation of dangling edges when a redex in a host graph is replaced.

Based and improved on LGG, Zhang et al. [8] proposed another context-sensitive grammar called *Reserved Graph Grammar* (RGG), which defines the structure of graphs by introducing a two-level structure for each node as a super-vertex containing sub-vertices connected with edges. With the introduction of selection-free productions to graph grammars, a

DOI reference number: 10.18293/DMSVIVA2018-015

Selection-Free Parsing Algorithm (SFPA) is designed for a selection-free RGG, which only needs to consider one parsing path and thus can efficiently parse graphs with polynomial time complexity [8]. Later on, Kong et al. [24, 25] extended RGG by introducing spatial notations and mechanisms. The spatial specifications of the extended RGG, called *Spatial Graph Grammar* (SGG), can qualitatively express the spatial relationships among objects and reduce the parsing complexity using the spatial information.

Both LGG and RGG have been applied widely to the definition, analysis and transformation of visual languages [28-37], such as Visual XML Schemas [29, 30], Design Pattern Evolution and Verification [32, 33], Generic Visual Language Generation Environments [28]. However, they still have deficiencies. For example, the LGG's context nodes and layer decomposition constraint make productions difficult to design and parsing algorithm complicated to implement with high time complexity. RGG's two-level node structure and marking mechanism are not intuitive and make them difficult to apply to general graphs.

This paper presents our work on the improvements over the existing graph grammars with the following contributions.

- A new context-sensitive graph grammar formalism called EGG, which uses edges instead of nodes to concisely express the context in productions for simply and efficiently solving the embedding problem.
- A size-increasing constraint applied to the structure of productions for solving the membership problem, easing the design of productions.
- A general parsing algorithm for checking the structural correctness and validity of given host graphs; and the implementation of an EGG graph grammar support system, which provides friendly GUI for end users to design and apply graph grammars.

The rest of the paper is organized as follows. Section 2 presents graphical and grammatical preliminaries, introducing new terms used in Section 3, which gives the formal definitions of EGG and its language. Section 4 presents a parsing algorithm. Section 5 describes the developed EGG support system. Finally, Section 6 concludes the paper.

II. Graphical and Grammatical Preliminaries

In node-edge graphs, a node typically represents an abstract object and an edge represents some kind of relationship between two connected nodes. Each node n in a node set N can be connected with none or more edges, and each edge e in an edge set E is only connected with two nodes. An edge can be directed or undirected depending on whether it has a direction between the two connected nodes. Because an undirected edge can be treated as two directed edges with reverse directions, without loss of generality this paper only considers directed edges.

In string grammars, labels play an important role as identifiers, and so do labels in graph grammars. Let L be a finite set of labels. Depending on the usage of a label, L can further be divided into terminal label set L_T, nonterminal label set L_{NT}, and mark label set L_M, namely $L = L_T \cup L_{NT} \cup L_M$, $L_T \cap L_{NT} = \Phi$, and $L_M \cap (L_T \cup L_{NT}) = \Phi$.

By combining the techniques of both graph theory and formal language, we introduce a series of new definitions and notations here.

Definition 2.1 n is a node with label l in a given finite label set L.

Definition 2.2 $\mathrm{e} = (\mathrm{n_s}, \mathrm{n_e})$ is a directed edge, where

- $\mathrm{n_s}$ is the start node of the edge;
- $\mathrm{n_e}$ is the end node of the edge.

Based on the above definitions of node and edge, we further introduce the following notations:

- $\mathrm{E_s}$ is a set of edges starting from a node;
- $\mathrm{E_e}$ is a set of edges ending to a node;
- $\mathrm{d_s(n)}$ is the out-degree indicating the number of edges starting from n, i.e. $\mathrm{d_s(n)} = |\mathrm{E_s}|$;
- $\mathrm{d_e(n)}$ is the in-degree indicating the number of edges ending to n, i.e. $\mathrm{d_e(n)} = |\mathrm{E_e}|$.

For simplicity, notations like $n.l$ and $n.E_s$ express the corresponding components of node n, and are applicable to other definitions throughout this paper.

Unlike an undirected edge, a directed edge needs to distinguish start node and end node. Besides, an edge may also carry a label for clear identification.

Definition 2.3 $\mathrm{G} = (\mathrm{N}, \mathrm{E})$ is a graph on given label set L, where

- N is a node set containing terminal and nonterminal nodes, i.e., $\mathrm{N} = \mathrm{N_T} \cup \mathrm{N_{NT}}$,
- E is an edge set with $\mathrm{E} \subseteq \mathrm{N} \times \mathrm{N}$.

We then have the following mappings for mathematically expressing grammatical items.

- $\mathrm{f_{NL}}: \mathrm{N} \to \mathrm{L}$, a mapping from node n to label $\mathrm{l} \in \mathrm{L}$, i.e., $\mathrm{f_{NL}(n)} = \mathrm{n.l}$;
- $\mathrm{f_{EN_s}}: \mathrm{E} \to \mathrm{N}$, a mapping from edge e to its start node, i.e., $\mathrm{f_{EN_s}(e)} = \mathrm{e.n_s}$;
- $\mathrm{f_{EN_e}}: \mathrm{E} \to \mathrm{N}$, a mapping from edge e to its end node, i.e., $\mathrm{f_{EN_e}(e)} = \mathrm{e.n_e}$.

In EGG, dangling edge set $\dot{E}$ is introduced to represent contexts, in which each edge is connected with only one node being either a start or end node, namely $\dot{E} = \dot{E}_s \cup \dot{E}_e$ with $\dot{E}_s = \{\dot{e}_s | \dot{e}_s = (n_s, \emptyset)\}$, $\dot{E}_e = \{\dot{e}_e | \dot{e}_e = (\emptyset, n_e)\}$ and $\dot{E}_s \cap \dot{E}_e = \Phi$. In addition to dangling edges, a marking mechanism is also introduced to mark dangling edges. The concepts of dangling edge and marking mechanism solves the embedding problem in EGG. Fig. 1 illustrates a graph including dangling edges with $\dot{E} = \{1,2,3\}$. The graph is called a *dangling edge graph* and can be defined as follows.

Definition 2.4 $\overline{\mathrm{G}} = (\mathrm{N}, \overline{\mathrm{E}}, \mathrm{M})$ is a *dangling edge graph* on given label set L, in which,

- N is a node set;
- $\overline{\mathrm{E}}$ is an edge set including dangling edges, i.e., $\overline{\mathrm{E}} = \mathrm{E} \cup \dot{\mathrm{E}}$;
- $\mathrm{M} \subseteq \mathrm{L_M}$ is a mark set for marking dangling edges to distinguish different contexts.

Essentially, $\overline{G}$ is an extension of G by introducing dangling edges, and G can be regarded as a special case of $\overline{G}$. Similarly, there is an extra mapping as follows.

- $f_{EM}: \dot{E} \to M$, an injective mapping from a dangling edge $\dot{e}$ to its mark m, i.e., $f_{EM}(\dot{e}) = m$.

Note that dangling edge set $\dot{E}$ may be empty, which leads to the empty corresponding mark set *M* and mapping f_{EM}. Based on the above defined dangling edge graph, a grammatical production can be defined as follows.

Definition 2.5 A production p is the expression $\overline{G}_L ::= \overline{G}_R$, which consists of a left dangling edge graph $\overline{G}_L$ and a right dangling edge graph $\overline{G}_R$ satisfying $\overline{G}_L.M = \overline{G}_R.M$.

In a production, dangling edges represent contexts and each pair of corresponding dangling edges between the left and right graphs are labeled by a unique mark to maintain their corresponding relationship. Using dangling edges and their corresponding marks, the replacement of a redex by either a left or right graph in a production can be done without ambiguity. Fig. 2 is an example of a set of EGG productions specifying a process flow diagram with $\{begin, assign, fork, join, send, receive, if, endif\} \subseteq L_T$ and $\{stat\} \subseteq L_{NT}$.

The function of a production is to transform a graph to another graph. However, the transformation needs to satisfy some conditions in which isomorphism is fundamental.

Definition 2.6 Graphs G and Q are isomorphic, denoted as $G \approx Q$, f_{NL} and f_{NL}' are two mappings for G and Q respectively, if and only if there exist two bijective mappings $f_{NN}: G.N \leftrightarrow Q.N$ and $f_{EE}: G.E \leftrightarrow Q.E$, and the following are satisfied:

- $\forall n(((n \in G.N) \vee (n \in Q.N)) \to (f_{NL}(n) = f_{NL}'(f_{NN}(n))))$;
- $\forall e(((e \in G.E) \vee (e \in Q.E)) \to (f_{NN}(f_{EN_s}(e)) = f_{EN_s}(f_{EE}(e))) \wedge (f_{NN}(f_{EN_e}(e)) = f_{EN_e}(f_{EE}(e))))$.

An isomorphism between two graphs means that their corresponding nodes have the same label, and the same out-degree and in-degree. In addition, the corresponding edges have the same start and end nodes.

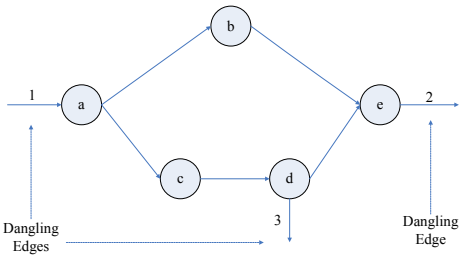


Figure 1. A dangling edge graph

Definition 2.7 Graph Q is the sub-graph of G, denoted as $Q \in Sub(G)$, if and only if the following are satisfied:

- $(Q.N \subseteq G.N) \wedge (Q.E \subseteq G.E)$.

Graph Q is the sub-graph of G means that Q is part of G.

Definition 2.8 Graph Q is the core graph of $\overline{G}$, denoted as $Q = Cor(\overline{G})$, if and only if the following is satisfied:

- $(Q.N = \overline{G}.N) \wedge (Q.E = (\overline{G}.\overline{E} - \overline{G}.\dot{E}))$.

Core graph Q is the sub-graph of graph $\overline{G}$ obtained by removing all dangling edges from graph $\overline{G}$ and keeping all the nodes and non-dangling edges of graph $\overline{G}$. The graph in Fig. 3 is the core graph of that in Fig. 1.

Definition 2.9 If graph Q is a sub-graph of graph G and may include dangling edges, and $\overline{G}_{L|R}$ is a graph being left or right side of a production, Q is a redex of G with respect to $\overline{G}_{L|R}$, denoted as $Q \in Redex(G, \overline{G}_{L|R})$, if and only if there exits bijective mappings $f_{NN}: Q.N \leftrightarrow \overline{G}_{L|R}.N$ and $f_{EE}: Q.E \leftrightarrow \overline{G}_{L|R}.E$, and the following are satisfied:

- $Cor(Q) \approx Cor(\overline{G}_{L|R})$;
- $\forall n((n \in Q) \to (d_s(n) = d_s(f_{NN}(n))) \wedge (d_e(n) = d_e(f_{NN}(n))))$.

To explain the above definition, we provide an example in the following three figures. Fig. 4 is graph $\overline{G}_{L|R}$, and Fig. 5 is a given host graph G. Obviously, graph Q in Fig. 6 is the sub-graph of G. According to *Definition 2.9*, Q is a redex of G with respect to $\overline{G}_{L|R}$.

In host graph G, if there is sub-graph Q being the redex of G with respect to $\overline{G}_{L|R}$ that is a left or right side graph of a production, then one could use the right or left side graph of the production to replace Q in G. This process is called *graph transformation* or *replacement*, as formally defined below.

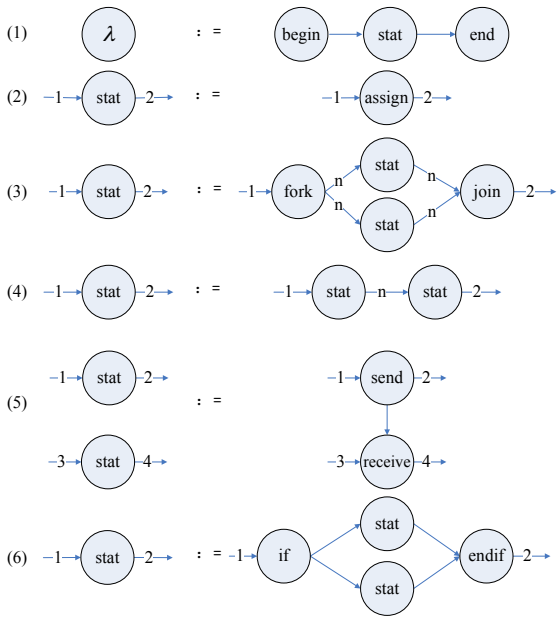


Figure 2. A set of EGG productions

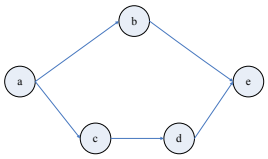


Figure 3. The core graph of the graph in Figure 1

Definition 2.10 An *L-application* to graph G is a transformation that generates graph G' using production p: $\overline{G}_L \coloneqq \overline{G}_R$, denoted as $G' = \mathrm{Tr}(G, Q, \overline{G}_L, \overline{G}_R)$, where $Q \in \mathrm{Redex}(G, \overline{G}_L)$, and $\mathrm{Cor}(\overline{G}_R)$ is used to replace Q in G. The L-application is also called *derivation* operation and denoted as $G \to^p G'$.

If a sequence of L-applications for graph G is: $G \to^{p_1} {G_1}', {G_1}' \to^{p_2} {G_2}', \ldots, {G_{n-1}}' \to^{p_n} {G_n}'$, then $G \to^* {G_n}'$ can be used to concisely express this process.

Definition 2.11 An *R-application* to graph G is a transformation that generates graph G'' using production p: $\overline{G}_L \coloneqq \overline{G}_R$, denoted as $G'' = \mathrm{Tr}(G, Q, \overline{G}_R, \overline{G}_L)$, where $Q \in \mathrm{Redex}(G, \overline{G}_R)$, and $\mathrm{Cor}(\overline{G}_L)$ is used to replace Q in G. The R-application is also called *reduction* operation and denoted as $G \mapsto^p G''$.

Similar to L-applications, a sequence of R-applications, which is $G \mapsto^{p_1} {G_1}'', {G_1}'' \mapsto^{p_2} {G_2}'', \ldots, {G_{n-1}}'' \mapsto^{p_n} {G_n}''$, can be expressed as $G \mapsto^* {G_n}''$.

Fig. 7 shows a derivation process from an initial graph using the productions in Fig. 2.

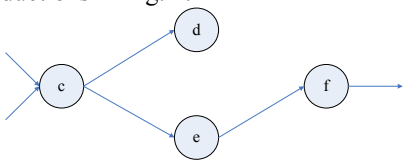


Figure 4. A graph $\overline{G}_{L|R}$ with dangling edges

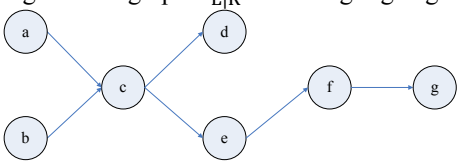


Figure 5. A host graph G

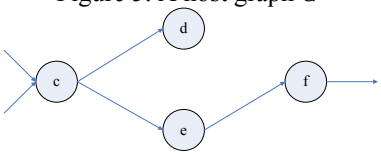


Figure 6. The sub-graph Q is a redex of G with respect to $\overline{G}_{L|R}$

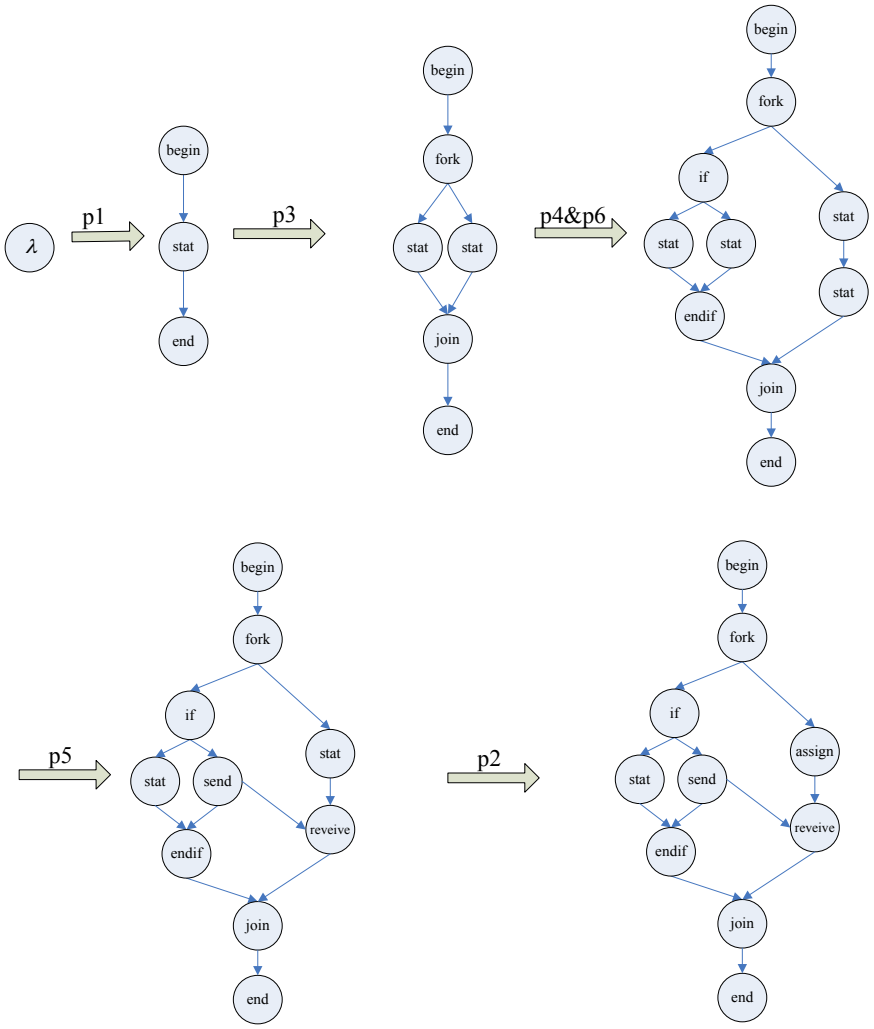


Figure 7. A graph L-application process using EGG productions

The formal definition of EGG and its language are discussed below.

