DMSVIVA 2019

**Proceedings of the 25th
International Distributed
Multimedia Systems
Conference on
Visualization and
Visual Languages**

**Lisbon, Portugal
July 8 - 9, 2019**

PROCEEDINGS
DMSVIVA2019

**The 25th International DMS Conference on
Visualization and Visual Languages**

Sponsored by

KSI Research Inc. and Knowledge Systems Institute, USA

Technical Program

July 8 to 9, 2019

Hotel Tivoli, Lisbon, Portugal

Organized by

KSI Research Inc. and Knowledge Systems Institute, USA

Copyright © 2019 by KSI Research Inc. and Knowledge Systems Institute, USA

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher.

ISBN: 1-891706-49-7

ISSN: 2326-3261 (print)

2326-3318 (online)

DOI: 10.18293/DMSVIVA2019

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://ksiresearch.org/seke/dmsvlss19.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 (DMSVIVA2019)*, we would like to welcome you to the beautiful city of Lisbon, Portugal. This 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 extended deadline of 14 April 2019, the conference received 22 submissions. All the papers were rigorously reviewed by three to five members of the international Program Committee. Based on the review results, 13 papers have been accepted as regular papers with an acceptance rate of 59%, and 3 as short papers and 4 as short papers with an acceptance rate of 13% and 18% respectively. We would like to thank all the authors for their contributions. We also would like to thank all the Program Committee members for their careful and prompt review of submitted papers.

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 invited speaker, Dr. Robert Laurini, for his inspiring talks, and to the members of the Publicity for their publicity efforts.

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.

Joseph J. Pfeiffer, Jr., New Mexico State University, USA; Program Chair

Vincenzo Deufemia, University of Salerno, Italy; Program Co-Chair

Andrew Fish, University of Brighton, UK; Program Co-Chair

DMSVIVA2019

The 25th International DMS Conference on Visualization and Visual Languages

July 8 and 9, 2019

Hotel Tivoli, Lisbon, Portugal

Conference Organization

DMSVIVA2019 Conference Chair and Co-Chairs

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

DMSVIVA2019 Steering Committee Chair

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

DMSVIVA2019 Steering Committee

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

DMSVIVA2019 Program Chair and Co-Chair

Joseph J. Pfeiffer, Jr., New Mexico State University, USA; Program Chair
Vincenzo Deufemia, University of Salerno, Italy; Program Co-Chair
Andrew Fish, University of Brighton, UK; Program Co-Chair

DMSVIVA2019 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
 Vittorio Fuccella, University of Salerno, Italy
 Jun Kong, North Dakota State University, USA
 Robert Laurini, University of Lyon, France
 Jennifer Leopold, Missouri University of Science & Technology, USA
 Alessio Malizia, University of Hertfordshire, UK
 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
 Franklyn Turbak, Wellesley College, USA
 Giuliana Vitiello, University of Salerno, Italy

Subcommittee on Sentient and Distributed Multimedia Systems

Flora Amato, Univ. of Salerno, Italy
 Arvind K. Bansal, Kent State University, USA
 Andrew Blake, University of Brighton, UK
 Loredana Caruccio, University of Salerno, Italy
 F. Colace, University of Salerno, Italy
 Gennaro Costagliola, Univ of Salerno, Italy
 Tiansi Dong, Bonn-Aachen International Center for Information Technology, Germany
 Martin Erwig, Oregon State University, USA
 Daniela Fogli, Università degli Studi di Brescia, Italy
 Kaori Fujinami, Tokyo University of Agriculture and Technology, Japan
 David Fuschi, Brunel University, UK
 Ombretta Gaggi, Univ. of Padova, Italy
 Angela Guercio, Kent State University, USA
 Carlos A. Iglesias, Intelligent Systems Group, Spain
 Erland Jungert, Linköping University, Sweden
 Yau-Hwang Kuo, National Cheng Kung University, Taiwan
 Fuhua Lin, Athabasca University, Canada
 Alan Liu, National Chung Cheng University, 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
 Marcus Winter, University of Brighton, UK
 Weiwei Xing, Beijing Jiao Tung University, China
 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
Pedro Isaias, University of Queensland, Australia
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
Veronica Rossano, University of Bari, Italy
Lidia Stanganelli, Freelance Software Engineer, Italy

Publicity Chair and Co-Chair

Jun Kong, North Dakota State University, USA; Publicity Chair
Flora Amato, Univ. of Salerno, Italy; Publicity Co-Chair
Marcus Winter, University of Brighton, UK; Publicity Co-Chair

Keynote

A Mathematical Language for the Modeling of Geospatial Static Rules

Professor Robert Laurini
Professor Emeritus
University of Lyon
Lyon, France

Abstract

More and more in information technologies, rules are considered as first-class citizens; and many applications in business intelligence are built on rules. But in territorial intelligence and in smart city planning, few works have been done in this direction. The role of this paper is to show the importance of geographic rules and to propose, beyond the modeling in natural language, a mathematical language to model them. This language is primary based on logic and set theory in which some relations and operators coming from topology, computational geometry and operation research are integrated, and will cover only static rules. By mathematical language, one means that it is independent from any software products and applications, and its formal grammar is presented by means of syntactic diagrams. Examples in urban planning are provided.