III. AN Edge-based Graph Grammar Formalism

To solve the embedding and membership problems, EGG employs edges rather than nodes in the two sides of a production to directly express contexts and introduces a size-increasing constraint to ensure the decidability of EGG.

3.1 Definition of EGG and its language

Based on the definitions in Section 2.1, an edge-based context-sensitive graph grammar formalism and its language can be defined as follows.

Definition 3.1 An EGG is a 3-tuple (λ, L, P), where:

- λ is an initial graph;
- L is a label set containing terminal and non-terminal labels, i.e., $L = L_T \cup L_{NT}$;
- P is a set of productions, and each production $p \in P$ in the form of $\overline{G}_L ::= \overline{G}_R$ must satisfy the following constraints:
  (1) λ must be a left graph of a production;
  (2) $\overline{G}_R$ must be nonempty;
  (3) The size of left graph must be no more than that of right graph, i.e., $|\overline{G}_L.N| \leq |\overline{G}_R.N|$. If they are equal, the number of terminal nodes in left graph must be less than that of right graph, i.e., $|\overline{G}_L.N_T| < |\overline{G}_R.N_T|$.

Similar to string grammars, graph grammars with arbitrary graphs on the left and right sides of productions may face the membership problem, that is, their languages are not decidable in general. EGG introduces a size-increasing constraint for each production to solve the membership problem. The constraint ensures that any given host graphs can be parsed with EGG productions within a finite number of R-applications. Also, the constraint is weak with little impact on the flexibility of context-sensitive grammars and easier to implement than that of LGG and RGG for grammar designers.

Theoretically, a graph grammar is a formal tool for rigorously defining a graph language, which is a set of graphs that can be derived from the initial graph. Below is the formal definition of a graph language.

Definition 3.2 Let $egg = (\lambda, L, P)$ be a grammar of EGG, its language $\Gamma(egg)$ can be formally defined as $\Gamma(egg) = \{G | (\lambda \rightarrow^* G) \wedge (f_{NL}(G.N) \subseteq L_T)\}$.

Practically, a graph grammar is a useful tool for automatically analyzing graphs' validity. If a given graph can

be reduced to the initial graph with a finite series of R-applications of a graph grammar, this graph is regarded as belonging to the grammar's language. Otherwise, the graph does not belong to the graph language or the graph grammar is not decidable.

IV. PARSING ALGORITHM OF EGG

Generally, a graph grammar needs to be equipped with a parsing mechanism for automatically checking whether a given graph, called *host graph*, is structurally correct or valid with respect to the graph language defined by the grammar. This section presents a parsing algorithm, which checks if a host graph can be reduced to the initial graph by applying the EGG grammar's productions to perform a series of R-applications.

```
Parsing (Graph G, ProductionSet P)
{
loop-1:  while (G ≠ λ)
   {
            DELIMITER → RedexStack;
            // push
loop-2:     for all p ∈ P
            {
               RedexSet = FindRedexForRight(G, p.G_R);
loop-3:              for all Redex ∈ RedexSet;
                        (Redex, p)→ RedexStack;
            // push
             }
            (Redex, p) ← RedexStack;
            // pop
loop-4:     while (Redex = DELIMITER)
            {
               If (HostStack != NULL ⋀ RedexStack != NULL)
              G← HostStack;
              // pop
 (Redex, p) ← RedexStack;
 // pop
               else
              return("Invalid");
             }
             HostStack ←G ;
            // push
                 G = RightApplication(G, Redex, p);
          }
          return("Valid");
}
```

To trace all possible R-application paths starting from a given host graph, a mapping between a redex and its host graph is needed. As such a mapping is usually many to one, the tracing employs two stacks to separately store the redexes found and the intermediate host graph yielded, and employs a delimiter in the redex stack to delimit a group of redexes that correspond to the same host graph. The delimiter makes the correspondence manageable by synchronizing the contents in the two stacks. The function takes a graph and a set of productions as input and returns a definite answer indicating whether the graph is valid or not.

V. IMPLEMENTATION OF AN EGG SUPPORT SYSTEM

A graph grammar support system is a software platform that can be helpful for end users to easily use graph grammars. This section briefly describes the architecture and functions of an EGG support system, abbreviated as EGGSS.

Fig. 9 illustrates the end user view of EGGSS. From a user point of view, EGGSS supplies, besides normal GUI of Windows, extra graphical and grammatical tools to assist the user to draw graphs, design graph productions, define graph languages, perform graph transformations and parse graphs. Fig. 10 is an example window of EGGSS's user interface, where the upper row is the main menu with all operational items including not only graphical and grammatical operations but also other Window GUI operations. On the left, a tree view allows users to manage XML files with saving, accessing and deleting operations. They can read graphs in XML format from the memory and save graph data to an XML file. On the right, the upper part shows an edited host graph and the lower part shows a designed production. Fig. 11 shows the system architecture with relevant modules. In the architecture, three upper layers are implemented using C++ in the environment of Visual Studio 2005, while two lower layers are implemented using the existing XML open sources and software tools.

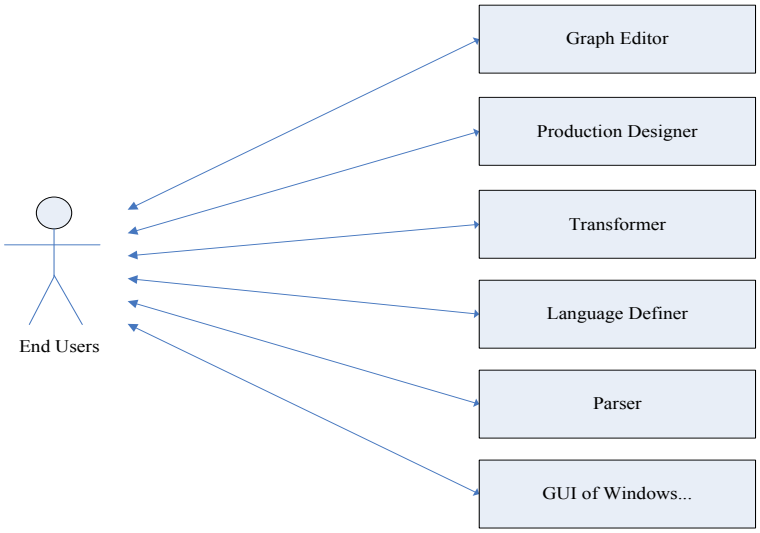


Figure 9. End user view of EGGSS

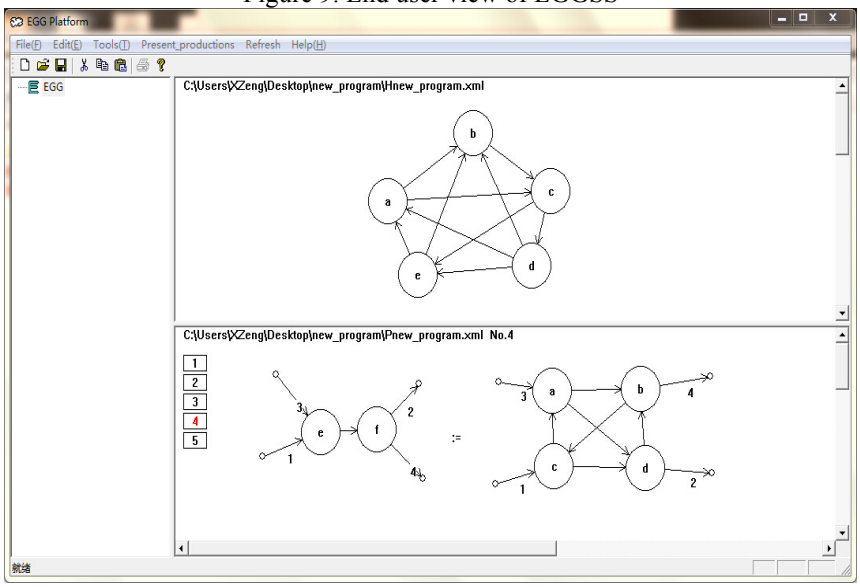


Figure 10. A window of EGGSS's user interfac

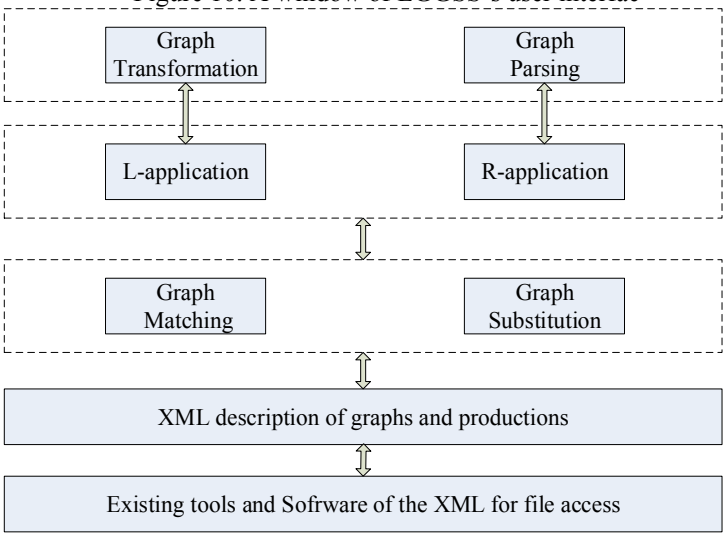


Figure 11. The architecture of EGGSS

I. CONCLUSIONS

This paper has proposed a new graph grammar formalism, namely EGG, which aims at making the design and implementation of a graph grammar simple without weakening the expressive power of the grammar. The proposed EGG lays a solid foundation for a wide range of applications using graph grammars. Specifically, EGG focuses on tackling general graph languages and graph transformations with productions as simple as possible. First, EGG simplifies the expression of productions, in which the context nodes are eliminated and only edges linked to context nodes are kept. In this way, the structural information of graphs is still kept. Second, using dangling edges and their corresponding marks, the replacement of a redex by either a left or right graph in a production can be easily done without ambiguity. Third, the introduction of size-increase constraint to productions solves the membership problem, making EGG parsing algorithm terminable.

As a future research, we will attempt to find the way to reduce the parsing complexity, to further improve EGGSS to be friendlier for end users.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under grant 61170089.

REFERENCES

[1] P. Bottoni, S. K. Chang, M. F. Costabile, S. Levialdi, P. Mussio. On the specification of dynamic visual languages, Proc. IEEE Symposium on Visual Languages, 14-21, 1998.

[2] P. Bottoni, S. K. Chang, M. F. Costabile, S. Levialdi, P. Mussio. Modeling visual interactive systems through dynamic visual languages, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 32(6): 654-669, 2002.

[3] S. K. Chang. Extending visual languages for multimedia, IEEE Multimedia, 3(3): 18-26, 1996.

[4] S. K. Chang. A visual language compiler for information retrieval by visual reasoning, IEEE Transactions on Software Engineering, 16(10): 1136-1149, 1990.

[5] G. Rozenberg, H. Ehrig, Handbook of graph grammars and computing by graph transformation, Handb. Graph Grammars. 1 (1997) 1–8.

[6] H. Fahmy, D. Blostein, A Survey of Graph Grammars: Theory and Applications, in: IAPR Int. Conf. Pattern Recognit., 1992: pp. 294–298.

[7] C. Ermel, M. Rudolf, G. Taentzer, The AGG Approach: Language and Environment, in: Handb. Graph Grammars 2, 1999: pp. 551-603.

[8] D.-Q. Zhang, K. Zhang, J. Cao, A context-sensitive graph grammar formalism for the specification of visual languages, Comput. J. 44 (2001)

[9] X.-Q Zeng, K. Zhang, J. Kong, G.-L. Song, RGG+: An enhancement to the reserved graph grammar formalism, in: Proc. 2005 IEEE Symp. Vis. Lang. Human-Centric Comput., 2005: pp.272–274.

[10] D. Goik, K. Jopek, M. Paszyński, A. Lenharth, D. Nguyen, K. Pingali, Graph grammar based multi-thread multi-frontal direct solver with Galois scheduler, in: Procedia Comput. Sci., 2014: pp.960–969.

[11] L. Fürst, M. Mernik, V. Mahnič, Converting metamodels to graph grammars: doing without advanced graph grammar features, Softw. Syst. Model 14 (2013) 1297–1317.

[12] J. Heinen, C. Jansen, J.-P. Katoen, T. Noll, Verifying pointer programs using graph grammars, Sci. Comput. Program. 1 (2013) 7–12.

[13] Z. Shi, X.-Q. Zeng, S. Huang, H. Li, Transformation between BPMN and BPEL based on graph grammar, in: Proc. 5th Int. Conf. Comput. Commun. Netw. Technol., 2014: pp.1-6.

[14] F. Hermann, S. Gottmann, N. Nachtigall, H. Ehrig, B. Braatz, G. Morelli, A. Pierre, T. Engel, C. Ermel, Triple Graph Grammars in the Large for Translating Satellite Procedures, Theor. Prac. Model Transforms. 8568 (2014) 122-307.

[15] Y. Ong, K. Streit, M. Henke, W. Kurth, An approach to multiscale modelling with graph grammars, Ann. Bot. 114 (2014) 813–827.

[16] K. Wittenburg, L. Weitzman, Relational grammars: theory and practice in a visual language interface for process modeling. Vis. Lang. Theor. (1998) 193-217.

[17] G. Rozenberg, E. Welzl, Boundary NLC graph grammars-Basic definitions, normal forms, and complexity, Inf. Control. 69 (1986) 136–167.

[18] D. Janssens, G. Rozenberg, Graph grammars with neighbourhood-controlled embedding, Theor. Comput. Sci. 21 (1982) 55–74.

[19] F. Drewes, H.-J. Kreowski, A. Habel, Hyperedge Replacement Graph Grammars, in: Handb. Graph Grammars 1, 1997: pp.95–162.

[20] K. Wittenburg, Earley-style parsing for relational grammars, Proc. 8th IEEE Workshop. Vis. Lang., 1992: pp. 192-199.

[21] E. Golin, A Method for the specification and parsing of visual languages, PhD Thesis, 1991, Department of Computer Science, Brown University.

[22] K. Marriott, Constraint Multiset Grammars, Proc. IEEE Symp. Vis. Lang., 1994: pp. 118–125.

[23] J. Rekers, a Schürr, Defining and parsing visual languages with layered graph grammars, J. Vis. Lang. Comput. 8 (1997) 27–55.

[24] J. Kong, K. Zhang, X.-Q. Zeng, Spatial graph grammars for graphical user interfaces, ACM Trans. Comput. Interact. 13 (2006) 268–307.

[25] J. Kong, K. Zhang, Parsing Spatial Graph Grammars, Proc. 2004 IEEE Symp. Vis. Lang. Hum. Centric Comput. (2004) 99–101.

[26] M. Decker, H. Che, A. Oberweis, P. Stürzel, M. Vogel, Modeling mobile workflows with BPMN, in: ICMB GMR 2010 - 2010 9th Int. Conf. Mob. Business/2010 9th Glob. Mobil. Roundtable, 2010: pp.272–279.

[27] C. Kim, M. Ando, Node replacement graph grammars with dynamic node relabeling, Theor. Comput. Sci. 583 (2015) 40–50.

[28] K. Zhang, D.-Q. Zhang, J. Cao, Design, construction, and application of a generic visual language generation environment, IEEE Trans. Softw. Eng. 27 (2001) 289–307.

[29] G. Song, K. Zhang, Visual XML schemas based on reserved graph grammars, in: Proc. Int. Conf. Inf. Technol. Coding and Computing, 2004: pp. 687-691.

[30] K. Zhang, D.-Q. Zhang, Y. Deng, A Visual Approach to XML Document Design and Transformation, Proc. IEEE Symp. Human-Centric Comput. Lang. Environ., 2001: pp.312–319.

[31] K.-B. Zhang, M.A. Orgun, K. Zhang, A prediction-based visual approach for cluster exploration and cluster validation by HOV3. Lec. Notes Comput. Sci. 4702 (2007) 336-349.

[32] C. Zhao, J. Kong, K. Zhang, Design pattern evolution and verification using graph transformation, Proc. 40th Annual Hawaii International Conference on System Sciences (HICSS'2007), 2007: pp.290a-290a.

[33] C. Zhao, J. Kong, J. Dong, K. Zhang, Pattern-based Design Evolution Using Graph Transformation, J. Vis. Lang. Comput. 18 (2007) 378–398.