About the Speaker

Professor Robert Laurini holds two doctorates (1973 and 1980) from the University of Lyon, France in which he was professor in information technologies at INSA-Lyon. He is a specialist in geographic information systems and more recently on knowledge engineering for smart cities. In 76-77, he was researcher at the Martin Centre, University of Cambridge, UK; in 86-87, visiting professor at the University of Maryland, College Park; during the period 1990-2000, he was part-time professor at the IUAV University of Venice, Italy. He has supervised or co-supervised 45 PhD's. He was involved in PhD committees in 17 countries. He has authored or co-authored more than 250 papers and 8 books, the last of them is "Geographic Knowledge Infrastructure, Applications to Territorial Intelligence and Smart Cities". He was one of the founders of the ACM Spatial Group, European editor of "Computers, Environment and Urban Systems" and associated editor of the "Journal for Visual Languages and Computing". He is now fellow of the Knowledge Systems Institute, USA. He fluently speaks English, French, Italian and Spanish. He is also the founder of "Academics Without Borders", which is an NGO devoted to modernizing universities in developing countries. For more details please see: <http://www.laurini.net/robert/>

Table of Contents

Keynote

A Mathematical Language for the Modeling of Geospatial Static Rules	1
<i>Robert Laurini</i>	

Session I: Models and Graphs

Metaphorical Parametric Model for Brand Mark Design: Towards a universal model of computational visual communication design	11
<i>Po Choy Ng and Clifford Sze-Tsan Choy</i>	
Approach to Context Computation for Implicit Context-sensitive Graph Grammars.....	21
<i>Yang Zou, Xiaoqin Zeng, Yufeng Liu and Huiyi Liu</i>	
Multilevel Topic Dependency Models for Assessment Design and Delivery: A Hypergraph Based Approach	30
<i>Kendra Cooper and Hassan Khosravi</i>	

Session II: Sketching and Interactive Systems

Crosside: A Cross-Surface Collaboration by Sketching Design Space	40
<i>Jorge-Luis Pérez-Medina, Jean Vanderdonckt and Santiago Villarreal-Narvaez</i>	
C-QWERTY: a text entry method for circular smartwatches (S)	51
<i>Gennaro Costagliola, Mattia De Rosa, Raffaele D'Arco, Sabato De Gregorio, Vittorio Fuccella and Daniele Lupo</i>	
Vector-based, Structure Preserving Stroke Gesture Recognition (P)	58
<i>Nathan Magrofuoco, Paolo Roselli, Jorge Luis Perez-Medina, Jean Vanderdonckt and Santiago Villarreal-Narvaez</i>	

Session III: Software and Metrics

A Systematic Mapping on Visual Solutions to Support the Comprehension of Software Architecture Evolution	63
<i>Joao Werther Filho, Glaucio Carneiro and Rita Maciel</i>	
Design of a Dashboard of Software Metrics for Adaptable, Energy Efficient Applications (S)	75
<i>Vladimir Ivanov, Daria Larionova, Dragos Strugar and Giancarlo Succi</i>	
An Empirical Analysis on Effectiveness of Source Code Metrics for Aging Related Bug Prediction	83
<i>Chinmay Hota, Lov Kumar and Lalita Bhanu Murthy Neti</i>	
A new design model of the aesthetic data visualization (P)	92
<i>Qi Li and Chuanhong Xu</i>	

Session IV: Systems and Applications

When E-Learning Meets Big Data, Cognitive Computing, and Collaborative Environments	95
<i>Mauro Coccoli, Paolo Maresca and Andrea Molinari</i>	
Event Detection in Complex Environments: An Effective and Efficient Machine-Learning-Based Framework (S)	103
<i>Alfredo Cuzzocrea and Enzo Mumolo</i>	
Exploiting Coordinated Views for Scholarly Reading and Analysis	113
<i>Francesco Poggi, Paolo Ciancarini, Angelo Di Iorio, Silvio Peroni and Fabio Vitali</i>	
Graphical Animations of the Suzuki-Kasami Distributed Mutual Exclusion Protocol	125
<i>Dang Duy Bui and Kazuhiro Ogata</i>	
A study on the interaction of Environmental Information in the Protection of Shanghai Guangfulin Cultural Heritage (P)	135
<i>Chuanghong Xu and Mengying Kong</i>	

Note: (S) indicates a short paper and (P) indicates a poster paper.

A Mathematical Language for the Modeling of Geospatial Static Rules

Robert Laurini

*Knowledge Systems Institute, USA,
and University of Lyon, France)*

Roberto.Laurini@gmail.com; ORCID 0000-0003-0426-4030

Abstract—More and more in information technologies, rules are considered as first-class citizens; and many applications in business intelligence are built on rules. But in territorial intelligence and in smart city planning, few works have been done in this direction. The role of this paper is to show the importance of geographic rules and to propose, beyond the modeling in natural language, a mathematical language to model them. This language is primary based on logic and set theory in which some relations and operators coming from topology, computational geometry and operation research are integrated, and will cover only static rules. By mathematical language, one means that it is independent from any software products and applications, and its formal grammar is presented by means of syntactic diagrams. Examples in urban planning are provided.