[34] C. Zhao, J. Kong, K. Zhang, Program behavior discovery and verification: A graph grammar approach, IEEE Trans. Softw. Eng. 36 (2010) 431–448.

[35] K. Zhang, J. Kong, Exploring semantic roles of Web interface components, Proc. Int. Conf. Mach. Web Intell., 2010: pp.8-14.

[36] K. Zhang, J. Kong, M. Qiu, G. Song, Multimedia layout adaptation through grammatical specifications, Multimedia Syst. 10 (2005) 245-260.

[37] J. Kong, K.-L. Ates, K. Zhang, Y. Gu, Adaptive mobile interfaces through grammar induction, Proc. Int. Conf. Tools with Artif. Intell. ICTAI, 2008: pp.133–140.

BugHint: A Visual Debugger Based on Graph Mining

Jennifer L. Leopold
Missouri University of Science and Technology
Department of Computer Science
Rolla, MO, USA
leopoldj@mst.edu

Nathan W. Eloe
Northwest Missouri State University
School of Computer Science and Information Systems
Maryville, MO, USA
nathane@nwmissouri.edu

Patrick Taylor
Missouri University of Science and Technology
Department of Computer Science
Rolla, MO, USA
taylor@mst.edu

Abstract – *Why doesn't my code work? Instructors for introductory programming courses frequently are asked that question. Often students understand the problem they are trying to solve well enough to specify a variety of input and output scenarios. However, they lack the ability to identify where the bug is occurring in their code. Mastering the use of a full-feature debugger can be difficult at this stage in their computer science education. But simply providing a hint as to where the problem lies may be sufficient to guide the student to add print statements or do a hand-trace focusing on a certain section of the code. Herein we present a software tool which, given a C++ program, some sample inputs, and respective expected outputs, uses graph mining to identify which lines in the program are most likely the source of a bug. The tool includes a visual display of the control flow graph for each test case, allowing the user to step through the statements executed. Experimental results from a group of CS1 students show that practice with this method: (1) makes students faster at finding bugs, (2) improves the way students test a program, and (3) improves program comments by students.*

Keywords – debugging; graph; data mining; visualization

I. INTRODUCTION

As discussed in [1], instructors and teaching assistants for introductory programming courses frequently are asked by their students: why doesn't my program work? Often the students understand the problem they are trying to solve well enough to articulate a variety of input and output scenarios. For example, if they are trying to find the sum of all even values in a list of numbers, they know that the input list {1, 2, 3, 4, 5} should produce a result of 6, and the input list {1, 3, 5, 7} should produce a result of 0. However, they frequently lack the ability to identify, or even narrow down, where a bug is occurring in their code when it does not produce the correct results. The recommendation to add print statements, although easy for experienced programmers, can require some skill and practice to master, and the use of a full-feature debugger can be cumbersome and intimidating to a novice programmer.

Herein we present BugHint, a software tool which, given a C++ program, some sample inputs, and respective outputs, uses graph mining to identify which lines in the C++ program are most likely causing the erroneous results. The tool includes a visual display of the control flow graph for each test case (i.e., sample input), allowing the user to step through the statements as they are executed. The goal is that the student will take the bug hint and subsequently scrutinize the logic in the identified section of the program, thereby finishing t he d ebugging process on his/her own. Experimental results with a group of CS1 students have shown that practice with this tool not only makes students find bugs faster after training, but also improves the way students test their programs and comment their programs. The organization of this paper is as follows. Section II provides a brief overview of related work in debugging experiences with beginning programmers and the use of visualization in debugging. Section III discusses the foundation for and implementation of our software tool, including the graph mining analysis and the graphic user interface. The experimental design and results are presented in Section IV. Future work is discussed in Section V. A summary and conclusions are given in Section VI.

II. RELATED WORK

A. *Debugging Experiences with Beginning Programmers*

Several studies (e.g., [1], [2], and [3]) have identified problems that students experience with coding in introductory computer science courses, resulting in a proliferation of program bugs. Debugging strategies such as strategically placed print statements can be difficult to teach [1]. There are full-feature debugging tools such as GDB, which allow one to set breakpoints in the code and/or watch the values of variables change during execution of the program. However, for some novice programmers these tools can be too cumbersome and/or intimidating to use.

DOI reference number: 10.18293/DMSVIVA2018-006

After years of study, there is no consensus as to whether beginning programmers should be exposed to a full-feature debugger.

There have been studies that have successfully integrated the teaching of programming and a debugger at the introductory level. In [2] the authors used a debugger to demonstrate construction of Java objects and function calls in addition to using the debugger to find bugs in programs. Similarly, the authors of [4] used debugging exercises and simple debugger functions to reinforce programming concepts (e.g., loops) that they were teaching.

However, full-feature debugger tools are not without criticism. In addition to the complaint that they may further confound the debugging experience for novice programmers who are already dealing with learning about an editor, operating system commands, compiler error messages, and programming language syntax, there is the issue that debuggers can potentially introduce additional bugs. A heisenbug is a software bug that is introduced when one attempts to study or analyze a program. Running a program in a debugger can actually modify the original code, changing memory addresses of variables and the timing of the execution. Debuggers often provide watches or other user interfaces that cause additional code to be executed, which, in turn, modify the state of the program. Time also can be a factor in heisenbugs, because race conditions may not occur when the program is slowed down by single-stepping through lines of code with the debugger.

Herein we do not seek to answer the question of whether the use of a full-feature debugger should be integrated into an introductory programming course. Rather, it is our intention to present a simple tool which the student can use as a debugging aid and training tool. Our aim is similar to the function of the instructor or teaching assistant who provides a hint as to where in the student's code the bug might be occurring. It is still up to the student to add print statements, do a hand-trace focusing on those particular statements, or use other techniques to try to fix the problem on his/her own, considering various input-output test cases.

B. *Visualization in Debugging*

Many contemporary debugging tools provide some type of visual representation of the source code in addition to displaying the program as text. This visual representation could be in the form of a flow chart (e.g., Visustin [5]), a control flow graph (e.g. KDevelop [6] and Dr. Garbage [7]), or UML diagrams (e.g., Eclipse ObjectAid [8]). The objective of the visualization is to facilitate understanding of some properties of the program such as the logic and/or the interactions between code blocks. To this end, animation (not just a static representation) of program execution has long been found to be useful.

Just as UML diagrams were deemed to be particularly helpful for object-oriented programming languages like Java and C++, control flow graphs have been found to be useful in debuggers for various programming paradigms. The authors of [9] presented GRASP, a graphical environment for analyzing Prolog (i.e., logic) programs; the tool dynamically animates the executed sequence of Prolog sub-goals as a control flow graph and allows the user to inspect instantiation of variables as s/he steps through the execution. In [10] the authors introduced a debugging tool for MPI (i.e., parallel) programs that displays a message-passing graph of the execution of an MPI application; parts of the graph are hidden or highlighted based on the sequence of MPI calls that occur during a particular execution. Mochi [11] was created as a visual debugging tool for Hadoop (i.e., distributed programs); it displays the control flow of the workloads of each processor as a graph, allowing the user to observe the map and shuffle processing that takes place, and possibly identify erroneous sequencing and/or data partitioning.

III. IMPLEMENTATION

A. *Discriminative Graph Mining*

Our tool, BugHint, was motivated by the work presented in [12] for identifying bug signatures using discriminative graph mining. The basic idea is to first produce a control flow graph for a program written in a procedural programming language (in our case, this is C++). In brief, a control flow graph is a directed graph made up of nodes representing basic blocks. Each basic block contains one or more statements from the program. There is an edge from basic block B_i to basic block B_j if program execution can flow from B_i to B_j. For more information on control flow graphs and determination of basic blocks, see [13]. For C and C++ programs, a control flow diagram can be generated by compiling the program with clang and opt (we specify no optimization), and then creating the graph as a dot graph description language file using dot.

As an example, consider the C++ program shown in Fig. 1 which is supposed to replace only the first occurrence of either x or y in an array a with the value of z. This program does not perform that task correctly; it contains a bug. For simplicity, the code to output the final values of the array is commented out in this program since it is not where the bug occurs.

```
int main(){                                  //line 1
  // inputs to the program
  int x = 1;
  int y = 7;
  int z = 0;
  int a[2] = {1, 2};
  int arraySize = 2;
  for(int i = 0; i < arraySize; i++){ //line 2
    if(a[i] == x){                           //line 3
      a[i] = z;                              //line 4
    }                                        //line 5
    if(a[i] == ){                            //line 6
      a[i] = z;                              //line 7
    }                                        //line 8
  }                                          //line 9
  // code to output a[ ]...
  return 0;                                  //line 10
}
```

Figure 1: Example C++ program: replacement

An example of a control flow graph for this program is shown in Fig. 2. In this graph there are eight blocks; the figure shows which lines of code are contained in each block.

After constructing a control flow graph for the program to be analyzed, our tool needs to consider test cases. These need to be specified in terms of sample input and expected output. The test cases should be as representative as possible of all boundary conditions for the program. However, a novice programmer may be unfamiliar with that notion. At the very least, the user must specify at least one input sample that is known to produce correct output and at least one input sample that is known to produce incorrect output; the user must distinguish these as 'correct' and 'incorrect.' In Table 1 we list some sample test cases for the example program shown in Fig. 1.

For each sample case, our tool produces a code trace in terms of the lines executed for the specified input. The code traces for the four sample cases shown in Table 1 are listed in Table 2. It should be noted that if there is an infinite loop (which is a common bug) during execution of one of the sample input cases, the output from the code trace should be sufficient to identify the line(s) where the bug is occurring and no further analysis should be necessary. From each code trace, we also generate a control flow graph for that sequence. The control flow graphs for code traces 1 and 2 from Table 2 are shown in Fig. 3; the control flow graphs for code traces 3 and 4 are the same as the graph shown in Fig. 2.

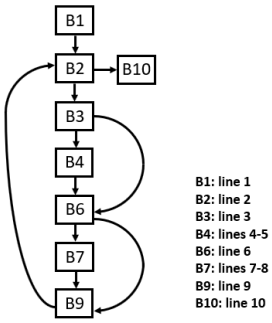


Figure 2: Control flow graph for the example program

Table 1: Sample Test Cases for the Example Program

Sample Case Num	a	x	y	z	Result	
1	{1, 2}	1	7	0	{0, 2}	correct
2	{1, 2}	7	1	0	{0, 2}	correct
3	{1, 7}	1	7	0	{0, 0}	incorrect
4	{1, 7}	7	1	0	{0, 0}	incorrect

Table 2: Code Traces for the Example Program

Sample Case Num	Trace Line Numbers	Trace Block Numbers
1	1 2 3 4 5 6 9 2 3 6 9 2 10	B1 B2 B3 B4 B6 B9 B2 B3 B6 B9 B2 B10
2	1 2 3 6 7 8 9 2 3 6 9 2 10	B1 B2 B3 B6 B7 B9 B2 B3 B6 B9 B2 B10
3	1 2 3 4 5 6 9 2 3 6 7 8 9 2 10	B1 B2 B3 B4 B6 B9 B2 B3 B6 B7 B9 B2 B10
4	1 2 3 6 7 8 9 2 3 4 5 6 9 2 10	B1 B2 B3 B6 B7 B9 B2 B3 B4 B6 B9 B2 B10

The collection of graphs for the sample cases are next analyzed to identify non-discriminative edges. A non-discriminative edge is an edge that appears in every graph that is in the collection of execution graphs. Such edges are removed from each graph in the collection since they are the same in each execution, and, as such, are not informative in distinguishing where the bug occurs. The collection of control flow graphs with non-discriminative edges removed for our running example is shown in Fig. 4.

Finally, the collection of graphs is analyzed to determine what subgraph is common to the faulty (i.e., incorrect output) execution graphs, but not common to the correct execution graphs. This corresponds to the section of code where the bug likely occurs. For our running example, such a discriminative control flow graph is shown in Fig. 5. It tells us that the bug involves blocks B4, B6, and B7, which correspond to lines 4-8 in the program. The hope is that the student will use this information to realize that, after changing the value to z in line 4, the program should not proceed to lines 6-8 since the specifications of the problem were to change either, not both, the occurrence of x or y to z.

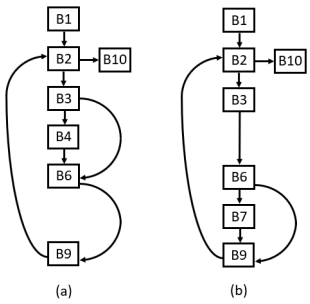


Figure 3: Control flow graphs for (a) trace 1 and (b) trace 2 from Table 2

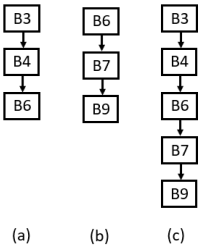


Figure 4: Control flow graphs with non-discriminative edges removed for (a) trace 1, (b) trace 2, and (c) traces 3 and 4 from Table 2

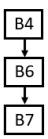


Figure 5: Discriminative control flow graph for the example program

The discriminative graph, and hence the bug in the program, may not consist of lines that are executed in the incorrect cases, but not executed in the correct cases (as was the situation in this example program); it could be the reverse situation. Or it could be the case that we cannot find a subgraph that is common to all faulty (or correct) execution graphs, but not common to the correct (or faulty) execution graphs. The algorithms we utilize for identifying the "best" discriminative graph are explained next. These differ slightly from those proposed in [12] and [14] for discriminative graph mining. Let C^+ and C^- represent the sets of control flow graphs for the sample test cases producing correct and incorrect results, respectively; there must be at least one graph in each such set. The function FindDiscriminativeGraph (Alg. 1) first removes non-discriminative edges from the graphs in both sets. It then calls CreateDiscriminativeGraph (Alg. 2) to try to find a subgraph that is common to all faulty execution graphs, but not common to all the correct execution graphs. If we are unable to find such a graph, then the function RelaxedCreateDiscriminativeGraph (Alg. 3) is called, which relaxes the requirement that the subgraph we seek not be present in all of the correct execution graphs; instead the subgraph only has to not be present in $\alpha * |C^+|$ of the correct execution graphs, where α is a user-specified parameter (our default is $\alpha = 0.5$).

The algorithms FindDiscriminativeGraph and CreateDiscriminativeGraph use a function called Augment; this function takes the subgraph G and adds to it an edge (and possibly a node) such that the source vertex exists in G, and the edge (and destination node) exists in all graphs in S1. In this way, a subgraph with an additional edge that exists in all elements of S1 is created and considered by the algorithm.

If we still fail to find a discriminative subgraph, then the bug likely does not involve code that is executed in all faulty cases and not in correct cases, but rather involves code that is executed in correct cases and not in faulty cases. Thus, we again call CreateDiscriminativeGraph, but reverse the order of the parameters (C^+ and C^-) from our previous call. If we still fail to find a discriminative subgraph, we again call RelaxedCreateDiscriminativeGraph and look for a subgraph that only has to not be present in $\beta * |C^+|$ of the correct execution graphs, where β is a user-specified parameter (our default is $\beta = 0.5$). It is possible that the resulting discriminative graph will be disconnected. We output the smallest connected component in that graph using the assumption that a novice programmer will want to focus on a single, sequential section of his/her program for scrutinizing the bug, rather than examining multiple, "fragmented" sections of code.

Algorithm 1 FindDiscriminativeGraph(C^+, C^-, α, β)

Require: C^+: set of control flow graphs for inputs producing correct output
Require: C^-: set of control flow graphs for inputs producing incorrect output
Require: α: percentage of graphs that discriminative subgraph need not be present in C^+ when relaxing conditions
Require: β: percentage of graphs that discriminative subgraph need not be present in C- when relaxing conditions
remove non-discriminative edges from graphs in C^+ and C^-
G = CreateDiscriminativeGraph(C^-, C^+)
if G is empty **then**
  $G = RelaxedCreateDiscriminativeGraph(C^-, C^+, |C^+| * \alpha)$
  if G is empty **then**
    $G = CreateDiscriminativeGraph(C^+, C^-)$
    if G is empty **then**
      $G = RelaxedCreateDiscriminativeGraph(C^+, C^-, |C^-| * \beta)$
    end if
  end if
end if
G' = smallest connected component in G
output G'

Algorithm 2 CreateDiscriminativeGraph(S1, S2)

Require: S1: set of control flow graphs
Require: S2: set of control flow graphs
FreqSG = queue of 1-edge subgraphs in S1
while FreqSG is not empty **do**
  G = FreqSG.dequeue()
  if G is not in any graph in S2 **then**
    return (G)
  end if
  NewGraphs = Augment(G)
  for each graph G' in NewGraphs **do**
    FreqSG.enqueue(G')
  end for
end while
return (empty graph)

Algorithm 3 RelaxedCreateDiscriminativeGraph(S1, S2, γ)

Require: S1: set of control flow graphs
Require: S2: set of control flow graphs
Require: γ: threshold for number of graphs discriminative subgraph must be present in
FreqSG = queue of 1-edge subgraphs in S1
while FregSG is not empty **do**
  G = FreqSG.dequeue()
  if G is in $< \gamma$ graphs in S2 **then**
    return (G)
  end if
  NewGraphs = Augment(G)
  for each graph G' in NewGraphs **do**
    FreqSG.enqueue(G')
  end for
end while
return (empty graph)

It should be noted that it is possible that our algorithm will not find any graph that meets the discriminative conditions; this is largely dependent upon the specified test

cases. If the resulting discriminative graph is empty, the user will be told that no hint can be provided and that specification o f a dditional t est c ases (that p roduce both correct and incorrect output) might help.

B. *Graphic User Interface*

Some of the tools used to generate the information needed for the GUI are not easily installable on all platforms (specifically, clang/LLVM). To make BugHunt available to a broad range of novice programmers, a primary concern was making the tool platform-independent. The GUI for BugHunt is a web application written in Node.JS, using vis.js for the visualization of the control flow g raphs. The web application is integrated with various utilities written in Python, and makes the necessary system calls to add code to display basic block information, compile, and run the various traces, as well as to analyze the correct and incorrect execution traces to provide the bug hint. Hence the tool can be run from any type of computer but can be deployed using emerging technologies such as Docker or the Linux Subsystem for Windows.

Fig. 6 shows a screenshot of the BugHunt GUI with the example program from Fig. 1 and the test cases from Table 1. The arrow buttons in the GUI allow the user to step forwards and backwards through a selected execution case; both the corresponding nodes and (text) lines in the program subsequently will be highlighted. Any particular block in an execution sequence (listed below the graph display) also can be selected (i.e., clicked-on) with the mouse.

The sources for the web-based GUI are available at the following links:
`https://bitbucket.org/neloe/cfg_tracer/src/master/`
`https://bitbucket.org/neloe/cfg_debug_backend/src/master/`

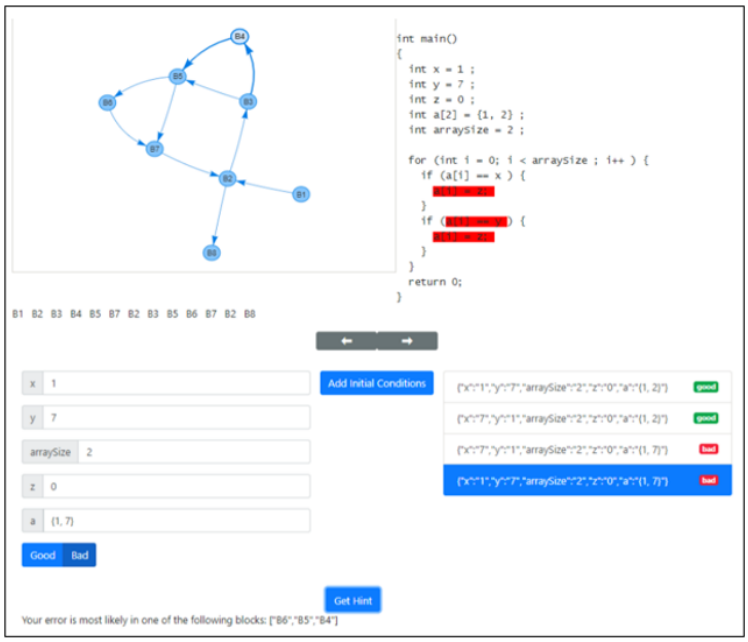


Figure 6: BugHunt GUI, showing the possible erroneous lines as determined by control flow graph trace analysis.

IV. EXPERIMENTS

To test the hypothesis that experience with BugHint improves debugging skill, we tested training with the software using students who had just recently completed CS1. Notably, we did not merely test whether BugHint made it faster to find the bug in a program where a hint was given, but instead whether experience with BugHint would actually improve debugging skill such that students would be better at debugging new programs without BugHint.

A. *Experimental Design*

We employed a between-subjects design, as this is usually less confounded by the design itself, and stronger, when the number of subjects is sufficiently large to accommodate two groups. Two groups of subjects experienced different pre-training, with identical post-training testing. The human subjects were tested in four sections of a CS2 laboratory class during the second week of the semester (immediately following their recent completion of CS1); two sections were the treatment condition, and two were the control. A total of 163 students participated in this study. All experiments were performed in accord with human subjects and institutional review board guidelines. The two groups of subjects are categorized as follows (Fig. 7):

1. Treatment pre-training - This group was introduced to BugHint and instructed how to use it. They completed a pre-training exercise using BugHint on

three small programs that contained bugs.

2. Control pre-training - This group was given instructions and tips on manual debugging. They completed a pre-training exercise using standard manual debugging methods (e.g., printing out values in for-loops and binary search through code with print statements) on the same three small programs that contained bugs that were given as a practice exercise to the treatment group.

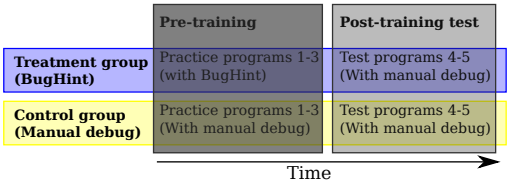


Figure 7: Experimental design. Performance on the Post-training test is the key indicator of training success. Post-training test is identical between conditions.

These two groups of subjects completed identical post-training debugging testing activities on three more small programs that contained bugs; neither group received BugHint help. In addition to being required to fix and document a bug, subjects were given the following questions for subjective evaluation:

- What time did you start this problem? __ :__ AM / PM (circle one)
- What time did you fix the bug? __ :__ AM / PM (circle one)
- How confident are you that your program works correctly now? 1-5 Likert
- How difficult was the problem? 1-5 Likert
- How much would comments in the code have helped you debug it? 1-5 Likert
- How well-written was the code? 1-5 Likert
- What would make informative test cases to test the buggie program and why? Free response
- What would make have been helpful comments to have added to the buggie program to have helped find the bug? Free response

In each training or test case, students encountered a simple program to debug. One example was to flatten and reverse a 2D array (Fig. 8). Other examples included a simple sort, replacement of values, increment and sum computation, smallest subsequence, and base conversion.

The example C++ program given below is supposed to "flatten" and reverse the order of the elements of a given 2D array. By "flatten" we mean a n x m 2D array should become a vector of n * m elements. For example, $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ should become $\{6, 5, 4, 3, 2, 1\}$.

```
int main(){                                    //line 1
  int rowSize = 3;
  int colSize = 2;
  int a[3][2] = {{1, 2}, {3, 4}, {5, 6}};
  int b[6] = {0, 0, 0, 0, 0, 0};
  int k;
  int m;
  int m_last = -1;
  for(int i = 0; i < rowSize; i++){            //line 2
     for(int j = 0; j < colSize; j++){         //line 3
       m = i * rowSize + j;                    //line 4
       if((m >= rowSize * colSize) ||
           (m <= m_last)){                     //line 5
         m = 1;                                //line 6
       }                                       //line 7
       k = rowSize * colSize - m - 1;          //line 8
       b[k] = a[i][j];                         //line 9
       m_last = m;                             //line 10
     }                                         //line 11
  }                                            //line 12
  return 0;                                    //line 13
}
```

Sample Case Num	a	Result	
1	{{1, 2}, {3, 4}}	{4, 3, 2, 1}	correct
2	{{1, 2}, {3, 4}, {5, 6}}	{0, 4, 3, 0, 6, 1}	incorrect
3	{{1, 2, 3}, {4, 5, 6}}	{0, 6, 5, 3, 4, 1}	incorrect

Hint: The bug may be **around** lines 6 or 7.

Figure 8: Example case that participants encountered. Solution/Bug for the above program: Line 4 should use colSize instead of rowSize.

Depending on the condition, participants may have been queried about finding correct and incorrect test cases themselves, then been given the correct and incorrect test cases, and hints about line numbers revealed progressively. In the control condition, participants were not presented with discussion of correct and incorrect test cases nor line number hints. Participants were expected to document the line number of the bug, and its fix.

Importantly, during data entry, all answers were scored and entered by a 'blind' grader who did not know the full study design, intended results, or purpose of the study. All data processing was entirely automated using the same procedures for each measure (barring different statistical tests for binomial versus numerical data).

B. *Experimental Results*

We analyzed the proportion of students who found each bug in post-testing, comparing the group with BugHint pre-training (the treatment group) to the group with normal debugging pre-training (the control group). As shown in Fig. 9, the left panel represents the results for one test program, and the right panel represents the results for another test program. Z-scores for a between-group test for binomial data indicate the treatment group found more bugs during post-testing; success or failure finding the bug was encoded as a Boolean, and thus a binomial test was required. Further, when turned into proportions, t-tests were also significant, though are not the most appropriate test. The t-statistic is the difference of means between compared groups, divided by the standard error of the mean, $(u_1 - u_2)/SEM$. Generally with reasonable number of samples or participants, it is necessarily the case that when

SEM error bars are not overlapping, the t-test would be significant, with α about $p < 0.03$ on a 1-tailed test in the expected direction. Thus, SEM bars are a good indicator of "significance."

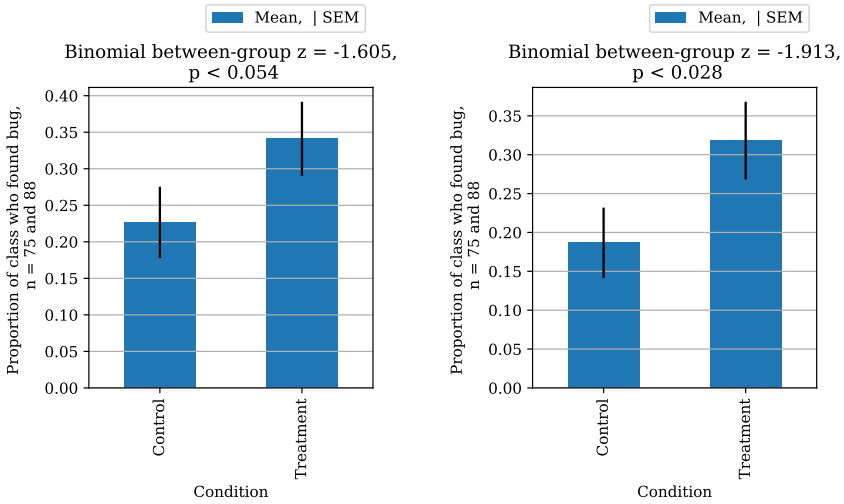


Figure 9: Proportion of students who found each bug for two post-test programs. Performance on first post-test on the left, and second post-test on the right.

Additionally, we examined how much time the students spent before they thought they had found a bug. Students in the treatment group reported finishing searching for the bugs with non-significantly less time than those in the control group. These data were not filtered by students actually having correctly found the bug, and so data merely represent the duration of time until they found what they thought was the solution. The comparisons between the treatment and control groups for two test programs are shown in Fig. 10.

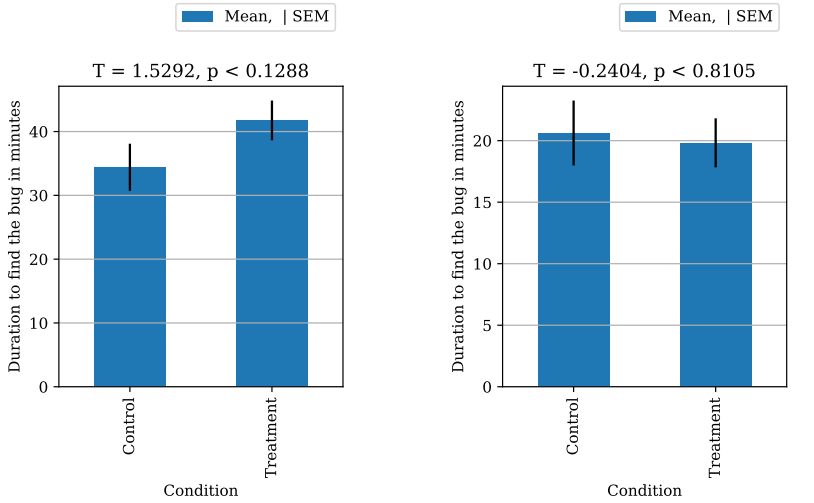


Figure 10: Duration of time spent before students thought they had found the bugs for two test programs

The students were given a few sample test cases (in terms of input and output) for each program that they were asked to debug. They were then asked to come up with additional informative test cases, and their reasoning for those test cases. These responses were blindly graded and given a point value from 1-5, with 5 being correct and good reasoning. Students who completed the BugHint pre-training demonstrated superior ability to come up with and reason about test cases for debugging on later post-test. As shown in Fig. 11, t-tests and SEM bars indicate "significance", and support the conclusion that the BugHint group demonstrated better post-test performance. D'Agostino, and Pearson's test of normality suggest that these two measures did not significantly deviate from

normal, $k = 4$, $p = 0.13$; $k = 0.22$, $p = 0.89$, where $p < 0.05$ indicates non-normality. Parametric t-tests assume normality, but with sufficiently large sample size are robust to non-normality [11]. Distributions are plotted in Fig 13.

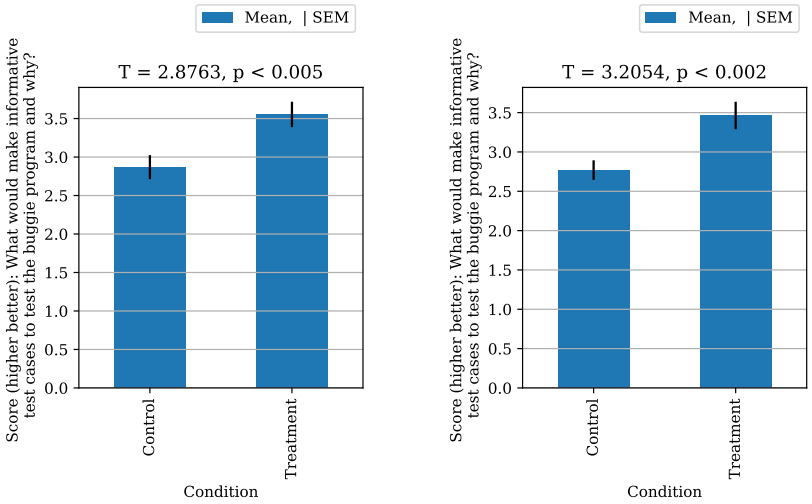


Figure 11: How well students came up with informative test cases for two test programs

In addition to assessing the importance of test cases, we sought to evaluate the impact of commenting in relation to debugging. The students were asked what would have been helpful comments to have included in each program they debugged. These comments were manually graded and evaluated by the blind grader, and given a score from 1-5, with 5 being best. Again, t-tests and SEM bars tend toward "significance." As shown in Fig. 12, we found that the BugHint group appeared to demonstrate better post-test performance on two separate programs.

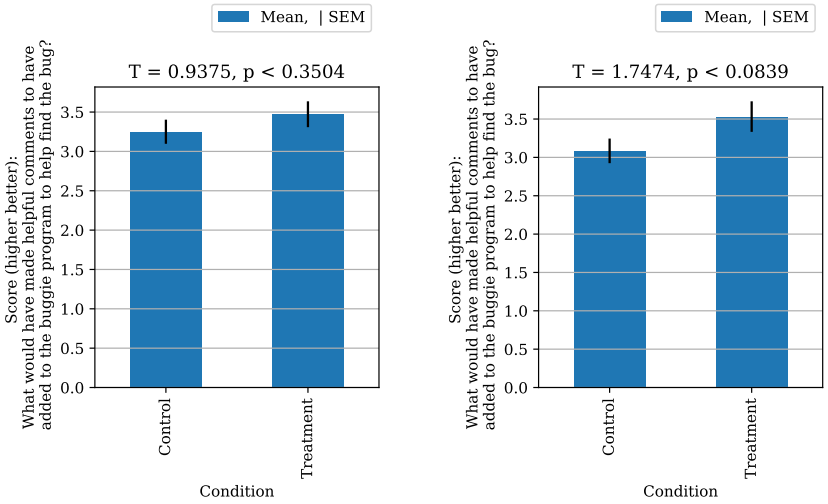


Figure 12: Helpfulness of comments that students added for two test programs

To explore the data distributions for each of the above non-continuous measures, and elaborate further on the pattern of performance we generated violin plots. First, the plots in Fig. 13 recapitulate the patterns in Fig. 9. Second, these suggest that the distribution of scores (over 1-7) were reasonably amenable to evaluation via t-test, and via observing the SEM.

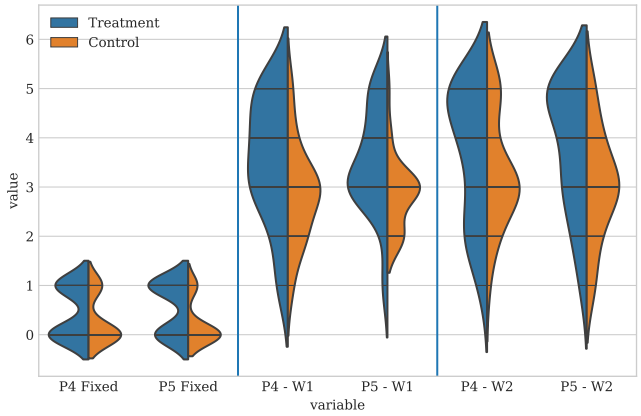


Figure 13: Distributions of scores plotted in previous box-plots, grouped by condition, for performance on two post-test programs (P4 and P5 above). Higher on the Y-axis is better. Left two plots show percent of bugs found (0-1). Middle two plots show grader-evaluated performance on student-generated test cases (1-7). Right two plots show grader-evaluated performance on student-generated comments (1-7).

Finally, we asked two subjective questions regarding the students' overall experience. The first question was whether they felt that debugging was easier with BugHint than without. We found that a majority of students reported that finding bugs was easier with the BugHint methods than the manual debugging post-test. This is to be trivially expected since hints (if correct and specific enough) should make finding bugs easier. The results are shown in Fig. 14 (left panel). Single-group binomial data do not have SEM, and a single-group binomial test-statistic with expected proportions of 50/50 was performed (as shown on the plot).

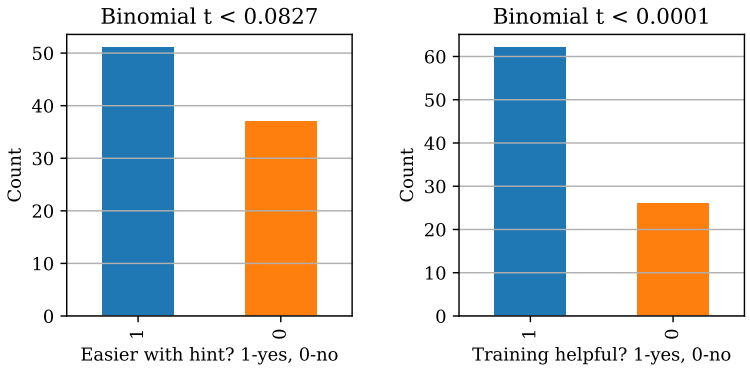


Figure 14: Left: Was debugging subjectively easier with BugHint? Right: Did training with BugHint help you learn to debug even without it?

The second question was whether the students thought that training with BugHint would help them learn to debug without BugHint while manually debugging. Students in the treatment pre-training condition felt that they benefited from the pre-training, specifically in regard to their ability to debug later without its help. Single-group binomial data do not have SEM, and a single-group binomial test-statistic with expected proportions of 50/50 was performed as shown on the plot in Fig. 14 (right panel).

V. FUTURE WORK

The current implementation of BugHint is restricted to programs that do not contain user-defined functions. Our algorithm for finding a discriminative subgraph, which is based on analysis of the control flow graphs for correct and incorrect test cases, will be easily extensible to additional block structures. The larger challenge will be accommodating this additional visual complexity in the graphical user interface. We intend to perform usefulness and usability studies with novice programmers to find ways of implementing visual representation and navigation of the different modules of the control flow graph in a manner that they (the students) can best understand.

We also intend to consider using other existing algorithms for finding discriminative subgraphs (e.g., [15] and [16]) and/or adding other options (in addition to our current α and β parameters) to our algorithm in order to

find t he b est d iscriminative s ubgraph, a nd h ence provide the best suggestion for the bug hint. For example, we may prioritize subgraphs containing statements with multiple operators and/or particular operators (e.g., && and ||), which are the source of common (logic) mistakes for novice programmers.

VI. SUMMARY AND CONCLUSIONS

Herein we have presented a simple debugging tool, which, given a C++ program that has a logic error just serious enough to occasionally produce erroneous output while sometimes producing correct output, and some sample inputs with corresponding outputs, uses discriminative graph mining to identify which lines in the program are most likely the source of the bug. The tool includes a visual display of the control flow graph for each test case, allowing the user to step through the statements executed. We have made the source code for this tool available on a publicly accessible website. Students who completed pre-training using BugHint did better on post-testing than students who completed pre-training without BugHint, even though all groups had no help for post-training. During a post-training exercise where both groups completed the exact same activity of debugging three more practice programs, the treatment group found more of the bugs, self-generated more informative test cases and reasoning regarding those test cases, and self-generated more helpful comments to add to the code itself. In general, it appears that the extra-formalized method of using BugHint may improve the way students think about the debugging practice. We speculate that this may be due to one requirement of the BugHint method, which is focused on the specification of some test cases that generate correct output and some test cases which produce incorrect output. This exercise itself could be performed to further test that hypothesis.

REFERENCES

[1] C. Lewis and C. Gregg, "How Do You Teach Debugging?: Resources and Strategies for Better Student Debugging", Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, TN, Mar. 2-5, 2016, p. 706.

[2] R.C. Bryce, A. Cooley, A. Hansen, and N. Hayrapetyan, "A One Year Empirical Study of Student Programming Bugs", Frontiers in Education Conference, Washington, DC, Oct. 27-30, 2010, pp. 1-7.

[3] J.H.I.I. Cross, T.D. Hendrix, and L.A. Barowski, "Using the Debugger as an Integral Part of Teaching CS1", "Frontiers in Education, Boston, MA, Nov. 6-9, 2002. pp. 1-6.

[4] G.C. Lee and J.C. Wu, "Debug It: A Debugging Practicing System", Computers & Education, 32, 1999, pp. 165-179.

[5] Visustin, `http://www.aivosto.com/visustin.html`

[6] KDevelop, `https://liveblue.wordpress.com/2009/07/21/3-visualize-your-code-in-kdevelop/`

[7] Dr. Garbage, `https://sourceforge.net/projects/drgarbagetools/files/`

[8] Eclipse ObjectAid, `http://www.objectaid.com/sequence-diagram`

[9] H. Shinomi, "Graphical Representation and Execution Animation for Prolog Programs", International Workshop on Industrial Applications of Machine Intelligence and Vision (MIV-89), Tokyo, Apr. 10-12, 1989, pp. 181-186.

[10] B. Schaeli, A. Al-Shabibi, and R.D. Hersch, "Visual Debugging of MPI Applications", in Recent Advances in Parallel Virtual Machine and Message Passing Interface, A. Lastovetsky, T. Kechadi, J. Dongarra (eds)., EuroPVM/MPI, Lecture Notes in Computer Science, vol. 5205, Springer, Berlin, Heidelberg, 2008, pp. 239-247.

[11] J. Tan, X. Pan, S. Kavulya, R. Gandhi, and P. Narasimhan, "Mochi: Visual Log-Analysis Based Tools for Debugging Hadoop", CMU-PDL-09-103, Parallel Data Laboratory, Carnegie Mellon University, Pittsburg, PA, May 2009.

[12] H. Cheng, D. Lo, Y. Zhou, X. Wang, and X. Yan, "Identifying Bug Signatures Using Discriminative Graph Mining", ISSTA, Chicago, IL, Jul. 19-23, 2009, pp. 141-151.

[13] A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques, and Tools, Addison Wesley, 2nd edition, 2006.

[14] X. Yan, H. Cheng, J. Han, and P.S. Yu, "Mining Significant Graph Patterns by Leap Search", SIGMOD 2008, Jun. 9-12, 2008, Vancouver, BC, Canada, pp. 433-444.

[15] N. Jin and W. Wei, "LTS: Discriminative Subgraph Mining by Learning from Search History", IEEE 27th International Conference on Data Engineering (ICDE), 2011, pp. 207-218.

[16] M.G.A. El-Wahab, A.E. Aboutabl, and W.M.H. El Behaidy, "Graph Mining for Software Fault Localization: An Edge Ranking Based Approach", Journal of Communications Software and Systems, Vol. 13. No. 4, Dec. 2017, pp. 178-188.

[17] de Winter, J.C.F. and D. Dodou , Five-Point Likert Items: t test versus Mann-Whitney-Wilcoxon, Practical Assessment, Research and Evaluation, 2010. 15(11).

A Logic Range-free Algorithm for Localization in Wireless Sensor Networks

Balzano Walter
Dipa. Ing. Elettrica e Tecnologie dell'Informazione
Univeristà di Napoli, Federico II
Napoli, Italy
Email: walter.balzano@gmail.com

Stranieri Silvia
Università di Napoli, Federico II
Napoli, Italy
Email: silviastranieri1047@gmail.com

***Abstract*—Nowadays, localization is a very important research problem in the context of Wireless Sensor Networks (WSNs). These networks are made by nodes connected to each other wireless and able to collect data: all the retrieved information can be then held by a central unit, a more performance machine, which can in turn collect and elaborate all the data. The main contribution of this work is in an innovative way to recognize the position of a point in a certain closed environment, by exploiting the RSSI (Received Signal Strength Indicator) and a logic approach: for this purpose, Prolog has been used in order to describe an intuitive non-greedy algorithm and an appropriate simulation program able to make an estimation of a point localization, providing the global optimum. To this aim, the placement of hub in the interesting area is crucial and, in particular, when this area presents some obstacles which can alter the transmission signal. The expressive power of Prolog and the way the its logic engine works made this programming language suitable for our purpose: in fact, the backtracking strategy opportunely reflects the way the relative positioning of hubs is performed with the aim of improving the cover of a specific indoor area at each step, which is a very important application of the localization problem.**

Key-words: Wireless Sensor Networks, localization, coverage, Prolog

I. INTRODUCTION

Wireless Sensor Networks constitute a spread field of research of the last years, due to the potential of that system which is made by nodes self-organized, broadcasting information and data all over the net. Many proposals have been made to improve the communication between the nodes, to select the best routing protocol possible, and to locate nodes inside the net. The localization is a crucial point in this issue, since a fast and efficient positioning technique can also provide a way to optimally cover an area, in such a way that a minimum signal is guaranteed to any point.

The hub distribution in an outdoor environment does not create any trouble, due to the absence of any kind of obstacle. In a closed area, instead, the presence of walls, doors, or any other impediment can alter the power of the transmission signal, and this is something we want to deal with in order to guarantee an optimal distribution of hub in indoor environment too.

The idea is to design and analyze an algorithm which preventively creates a mapping of hub inside the interesting area, then applies some localization techniques in order to estimate the position of a point in that area. The approach is the logic one, since the way the algorithm works is based on backtracking: whenever a hub is added into the environment, its total mapping is questioned and the position of each hub is computed again in order to optimize some metric on the signal (such as average or variance in the whole space). This approach provides a non-greedy algorithm whose solution is guaranteed to be the global optimum, rather than the local one. The algorithm presents two possible ways to start the mapping: (*i*) by using some fixed hub, whose position is known and uneditable for some logistic reasons; (*ii*) without any a priori fixed hub.

In this work, we consider three different kinds of localization techniques: range based, angle based, and range free, focusing our approach on the last one. Moreover, before performing the actual localization, we use a simulation program to generate the map of an environment with some obstacles, taking into account three possible nodes distribution: casual, geometric, and signal based.

The work is organized as follows: in section II, we give an overview of the actual state of the art, analyzing some works and proposes about localization in Wireless Sensor Networks, and strategies that use RSS measurements to improve some known localization techniques; in section III, we briefly introduce the most significant localization approaches, then we focus on our propose, providing a simulation program first, and a research algorithm next in order to estimate the position of any point on a network, with a logic approach; in section IV, we provide an example of simulation and we apply some analysis on the proposed technique, highlighting how the precision in the estimation can vary according to some values; finally, in section V, there are conclusions and future hints for the growth of this important research field.

II. RELATED WORK

Wireless Sensor Networks are, nowadays, one of the most studied research topics. The development of such networks was initially born for military purposes, while now, as explained in [2], there is a bunch of applications of these nets: environment and structures monitoring, traffic management, surveillance, and many others application fields. Actually, this is the reason why many studies are made about this topic and all the related issues, such as localization of nodes in such a network and signal distribution. An important indicator which is largely used in Wireless Sensor Networks for localization purposes is the

DOI reference number: 10.18293/DMSVIVA2018-008

RSSI (Received Signal Strength Indicator). This indicator provides useful information about the signal power for any retrieved hub in the environment. For instance, in [6] RSSI is exploited in traffic control field in order to estimate the positioning of vehicles. They state that Global Positioning System does not always guarantee the accuracy needed in cooperative-vehicle-collision-warning systems, while the radio-based-ranging approach founded on RSSI improves the accuracy. Using the same approach, in [7] they propose a range-free algorithm based on RSSI comparisons, called Ring Overlapping. Each node uses overlapping rings in order to guess the possible area in which it lie: given an anchor node *A*, each ring is actually generated by comparing the RSSs received by a node from *A* and the ones received by other anchor nodes from *A*. Even in [20], they highlight the importance of positioning accuracy in vehicle-to-vehicle field.

A crucial variation point in localization algorithms in WSN is in the choice of using anchor nodes or not. In [3] is proposed an anchor-based localization approach: the main idea is that each anchor is aware of its position, because equipped with GPS, and it periodically shares its current location with the other nodes which are able, thanks to this information, to locate themselves. This approach tolerates the presence of obstacles and has the benefit of not requiring any hardware modification. Oppositely, in [12] they prefer an anchor-free approach, summarizing all the drawbacks of having fixed nodes in a network.

In our previous work, we focus on logic strategies in order to deal with many problems related to traffic control, such as in [15], sometimes integrating it with clustering techniques ([14], [16]), or Distance geometry problem, like in [17]: even in this work we use the logic approach (*i*) to facilitate the comprehension of the algorithm behavior, through elegant and compact code, and (*ii*) to exploit the expressiveness power of Prolog and its cut operator to prune useless computational paths. But, many other localization techniques are proposed in literature. In particular, in [4] they highlight three categories of localization approaches: (*i*) AOA (Angle of Arrival) represents the angle between the propagation direction and some reference direction (orientation) and it constitutes the information which is exchanged between nodes, so that their localization can be performed by using trilateration [8], (*ii*) Distance Related Measurements, and (*iii*) RSS (Received Signal Strength) profiling. Moreover, in [5] they propose an indoor localization approach, called EZ localization algorithm which estimates the positioning of 2D point in terms of absolute coordinates: latitude and longitude.

III. BACKGROUND

In this section, we are going to describe our contribute to the localization problem in Wireless Sensor Network, by explaining how the range-free localization algorithm works, with the support of a simulation program, which generates an environment with nodes and obstacles where the signal power is represented and is used to locate a point. Before starting explaining our work in detail, (*i*) we briefly analyze the Wireless Sensor Networks and their topologies, (*ii*) we introduce the Received Signal Strength Indicator (RSSI), and (*iii*) we illustrate other localization techniques, such as the range-based and angle-based ones.

First, we introduce Wireless Sensor Networks, a system of nodes which exchanges data wireless. All this information can be possibly held and elaborated by a control center. As known, each net can have a particular topology, which characterizes the behavior of its component. In [1], they summarize essentially six kinds of network topologies:

1. Star topology: each node is connected to a single hub which filters any communication;
2. Ring topology: there isn't a leader, the information exchange follows one direction (the one of the ring);
3. Bus topology: there is a communication channel were all the information passes through;
4. Tree topology: hierarchical structure is the base of any communication;
5. Fully connected topology: each node is connected to any other node and this makes this topology suffer from NP-complexity;
6. Mesh topology: nodes have a regular distribution and each node communicates with its nearest neighbor.

Another important ingredient concerning localization is the Received Signal Strength Indicator (RSSI). This indicator provides the power of the received signal in a certain point and it has a strong relevance since not only it gives important knowledge for the purpose of localization, but it is also recognizable by any device on the market. For instance, *WirelessNetView* is an application which freely provides the percentage of the received signal by any retrieved hub.

We present the most famous approaches to estimate the position of a point in a Wireless Sensor Network. The initial classification we can introduce divides localization techniques in *anchor-based* and *anchor-free*: in the first approach, the network presents some special nodes, the *anchors*, which are aware of their position since they are equipped with a Global Positioning System, while all the other nodes, the *targets*, guess their location with respect to the anchors one; while someone actually prefers this kind of approach, such as in [13], some other authors have found some limitations in anchor-based algorithm, hence an anchor-free approach has been introduced. For instance, in [12] they suggest this kind of approach, since they indicate three reasons why the anchor-base algorithms are not the best choice: (*i*) there is a waste of time due to the manual insertion of anchor nodes; (*ii*) anchor-based algorithms are instable, since a small mistake in the anchors positioning may cause a huge mistake in the wireless sensor network final configuration; (*iii*) anchor-based algorithms are not scalable.

A. Range-based

Range-based algorithms can be classified into three main groups of approaches to the localization problem, which we are going to analyze. The first one is trilateration: this is a well known approach used by the Global Positioning System (GPS)

and it consists in the estimation of a point positioning by computing the intersection between four spheres, among which the Earth. Typically, the remaining three spheres are generated by satellites: for this reason, this technique is clearly optimal in outdoor environments, but it is not useful in case of indoor ones, or situations like "urban canyon", where points are not visible by satellites.

The second one is min-max: in order to locate a point *P*, each node creates a square around itself: the sides of this square are as far from the node as the distance between the node and *P*. This is made exploiting the values provided by RSSI. Eventually, the intersection between these squares is taken:

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \quad (1)$$

where x_i and y_i represent the coordinates of *i*-th point and *d* the distance between the *i*-th node and *P*. The estimation of the position of *P* is given by the center of the area obtained by the intersection, as explained in [9] and [10].

Finally, maximum likelihood: again, the distance between a node and the point *P* we want to locate is used, by exploiting the RSSI information. The aim here is to minimize the average quadratic error.

B. *Angle-based*

Angle-based localization technique provides a way to estimate the absolute position of a point in a given area: typically, the angular distance between nodes is computed and then used in order to estimate their position through trilateration.

Another possible application of this technique is the algorithm based on DoA (Direction of Arrival) shown in [11]. They explain how, given an antenna, the algorithm tries to estimate its location, by following three main steps:

1. Initialization: instantaneous angular estimate is computed;
2. Tracking: movements of the transmitter in the angular domain are traced in real-time;
3. Data mapping: angular estimates and other information about the height of the transmitter are used in order to project the antenna position over an indoor map.

IV. Our Approach

Our work focuses on *range-free* approach in order to provide a localization algorithm. This kind of localization technique uses some particular maps, called *fingerprinting radio-maps* where the signal power of each node is represented by its fingerprint (fingerprint-based techniques have been used in [19] and [21], too).

Since these maps are created through measurements of the retrieved signal in various points of the environment, the presence of obstacles has an impact on the signal power, as we can observe in Figure 1, where the variation of color intensity reflects the signal attenuation. Each device is represented by the black areas, and as the distance from it increases the strength of the signal decreases: this is expressed in a color variation from dark red to yellow. Moreover, we can observe that this variation is not regular nearby the obstacles (represented by the black lines): in fact, the presence of obstacles deforms the signal and lets the color turn into yellow more quickly (such as in the case of hub 2).

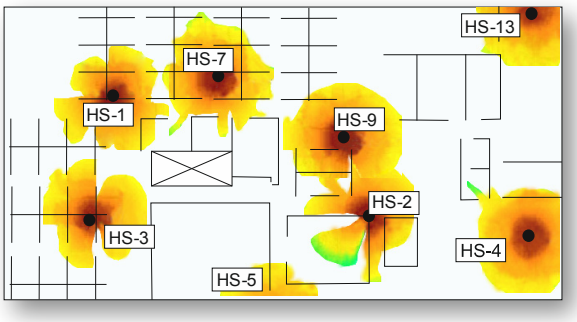


Figure 1: Generic fingerprinting radio-map with different nodes and variation of signal due to presence of obstacles

In this work, we propose an algorithm that uses this kind of maps and a simulation program to create them. For each point the power level is computed by using the inverse-square law:

$$P = \frac{P_M}{(x_i - x_n)^2 - (y_i - y_n)^2} \quad (2)$$

where (x_i, y_i) are the coordinates of one of the point, (x_n, y_n) are the coordinates of one possible point in the environment, P_M is the maximum signal for the node, and *P* is the computed signal power for that point with respect to that node. This law tells us that the signal power is inversely proportional to the square of the distance. Our simulation program works as follows: initially, it provides an environment in which we can put nodes and obstacles; then, it creates a list of points which represent the grid where we are going to simulate the retrieved power. For each obstacle, it picks a point and it generates for it the dimensions and an attenuation factor α which indicates how that obstacle influences the signal. In the end, a vector for each obstacle is obtained. Now, nodes have to be located and the program can do this by following three different node distributions:

- Random;
- Geometric;
- Signal-based.

Following a random distribution, nodes are placed randomly all over the environment, without any kind of optimization criteria.

With a geometric distribution, instead, the program tries to place the nodes in a geometric way in the environment, according to its the shape.

In the third case, with a signal-based distribution, the nodes are placed trying to optimize the signal spread over the environment. In this case, each insertion of a node in the area puts in doubt the previous placements if the signal could have been distributed in a better way.

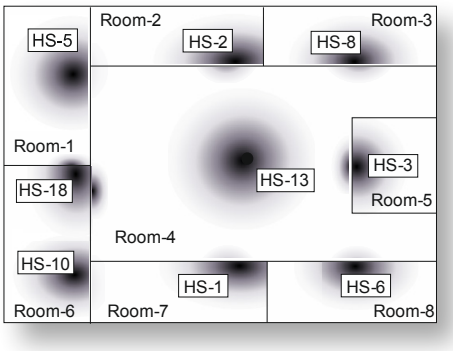


Figure 2: Example of a generated map with nodes and obstacles made by the simulation program

Subsequently, a vector for each point is generated, where each component represents the simulated signal power from a node of the network. By building the union of a specific component taken by all the points, we obtain a fingerprinting radio-map. In Figure 2, an example of how our simulation program generates the map is shown: we start from an environment divided into rooms surrounded by walls that surely alter the signal transmission. In order to have the best signal spread possible, hubs are placed trying to cover the greatest room of the environment till coverage of any room is guaranteed.

We show some code lines in order to explain how the simulation program works. It is executed in "mode 6", meaning some information are supposed to be given: parameters for map building, nodes position and obstacles features. The used variables are: *V* is the values variation percentage; *A* is the length of the map; *AN* is the number of points on a row; *B* is the height of the map; *BN* is the number of points on a column; *M* are the map points where the signal is evaluated; *OS1* and *OS2* are the central positions, radius and obstacles influence on the signal; *NOD* are nodes positions; *MPot* is the list of points signal powers from any node.

Procedure: Map generation

```
1: genmap(6,V,A,AN,B,BN,M,OS1,OS2,NOD,MF,MPot):-
2:    AN1 is A/AN,
3:    BN1 is B/BN,
4:    gengrid(M,A,AN1,B,BN1),
5:    map(Mpot,V,M,OS1,OS2,NOD),
6:    concatMeMpot(MF,M,MPot),!.
```

The procedure *gengrid* generates the list of points where we have to simulate the signal power; *map* creates the map with relative signal powers, taking one point at a time and generating for it a vector whose components depend on the nodes and obstacles positions; *concatMeMpot* concatenates the vectors of a list with those of another one. Now, the research algorithm which has to locate a point in the environment exploits the results of the simulation program. The behavior of our research algorithm can be summarized as shown in Figure 3.

Let us suppose that *P* is the point we want to find in the simulation of our algorithm: first, the procedure receives as input the signal power retrieved by the simulation program for the point *P* and the one for all the other points of the map. Then, the average (or alternatively the variance) is computed between the measurements and a comparison between the points of the map is performed, by using a tolerance τ given by the Mean Square Deviation (MSD). A list of coordinates of those points which satisfy the tolerance is obtained: if this list ends up being empty, the tolerance can be increased, by following the cyclic path; otherwise, the average between the coordinates of that list is computed in order to obtain a position for *P*, and the procedure terminates.

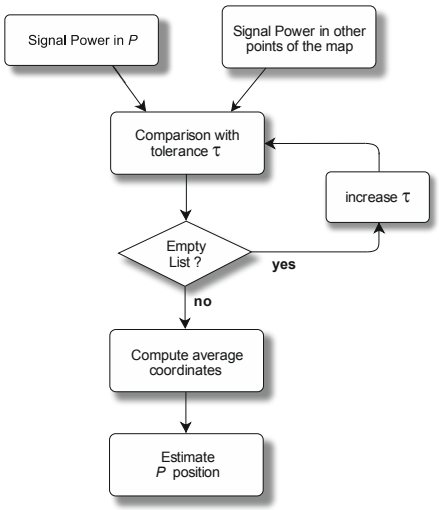


Figure 3: Control flow representing how the research algorithm works and finds an estimation of a point position

We show a code fragment below, where the research algorithm is explained in "mode 1", meaning that we assume that the map is given with points and power detections from any node retrieved in the point we want to find. First, we explain the used variables: *MF* represents map points with relative signal power from any node; *Pot* are measurements of power in the wanted point; *P* are coordinates close to the looked point; *List* is the list of map's points whose measurements are close to the wanted point; *Pots* is the tolerance; *Potm* is the average of powers of measurements in the point; *X* is the point we want to locate; *D* is the distance between the point we found and the one we wanted. Moreover, the procedure *average* computes the average between the vectors of powers; *meansqdev* computes the mean square deviation between vectors; *list* creates the list of points whose retrieved power is close to the one of the wanted point; *average1* computes the average of the list of points that should be close to the looked for point and it

increases the tolerance if needed; *dist* computes the distance between the found point and the wanted one.

Procedure: point research

```
1: search(1,..,MF,Pot,P,List,Pots,Potm,X,D):-
2:    average(Pot,Potm),
3:    meansqdev(Pot,Potm,Pots),
4:    list(List1,MF,Potm,Pots),
5:    average1(List1,List,MF,Potm,Pots,0,P),
6:    dist(X,P,D),!.
```

In this work we deal with localization problem by using a logic programming language: in both code pieces, we can see how the logic approach and Prolog programming language help us avoiding redundancy in computation. This is made thanks to the cut operator (!) that as soon as an advantageous computation branch is found, discards the other paths, in order to not analyze branches that would have led to useless solutions.

V. Evaluations and Case Studies

In this section, we provide some examples of how the model described above works and some analysis on it. First, we show in Figure 4 an example of how our simulation program fills a given environment by adding hub step by step.

A. Case Study

Let us suppose to have an environment with some obstacles as shown in Figure 4. Step by step, the program chooses where is preferable adding devices to the program have the best possible coverage. The program, as usually happens, places the first device in the middle of the hole environment, in order to have a good signal distribution. As we can observe, from Figures 4.1 and 4.b, the program tries to first cover the biggest area of the environment (the central one).

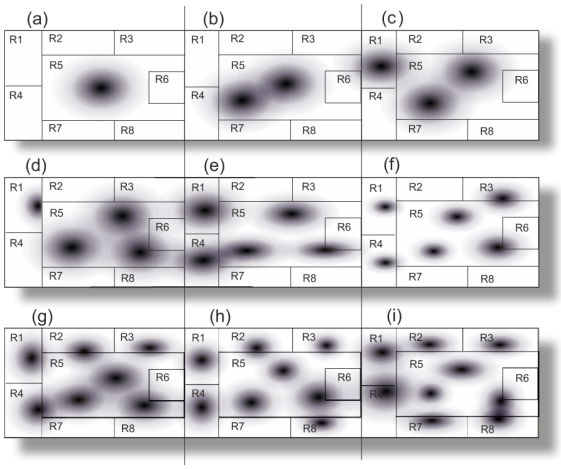


Figure 4: Evolution phases of the simulation program execution over an environment with some obstacles

In 4.c, the program decides to add a hub in one of the smaller rooms to have an improvement of signal distribution. After having covered the whole central area (Figure 4.d), the program starts adding devices in all the other rooms (Figures 4.e, 4.f, 4.g and 4.h), until it reaches the coverage of the entire environment and stops, as shown in Figure 4.i.

Moreover, we can see how the presence of obstacles determines a distortion in the signal shape, which is proportional to the attenuation factor α. This example is without any anchor nodes, but it is still possible adding some fixed node before the simulation start: in this case, clearly the addition of other hubs wouldn't have affected the position of anchors and thus it is likely that the final configuration of the network would have been different from the obtained one.

B. Evaluation

As we explained above, the second part of our work is the research algorithm, which estimates the position of a point in the environment, by exploiting the previous simulation program, based on fingerprinting radio-maps. We performed some analysis on how the variation of magnitude influences the precision of the algorithm. In particular, we consider three aspects: (*i*) number of involved nodes, (*ii*) grid dimensions, and (*iii*) environment size. In the following tables we show how the precision of our algorithm varies according to these parameters.

Table 1: variation of precision according to number of nodes

Nodes	**4**	**6**	**9**	**12**	**16**
Rand. Distr.	2,04m	1,72m	0,94m	0,59m	0,41m
Geom. Distr.	2,05m	1,45m	0,47m	0,33m	0,19m
Signal Distr.	2,02m	1,17m	0,48m	0,33m	0,2m

As we can observe from Table 1, the more the nodes are the more the precision of the algorithm grows. This does not mean that we can increase the number of nodes in an unchecked way, since we couldn't obtain an absolute precision: this is a consequence of the fact that measurements are made in map points which are in the detections grid too. This research gives as result all the points of the grid that are close to the point we are looking for.

Table 2: variation of precision according to grid dimensions

Row x Columns	**10x10**	**20x20**	**30x30**	**40x40**
Rand. Distr.	0,55m	0,41m	0,33m	0,29m
Geom. Distr.	0,42m	0,19m	0,15m	0,13m
Signal Distr.	0,43m	0,20m	0,15m	0,14m

As Table 2 shows, by increasing grid dimensions the precision increases. This is obvious, since there is a higher probability that points are close to the one we are looking for, during the comparison phase.

Table 3: variation of precision according to environment dimensions

Env. Dim. (mxm)	**5x5**	**10x10**	**15x15**	**20x20**
Rand. Distr.	0,12m	0,41m	1,16m	3,43m
Geom. Distr.	0,1m	0,19m	0,68m	1,84m
Signal Distr.	0,1m	0,2m	0,63m	1,97m

Finally, as we could expect, the more the environment grows the more the error increases, since each node influences just a small part of the entire environment, hence localization mistakes are more frequent (Table 3).

As we can clearly read by the tables above, the random distribution should be avoided, since it leads to less precise results; oppositely, both geometric and signal-based distributions provide solutions with a good precision, hence should be preferred to the random one.

VI. Conclusions and Future Work

In this work, we have initially described Wireless Sensor Networks and their main features and topology, with the aim to focus on the topic of point localization in indoor environments.

There are many techniques able to do this kind of localization, but in this work we try to propose a solution taking into account the environment features. Our localization algorithm is based on fingerprinting maps, and for this reason we have designed a simulation program which automatically generates them, rather than manually collect all useful information. These maps are generated by using RSS (Received Signal Strength) from any node, and we have considered three different node distributions: (*i*) random, (*ii*) geometric, and (*iii*) signal based. The approach is the logic one, based on backtracking: thanks to the Prolog cut operator (!), we are able to improve the computational effort of our algorithm, by pruning the useless computation paths, and just following the convenient ones. This technique allows us to provide the best solution according to the *global optimum*, rather than the *local* one: this is because at any insertion, previous decisions may possibly be changed, if a signal distribution improvement is supposed to be possible.

By analyzing our results, we have noticed that in order to have precise localization, a trade-off between environment dimensions, grid size, and number of involved nodes is necessary.

As hints for future works on this topic, we can observe that our simulation program can work in a 3D space as well, by providing adequate maps; moreover, it could also be opportunely integrated with a user interface and policy management tools [18], to facilitate its usage and control.

References

[1] Lewis, F. L. (2004). "Wireless sensor networks". Smart environments: technologies, protocols, and applications, 11, 46.

[2] Kumar, A., Shwe, H. Y., Wong, K. J., & Chong, P. H. (2017). Location-Based Routing Protocols for Wireless Sensor Networks: A Survey. *Wireless Sensor Network*, *9*(01), 25.

[3] Ssu, K. F., Ou, C. H., & Jiau, H. C. (2005). Localization with mobile anchor points in wireless sensor networks. *IEEE transactions on Vehicular Technology*, *54*(3), 1187-1197.

[4] Mao, G., Fidan, B., & Anderson, B. D. (2007). Wireless sensor network localization techniques. *Computer networks*, *51*(10), 2529-2553.

[5] Chintalapudi, K., Padmanabha Iyer, A., & Padmanabhan, V. N. (2010, September). Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (pp. 173-184). ACM.

[6] Parker, R., & Valaee, S. (2007). Vehicular node localization using received-signal-strength indicator. *IEEE Transactions on Vehicular Technology*, *56*(6), 3371-3380.

[7] Liu, C., Wu, K., & He, T. (2004, October). Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on* (pp. 516-518). IEEE.

[8] Rong, P., & Sichitiu, M. L. (2006, September). Angle of arrival localization for wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on* (Vol. 1, pp. 374-382). IEEE.

[9] Severino, R., & Alves, M. (2007, June). Engineering a search and rescue application with a wireless sensor network-based localization mechanism. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a* (pp. 1-4). IEEE.

[10] Savvides, A., Park, H., & Srivastava, M. B. (2002, September). The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 112-121). ACM.

[11] Belloni, F., Ranki, V., Kainulainen, A., & Richter, A. (2009, March). Angle-based indoor positioning system for open indoor environments. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on* (pp. 261-265). IEEE.

[12] Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003, November). Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 340-341). ACM.

[13] Mourad, F., Snoussi, H., Abdallah, F., & Richard, C. (2009). Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, *57*(8), 3226-3239.

[14] Balzano, W., Del Sorbo, M. R., Murano, A., & Stranieri, S. (2016, November). A logic-based clustering approach for cooperative traffic control systems. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 737-746). Springer, Cham.

[15] Balzano, W., Del Sorbo, M. R., & Stranieri, S. (2016, March). A logic framework for c2c network management. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on* (pp. 52-57). IEEE.

[16] Balzano, W., Murano, A., & Stranieri, S. (2017). Logic-based clustering approach for management and improvement of VANETs. *Journal of High Speed Networks*, *23*(3), 225-236.

[17] Balzano, W., & Stranieri, S. (2017, November). LoDGP: A Framework for Support Traffic Information Systems Based on Logic Paradigm. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 700-708). Springer, Cham.

[18] Giordano, M., Polese, G., Scanniello, G., Tortora, G. (2010, February). A System for Visual Role-Based Policy Modelling: In *International Journal of Visual Languages and Computing*, Vol. 21, No. 1, Elsevier, pp. 41-64.

[19] Balzano, W., Murano, A., & Vitale, F. (2016, March). WiFACT--Wireless Fingerprinting Automated Continuous Training. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on* (pp. 75-80). IEEE.

[20] Balzano, W., Murano, A., & Vitale, F. (2016). V2V-en–vehicle-2-vehicle elastic network. *Procedia Computer Science*, *98*, 497-502.

[21] Balzano, W., Murano, A., & Vitale, F. (2018). SNOT-WiFi: Sensor network-optimized training for wireless fingerprinting. *Journal of High Speed Networks*, *24*(1), 79-87.

On the Problem-Oriented Verification of Cyber-Physical Systems Using System-Level Test Sequences

Changlan Fu, Xiao Zhang, Zhi Li*, Ziyan Zhao, Chao Wang, Yuekun Yu
College of Computer Science and Information Technology
Guangxi Normal University
Guilin, Guangxi, China
*corresponding author, zhili@gxnu.edu.cn

***Abstract*—The heterogeneous implementations of Cyber-Physical Systems (CPS), including complex behaviors of physical devices and human users, pose significant challenges for verifying such systems. Since Jackson's Problem Frames approach (PF) provides facilities for representing interactions between the computing and physical components of CPS, it is applicable to the requirements analysis and modeling of CPS. In this work, we propose an approach to verifying whether the requirements are satisfied or not using system-level testing methods for CPS. A set of supporting tools have been developed for modeling and verifying CPS with test obligations and acceptance criteria for system-level testing prior to design time, which aims at reducing defects in requirements elicitation and documentation, thus supporting backtracking activities in the requirements analysis of a software development process. The work is illustrated in a real-world example.**

Keywords-Problem Frames; requirements verification; test sequences; cyber-physical systems;

I. INTRODUCTION

With the development of information technology, we will see increasing use of the Internet of Things, big data analytics platforms, and appearances of Cyber-Physical Systems (CPS)[1], which is a series of tight integration of computational processes and physical processes. Deep integration and real-time interaction are achieved through feedback loops in which computational processes and physical processes interact. With the increasing complexity of CPS, the testing and maintenance of such systems become very difficult, which may cause faults, failures or even great security risks.

Recently, some researchers use heterogeneous model fusion technology to achieve integration of the computing and physical systems[2]. For example, modeling languages such as UML, Modelica, and Simulink have extended their modeling elements. Although existing CPS modeling and simulation techniques reflect some advances in its verification technology, they are far from being able to meet the needs for verifying large-scale CPS design processes.

The Problem Frames (PF)[3] approach, which was first proposed by Michael A. Jackson in the field of software engineering, has established that software development problems can be divided into three parts: software *S*, real world *W*, and user requirements *R*. *S* represents a software system to be built; *W* represents the real-world environment in which the software system runs; *W* can be regarded as the physical component in the CPS architecture. PF provides facilities for representing the interactions between the computing and physical components of CPS, thus supporting the modeling and verification of complex CPS behaviors[4,5].

In this paper, we model the behavior of CPS and its requirements based on PF, and verify whether the integration of its computing and physical systems meets the user's requirements in terms of completeness and correctness, which can help find system defects and avoid serious failures. We propose a system-level testing method for CPS, and develop a set of supporting tools which can systematically assist software developers in modeling, and verifying cyber-physical systems prior to design time.

II. SYNTACTIC CHECKING OF PF DIAGRAMS

In PF, the computing machine domain is represented by the symbol  , the problem domain is represented by the symbol  and the requirement is represented by the symbol  . The connecting lines between the machine domain and problem domains are called interfaces, represented by the symbol ; the connecting lines between the requirements and problem domains can either be requirements references (represented by the symbol) or requirement constraints (represented by the symbol), as shown in Figure 1 (on the next page).

A. Integrity and correctness in PF syntax

The integrity of PF diagram refers to a set of basic completeness rules which are fundamental to PF models. Table 1 lists a sample of rules for PF. Note that rules can be accumulated and added as practitioners gain more experience of using the PF modeling.

Table 1. Integrity conditions of the problem diagram

No.	Integrity conditions
1	The name of the domain must be not be empty.
2	There is at least one machine domain in the diagram.
3	A problem diagram has at least one requirement.
……	……

DOI reference number: 10.18293/DMSVIVA2018-021

The correctness of PF diagram refers to the fundamental principles that must be obeyed in PF[3], as shown in Table 2.

Table 2. Correctness conditions of the problem diagram

No.	Correctness Conditions
1	The name of the domain must be unique.
2	The requirement cannot directly constrain the machine domain.
3	Each machine domain controls at least one interface.
……	……

When system analysts model the requirements, they hope that the model can help the requirements analysts to check if the design is complete and correct, so as to avoid the situation in which the design is difficult for the developers to understand. Here is an example of an incomplete and incorrect problem diagram, as shown in Figures 1 and 2. One problem domain in Figure 1 has no name, and the requirements in Figure 2 directly constrains the machine domain (by definition, a requirement that directly constrains the computing machine is called a "specification").

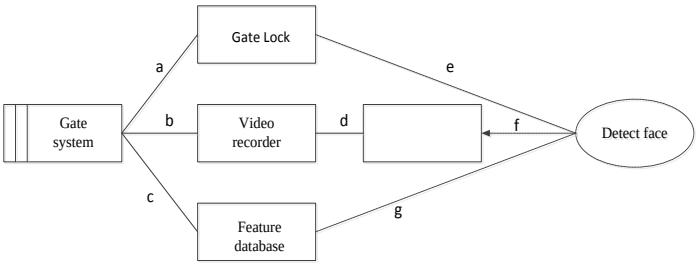


Fig. 1. A security gate control problem diagram (domain nameless)

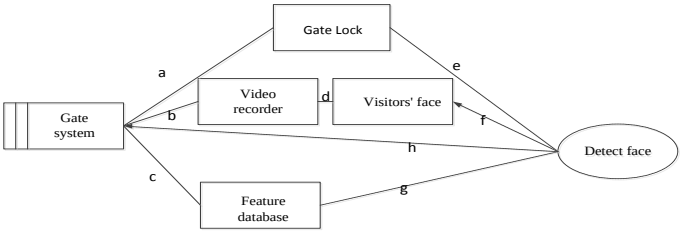


Fig. 2. A security gate control problem diagram (requirements directly constrain the machine)

B. A brief description of OCL

The object constraint language (OCL) is a language for applying constraints on specified model elements. OCLs express constraints on the object with the conditions and constraints attached to the model elements, including invariant or constraint expressions attached to the model elements, pre-conditions and post-attachments attached to operations, and methods and conditions, etc[6-7].

C. Using OCL to implement integrity and correctness check

We adopted the OCL's consistency verification method for UML models. We use OCLs to define PF constraints, and to develop the integrity and correctness verification modules by analyzing and checking OCL expressions[8]. The problem diagram constraint conditions in Tables 1 and 2 are represented by the following OCLs.

Constraint: The name of the domain must be set

OCL expression:

```
Class.allInstances()-
>select(c|c.oclAsType(Class).getValue(c.oclAsType(Class).ge
tAppliedStereotypes()->asSequence()->first(),'value')=null) -
>size()=0
```

Constraints: There is at least one machine domain in the context diagram

OCL expression:

```
Class.allInstances()-
>forAll(p|p.oclAsType(Class).getAppliedStereotypes().name-
>includes('Machine')->size()>=1)
```

Constraint: A problem diagram or framework has at least one requirement

OCL expression:

```
Class.allInstances()-
>forAll(p|p.oclAsType(Class).getAppliedStereotypes().name-
>includes('Requirement') ->size()>=1)
```

Constraint: The name of the domain must be unique

OCL expression:

```
Class.allInstances()-
>select(getAppliedStereotypes().name->includes('Domain'))-
>isUnique(name)
```

Constraint: The requirement cannot directly constrain the machine domain

OCL expression:

```
Interface.allInstances()-
>select(a|a.oclAsType(Dependency).getAppliedStereotypes().
name->includes('constrains'))-
>forAll(source.getAppliedStereotypes().name-
>includes('Requirement')implies not
target.getAppliedStereotypes().name->includes('Machine'))
```

Constraint: Each machine domain controls at least one interface

OCL expression:

```
Interface.allInstances()-
>select(getAppliedStereotypes().name-
>includes('observes')).target-
>forAll(ot|Interface.allInstances()-
>select(getAppliedStereotypes().name->includes('controls'))-
>select(target->exists(ct|ct=ot))->size()=1)
```

III. COMPLEX PROBLEM DECOMPOSITION

Hall provides a de-notational semantics for Problem Frames in [9], in which a generic problem diagram can be expressed as follows:

$c, o : [K, R]=\{S \mid S \text{ controls } c \wedge S \text{ observes } o \wedge K, S \vdash_{DRDL} R\}$,

where S represents the software *solution* to be built, K represents knowledge about S's context (i.e., physical devices or human-beings) and R represents user requirements. Figure 3 shows the corresponding generic problem diagram (Note: $S!c$ denotes "S controls c", $S?o$ denotes "S observes o", and *DRDL* is the short for a requirement and domain description language.

Fig. 3. A generic problem diagram

In this paper, a chain of causality is introduced to facilitate the understanding of complex problem diagrams and to automate the search for causal chains [10].

A. Causal chain

From Figure 3, we can see that if the domain sharing phenomenon *o* results in the occurrence of *c*, then this is a causal relationship.

In a problem diagram, if we find a path from *R* to *S*, or from *R* to *S and to R*, then we say that we have found a solution to the problem. The elements of this path populate the set of solutions to the problem. There are multiple such paths in a complex problem diagram, that is, there are multiple chains of causality from R to S, and each path may be respectively represented as R_1, S_1, R_2, S_2,... R_n, S_n .

We extend Hall's de-notational semantics by introducing the causality chain concept, as follows:

$$c, o : [K, R] = \{S \mid S\ !\ c \wedge S\ ?\ o \wedge K, S \vdash_{DRDL} R\}$$
$$= \{\ S1 \| S2 \| \ldots \| Sn\ \}$$

Figure 4 corresponds to a partial problem diagram, which shows a set of solution.

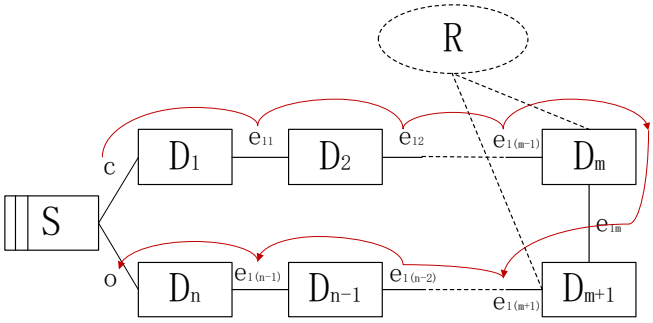


Fig. 4. A partial problem diagram

IV. Security Gate Control Problem Diagram: A Case Study

A security gate control problem is used to illustrate the work presented in this paper. The following is a rough sketch of the problem:

A computer that recognizes facial features is required to control the security gate. The face of each person who wants to enter the security gate is captured on a video tape. The records in the database are compared with the captured face features. These records contain facial features that have been explicitly accessible. Figure 5 is a diagram of the security gate control problem.

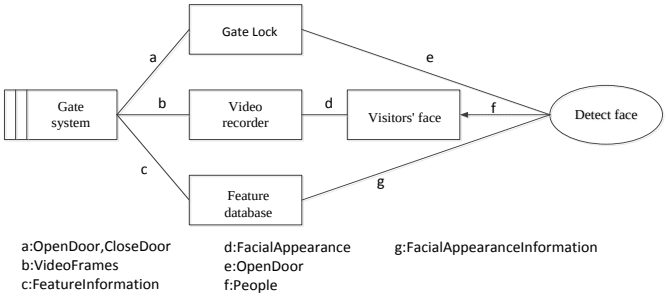


Fig. 5. Security gate control problem diagram

By comparing Figures 3 with 5, we can find the relationship between the problem diagram and Hall's de-notational semantics and two chains of causality, as shown in Figure 6.

S ={ Gate system },

c ={a}(b is initialized by S and therefore controlled by S),

o ={b,c}(b,c is received by S, so it is observed by S),

K ={ Gate lock, Video recorder, Visitors' face, feature database },

R ={ Detect face },

d ={e,f,g}.

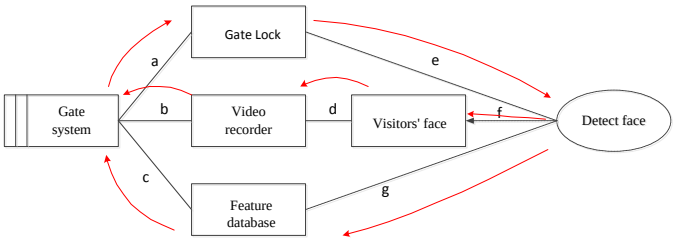


Fig. 6. Security gate control problem diagram causality chain

V. Requirements Satisfaction Verification Study

Precise mathematical methods and technologies are used in formal requirements verification, in which requirement models, are represented by mathematical expressions, operations and derivations so that ambiguities, incompleteness, unachievable expressions in requirement models can be detected or dicovered. The research in this paper is based on the Communication Sequential Process (CSP) algebraic theory and CSP scripts are generated for the description and verification of the requirements model.

A. Overview of Communication Sequential Process

The Communication Sequential Process is an algebraic theory proposed by the well-known computer scientist C.A.R. Hoare[11]. It is an abstract description language for parallel algebraic systems and specifically describes message interactions in concurrent systems. Because CSP is suitable for modeling and analyzing systems and describes complex message interactions, it is widely used.

B. Mapping of Problem Diagrams to Sequence Diagrams

Sequence diagrams are used to describe the sending of messages between objects. It can intuitively convey the interaction of various parts of the system[12]. For example, in the chains of causality in Figure 6, f->d->b->a->e and g->c->a->e, the requirement is used as a starting point and an ending point, and it is converted into a sequence diagram, as shown in Figures 7 and 8 .

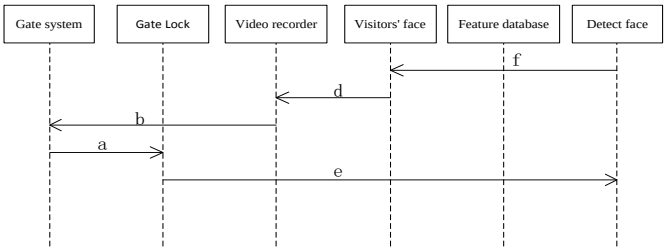


Fig. 7. A Sequence diagram

In these two figures, you can clearly see the triggering sequence of the sharing phenomena between various objects. The objects in the sequence diagram correspond to the domain and requirements of the problem diagram, while the objects in the sequence diagram can also be mapped to the processes in the CSP[13]. Here is an example of a vending machine for the reader to understand the mapping relationship between the problem diagram and CSP, as follows:

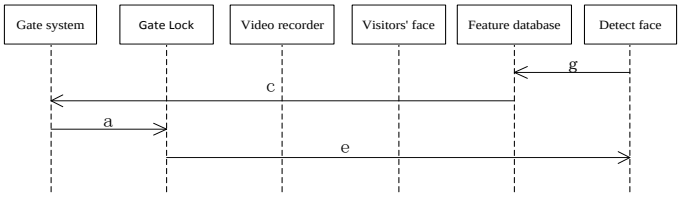


Fig. 8. A Sequence diagram

A vending machine (VM) receives a coin inserted by a customer (CUST) and automatically gives chocolate (choc) or coffee according to the customer's purchase request and choice.

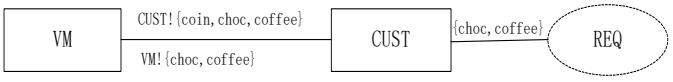


Fig. 9. The vending machine problem diagram

According to the problem description, the vending machine problem can be described as follows:

VM=coin→(choc→VM|coffee→VM)，
CUST=coin→(choc→CUST|coffee→CUST)，

which defines two process VM and CUST, where coin, choc and coffee are all events. From the CSP description and Figure 9,we can see that the domains in the problem frames can be mapped to the CSP processes, and the sharing phenomenon can be mapped to messages or events passed between processes. By studying this kind of mapping relationship, we can convert the sub-problems obtained through problem de-composition, that is, the causality chain, into a sequence diagram, and we have developed a tool which can automatically generate a CSP script based on the objects in the sequence diagram and message passing.

C. CSP script generation based on sequence diagram

The object in the sequence diagram is regarded as a process in CSP. For example, in Figure 7, the process of detecting face begins first, and a new process is generated by the process and the event of the operation. The CSP is expressed as: Detect face =f-> Visitors' face. Similarly, the sequence diagram in Figure 7 can be described using CSP as the following script code:

Detect face =f- > Visitors' face
Visitors' face =d-> Video recorder
Video recorder =b-> Gate system
Gate system =a-> Gate lock
Gate lock =e->STOP

In summary, the process of converting a problem diagram into an algebraic expression is demonstrated, and this abstract algebraic symbol can be run by a specialized CSP verification tool, FDR[17].

D. Requirements verification

A complex problem diagram is transformed into CSP scripts that run in the FDR tool. The requirements engineer can write code to verify that the requirements are satisfied or not. The support tools implemented in this paper integrate the FDR4 tool to automate the verification of CSP scripts. The FDR4 is a tool that can be used for model verification developed by scholars at Oxford University in the UK.

VI. RESEARCH ON REQUIREMENTS TESTING METHOD BASED ON PROBLEM FRAMES

Today's software systems are large and complex, and the task of software testing becomes error-prone and complicated. If the test task is clearly identified as early as possible, then the quality of requirements will be greatly improved.

A. Definition of Requirements Test

The requirements analysis model can be used as the guidance for documenting specifications in order to help the requirements analyst understand and help developers in system development. The requirements model can also be used as a test model[14].

B. Extending Problem Frames

This study needs to add a causal relationship attribute to each domain to record the triggering relationships between the sharing phenomenon related to the domain. For example, if there are two shared phenomena *a* and *b* in the machine domain and phenomenon *a* triggers the occurrence of the *b* phenomenon, then *a->b* is recorded. We need to expand the domain constraint attributes to record one-to-many or many-to-many trigger conditions. For example, in a ATM system, where the depositor withdraws money less than the amount of the account, then the cash is ejected; if the withdrawal amount is greater than the amount of the account, then a display will show that it cannot be withdrawn. This study uses the syntax of

object-constraint language to record these domain-constraint attributes. For example, the record is as follows:

Account (balance), Withdrawal amount (amount)

Pre process: balance>amount and amount >0

Post process:(balance=balance@Pre-amount) and balance>0

C. Generating test scenarios

The causality chain is a method of splitting complicated problem diagrams. The sub-problems obtained are a use case of the problem. Therefore, the sequence of causality chains is a test scenario. Testers can design test cases based on test scenarios generated by causali chains and constraints. Once a system fails, only the physical systems and computing systems related to the fault need to be tested. For example, the test trail A->B[@balance>amount and a mount >0@(balance=balance-amount) and balance>0]->C->D that with OCL constraint description and test trail A->B->C->D that without OCL constraint description, where A, C, and D are physical components and B is control software. Assume that the C physical device has a failure, the tester only needs to test all the devices on the causal chain containing the shared phenomenon triggered by B. If the system is working properly and only the result of the operation is different from what is expected, the design of the test case can be based on the pre-constraint and post-constraint conditions of B, thus facilitating the testing and maintenance of the later system.

D. Generating Test Cases

The requirement references and constraints of PF are respectively represented by dashed lines without arrows and dashed lines with arrows. The dashed line with an arrow indicates that this requirement refers to the phenomenon in the problem domain. The dashed line with an arrow indicates that the requirement reference is a constraint reference, this requirement not only refers to the domain phenomenon, but also provides some desired relationships or behaviors that involve these relationships. In layman's terms, the former refers to a desired value or event, the latter defines a value or event to be obtained, like the input and output in the program. Therefore, we can develop the use case template shown in Table 3 below.

Table 3 Test Case Template

Test scenarios	Requirement reference	Requirement constraints

VII. IMPLEMENTATION OF SUPPORT TOOLS

The support tools developed in this study employ a client/server (C/S) and browser/server (B/S) hybrid architecture, which contains features such as good openness, easy expansion and transplantation [15].

A. Software Architecture Diagram of C/S and B/S Mixed

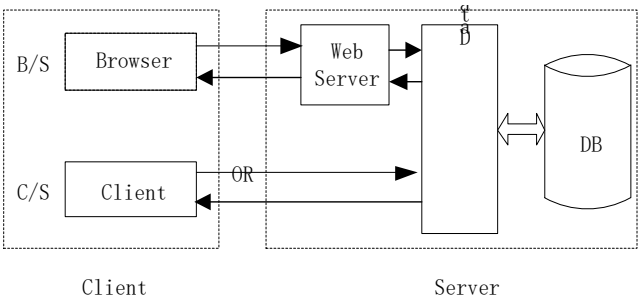


Fig.10. Support tool architecture diagram

B. Main functions of the tool

In addition to the basic function of drawing problem diagrams, this tool can also implement the function of checking the diagram for integrity and correctness. It can also automatically search and find causal chains and convert them into CSP scripts and verify the model automatically. Then test scenarios with constraints can be generated from these causal chains.

The tool can allow the problem diagram model to be saved in the XML format. Users can upload their own drawings on to the cloud server database. When modification is needed, the XML file can be opened from the database again. We first use the tool to draw the security gate control problem diagram (shown in Figure 11), and then the tool can automatically check the correctness and integrity of the diagram .

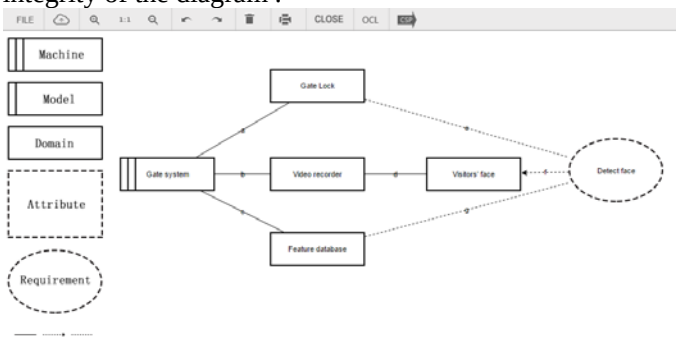


Fig.11. Drawing the right question diagram

After drawing the problem diagram, we click on the OCL button in the Tools menu to automatically verify that the problem diagram is complete and correct, as shown in Figure 12 below. If an incomplete or incorrect problem diagram is drawn, as shown in Figure 13, a domain has no name. The problem diagram is checked for completeness and correctness. The results are shown in Figure 14.

Once the diagram is verified to be complete, we can use the tool to find all the causal chains from the problem diagram (Figure 15).

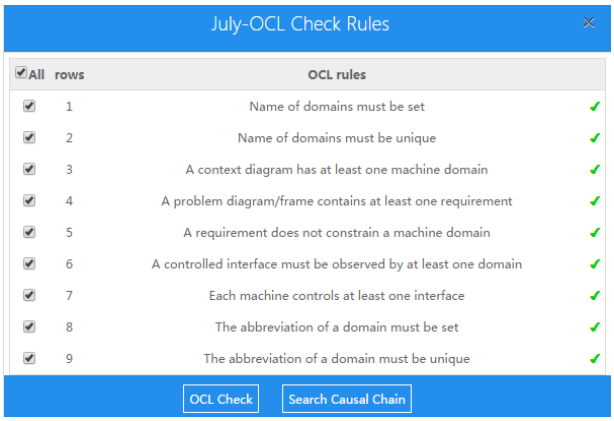


Fig.12. Screenshot of OCL check result

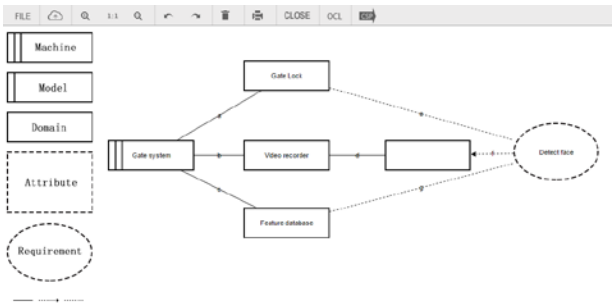


Fig.13. Draw error question diagram

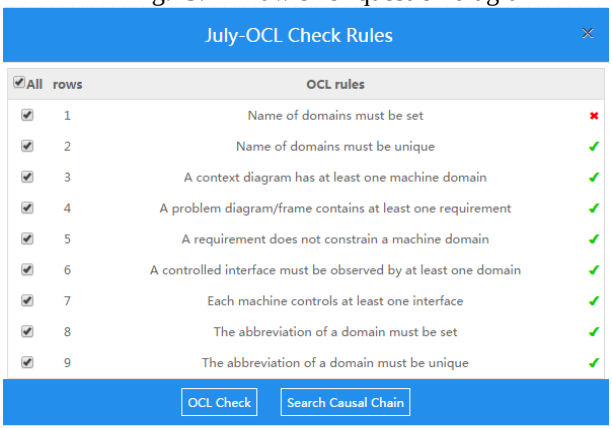


Fig.14. Screenshot of the check question diagram

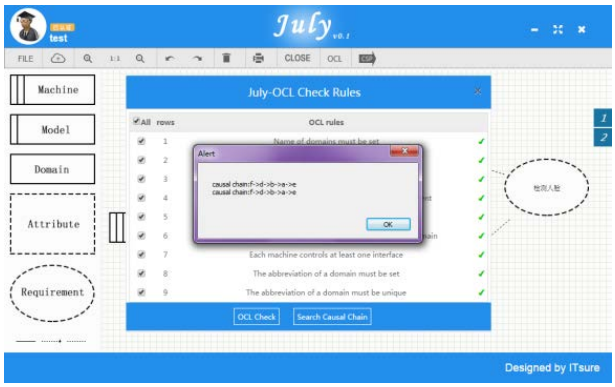


Fig.15. The result of Finding causality chain

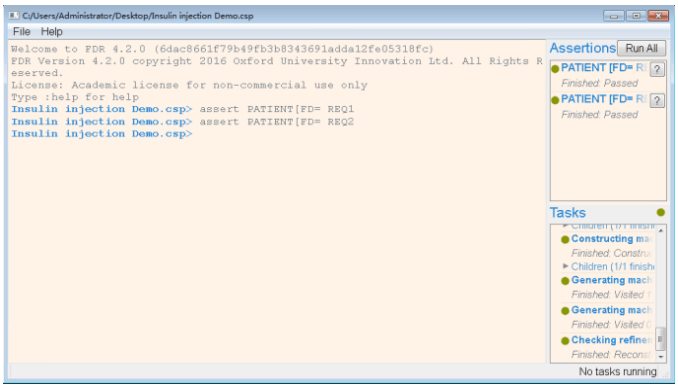


Fig. 16. The result of FDR check

Fig.17. generated test leads

Our tool can help system analysts to check if user requirements are satisfied or not by running the FDR4 tool, as shown in Figure 13. Based on those causality chains we can generate test scenarios, as shown in Figure 17.

VIII. CONCLUSIONS

In this paper, we provide a solution to the problem of automatically verifying CSP behaviors and user requirements, and searching and finding causal chains which helps de-compose a complex problem into sub-problems, and transform a problem diagram into a formal scripting language to verify whether the cyber-physical system design can satisfy end-to-end requirements[16]. Test scenarios for the system can be generated based on the causality chains. Testers can derive test cases from these test scenarios with constraints to improve the test efficiency. This paper demonstrates the feasibility of the proposed method by applying the support tools we develop in the case study of a safety gate control problem. Our case study shows that our method contributes to reducing defects in the requirements analysis phase and increasing the success rate of software development projects.

ACKNOLEDGEMENTS

The authors would like to thank the anonymous referees for their valuable comments and suggestions. This work is partially supported by the National Natural Science Foundation of China (61262004), Guangxi “Bagui Scholar”

Teams for Innovation and Research, the Project of the Guangxi Key Lab of Multi-source Information Mining & Security (Director's grant 14-A-03-01, 15-A-03-01), Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Innovation Projects of Guangxi Graduate Education (No. XYCSZ2017066), (No. XYCSZ2018075), (No. XJGY201809), 2017 Guangxi Normal University Bilingual Course Project (No. A-0201-00-00013F).

REFERENCES

[1] Wen J R, Mu-Qing W U, Jing-Fang S U. Cyber-physical System[J]. Acta Automatica Sinica, 2012, 38(4):507-517.

[2] Dongfang Liang, Yuying Wang, Xingshe Zhou, et al. Simulation modeling method based on heterogeneous model fusion for CPS system[J]. Journal of Computer Science,2012, 39(11):24-28.

[3] Jackson M. Problem frames: analyzing and structuring software development problems[M]. Addison-Wesley Longman Publishing Co. Inc. 2000.

[4] M. Jackson. System Behaviours and Problem frames: Concepts, Concerns and the Role of Formalisms in the Development of Cyber-physical Systems[M]. Dependable Software Systems Engineering,2015:79–104.

[5] Xiaohong Chen, Bin Yin, Zhi Jin. Demand Modeling Based on Problem Frames: A Method of Present System Guidance[J]. Journal of Software, 2011, 22(2):177-194.

[6] Gogolla M. Object Constraint Language[J]. 2016.

[7] Zefan Jiang, Linzhang Wang, Xuandong Li, et al. Test method based on UML sequence diagram[J]. Computer Science, 2004, 31(7):131-136.

[8] Queralt A, Teniente E. Verification and Validation of UML Conceptual Schemas with OCL Constraints[J]. Acm Transactions on Software Engineering & Methodology, 2012, 21(2):1-41.

[9] Li Z, Hall J G, Rapanotti L. On the systematic transformation of requirements to specifications[J]. Requirements Engineering, 2014, 19(4):397-419.

[10] Jackson M. Where, Exactly, Is Software Development?[M]// Formal Methods at the Crossroads. From Panacea to Foundational Support. Springer Berlin Heidelberg, 2003:115-131.

[11] C. A. R. Hoare, "Communicating Sequential Processes," The Origin of Concurrent Programming, Springer New York, pp 413-443, 2002.

[12] Yuzhen Wang, Wei Dong, Huowang Chen. Automatic verification of UML sequence diagram[J]. Computer Engineering and Applications, 2003, 39(29):80-83.

[13] Shuo Zhang, He Zi, Zhi. Utilizing the Sequence chart in problem frames[J]. Research Reports: se, 2013, 2013: 1-8.

[14] Gao M, Zhong D, Lu M, et al. Research on test requirement modeling for software-intensive avionics and the tool implementation[C]// International affiliation, maintenance, safety study meeting session.2007:6.D.2-1-6.D.2-10.

[15] Chen X, Liu J L. Analysis and Comparison between the Structures of Client/Server and Browser/Server[J]. Journal of Chongqing Institute of Technology Management, 2000.

[16] Seater R, Jackson D. Problem frames transformations: deriving specifications from requirements[C]// International Workshop on Advances and Applications of Problem frames. ACM, 2006:71-80.

[17] Gibson-Robinson T, Armstrong P, Boulgakov A, Roscoe A.W, Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science, volume 8413, pages.187-201, 2014.

Author Index

Balzano, Walter	51, 119
Bellini, Pierfrancesco	44
Cenni, Daniele	44
Chakraborty, Mihir	91
Chang, Shikuo	65, 71
Chen, Cuiling	71
Choudhury, Lopamudra	91
Clarizia, Fabio	8
Colace, Francesco	8, 31
Cuzzocrea, Alfredo	83
Eloe, Nathan	109
Fu, Changlan	125
Guo, Wei	71
J. Fonseca, Manuel	23, 57
Kim, Jung	65
Kong, Jun	101
Leopold, Jennifer	109
Li, Zhi	125
Liu, Yufeng	101
Lombardi, Marco	8, 31
M. Alarcão, Soraia	23, 57
Marazzini, Mino	44
Maresca, Paolo	15
Mitolo, Nicola	44
Molinari, Andrea	15
Nesi, Paolo	36, 44
Paolucci, Mchela	36
Paolucci, Michela	44
Pascale, Francesco	8, 31
Pavão, Ruben	23
Santaniello, Domenico	31
Shi, Zhan	101
Stapleton, Gem	1, 91

Stranieri, Silvia 119

Taylor, Patrick 109

Vitale, Fabio 51

Wang, Chao 125
Wang, Yingfeng 101
Wen, Nannan 71

Yu, Yuekun 125

Zeng, Xiaoqin 101
Zhang, Kang 101
Zhang, Xiao 125
Zhao, Ziyan 125
Zheng, Hanzhong 65
Zou, Yang 101

Program Committee

Bilal Alsallakh	BOSCH Research
Flora Amato	University of Naples
Danilo Avola	Sapienza University of Rome
Andrew Blake	University of Brighton
Paolo Bottoni	Sapienza University of Rome
Paolo Buono	University of Bari Aldo Moro
Loredana Caruccio	University of Salerno
Maiga Chang	Athabasca University
Shikuo Chang	Uinversity of Pittsburgh
Shikuo Chang	Uinversity of Pittsburgh
Shikuo Chang	Uinversity of Pittsburgh
Peter Chapman	Edinburgh Napier University
William Chu	Department of Computer Science and Information Engineering, TungHai University
Yuan-Sun Chu	National Chung Cheng University
Mauro Coccoli	DIBRIS - University of Genoa, Italy
Kendra Cooper	Independent
Gennaro Costagliola	Dipartimento di Informatica, Università di Salerno
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria
Sergiu Dascalu	University of Nevada, Reno
Aidan Delaney	University of Brighton
Vincenzo Deufemia	Department of Computer Science, University of Salerno
Tiansi Dong	Bonn-Aachen International Center for Information Technology B-IT
Martin Erwig	Oregon State University
Filomena Ferrucci	Università di Salerno
Andrew Fish	University of Brighton
Daniela Fogli	Università di Brescia
Manuel Fonseca	Universidade de Lisboa
Rita Francese	University of Salerno
Kaori Fujinami	Tokyo University of Agriculture and Technology
David Fuschi	Bridging Consulting Ltd
Angelo Gargantini	University of Bergamo
Angela Guercio	Kent State University at Stark
Pedro Isaias	The University of Queensland
Joaquim Jorge	IST/UTL/INESC-ID
Jun Kong	North Dakota State University
Robert Laurini	INSA Lyon
Jennifer Leopold	Missouri University of Science & Technology
Lian Li	Lanzhou University
Zhi Li	Guangxi Normal University
Fuhua Lin	Athabasca University
Hong Lin	University of Houston-Downtown
Alan Liu	National Chung Cheng University
Jonathan Liu	University of Florida
Paolo Maresca	University of Naples Federico II

Luana Micallef	Helsinki Institute for Information Technology HIIT
Andrea Molinari	University of Trento
Max North	Southern Polytechnic State University
Joseph Pfeiffer	New Mexico State University
Antonio Piccinno	University of Bari
Giuseppe Polese	University of Salerno
Elvinia Riccobene	Computer Science Dept., University of Milan
Michele Risi	University of Salerno
Peter Rodgers	University of Kent
Teresa Roselli	Department of Computer Science - University of Bari
Veronica Rossano	Depatment of Computer Science - University of Bari
Giuseppe Santucci	Sapienza University of Rome
Monica Sebillo	Dipartimento di Informatica - Università di Salerno
Lidia Stanganelli	PhD - Researcher
Gem Stapleton	University of Brighton
Genny Tortora	Department of Computer Science- University of Salerno
Atsuo Yoshitaka	Japan Advanced Institute of Science and Technology
Tomas Zeman	Czech Technical University in Prague
Kang Zhang	The University of Texas at Dallas

Additional Reviewers

Dewan, Ali
Smeltzer, Karl