Keywords—geographic knowledge; geographic rules; knowledge engineering; formal grammar; territorial intelligence; smart cities.

I. INTRODUCTION

According to Graham [6] and Morgan [14], rules must be considered as first-class citizens in information technologies, meaning first that several computer-based activities must be revisited. By definition, a rule is a sequence antecedents-implication-consequents which can be noted by $(A) \Rightarrow (B)$, in which A is a conjunction of conditions. Instead of antecedents, sometimes the expression ‘premise’ is used. In logic, B can be either a disjunction of conditions or a set of assertions. They come from the so-called Horn clauses which are also the basis of logic programming, where it is common to write definite clauses in the form of an implication: $(p \wedge q \wedge \dots \wedge t) \Rightarrow u$.

However, in urban planning, rules are essentially coming from laws and by-laws. Moreover, some experts can use other rules in their daily practice, sometimes called best practices. In addition to that more and more specialists in spatial data mining can discover what they call associative rules.

In the knowledge society, in territorial intelligence and in smart city management, it is important not only to identify rules, but also to combine them to automate reasoning. Facing this objective, the scope of this paper is to propose a language in order to model rules.

Indeed, rules are often made explicit in natural language (i.e. English, French, Spanish, etc.); two main sources will be used, namely rules written in natural language and associative rules as extracted from data mining. This language will be based on mathematics without taking into account practical implementation. In other words, this is not a computer language (code).

This paper is organized as follows. First, some elements will be given to explain the role of rules in computing, and

especially in geoprocessing. Then the formal grammar of this mathematical language will be detailed. Finally, examples especially in urban planning, will clarify the expressive power of this language.

Voluntarily, this paper will not deal with 3D issues, neither with temporal issues: only geographic static rules will be considered.

II. ABOUT RULES IN IT

In this section, the importance of rules will be emphasized in business intelligence and then in territorial intelligence.

In Business intelligence, generally, their implementation is based on two grammatical structures IF-THEN-Fact and IF-THEN-Action (Ross [16]). The first serves above all to involve new facts, that is new objects, attribute values, new relationships between objects. As to the second, it is to involve new actions. But who will be in charge of such new actions? In some cases, the computer itself may run procedures or send a message to other devices; in others, particularly in regulatory contexts, a decision maker (for example, the CEO of a company) must himself initiate the action. Another interpretation could be that the choice of alternatives of an action, for example when a law, in some well-defined contexts, opens many perspectives.

Thus, a rule is a basic element of a strategy to build reasoning. In contrast to algorithms, they are expressed declaratively. Among business rules, Dietz [3] distinguishes between three categories:

- rejectors typically those related to quality control, that allow a rejection (rejection rules),
- producers such as those determining new values (ex VAT calculation); they can be considered as rules of production of information,
- and projectors such as those related to the replenishment of stocks.

To conclude this section, let us mention that lots of business applications are based on rules, and several computer languages for encoding business rules have been proposed, as XML extensions, such as SWRL (Espinasse, [5]) or RuleML (Boley [1], Boley et al. [2]). The simplest of those extensions is as follows:

```
<Implies>
  <if>
<..>
  </if>
  <then>
  <..>
  </then>
</Implies>
```


Also, very recently, a visual language has been proposed such as by Pittl et al. [15], based on SWRL.

III. RULES IN GEOPROCESSING

In contrast, concerning geographic sciences and their applications, few works have been done in the spirit of knowledge engineering. However, let us mention Malerba et al. [13], Salieb-Aouissi et al. [18] and Varadharajulu et al. [22].

From spatial data mining, for instance, Malerba et al. (2003) have discovered the following rule:

$$\begin{aligned} is_a(X, large_town) \wedge intersects(X, Y) \wedge is_a(Y, road) \\ \rightarrow \\ Intersects(X, Z) \wedge is_a(Z, road) \wedge Z \neq Y (91\%, 85\%). \end{aligned}$$

Which states that ‘If a large town X intersects a road Y, then X intersects a road Z distinct from Y with 91% support and 85% confidence’. In other words, one can say that *is_a* allows a kind of definition, *intersects* a relation and $\neq$ an additional condition.

Salieb-Aouissi et al. [18] have studied geology and mineral deposits in South America. They extended the existential and universal quantifiers ($\forall$, $\exists$) by incorporating buffer zones (here 5 km), and have proposed rules such as:

$$\begin{aligned} Mines - \exists_{5km}^3 Faults \rightarrow True \\ Mines - \exists_{1km}^1 Volcano \rightarrow (active=yes) \end{aligned}$$

The first rule means that there exist at least 3 faults within 5 km of a target object (mineral deposits), whereas the second states that there exists at least one active volcano within 1 km of a target object. They also propose to write $\forall_{5km}^{80\%}$ for considering 80% of items with 5 km. Remark that the distance is taken as a modification of the quantifiers. As it could be nice for colocation rules, other extensions must be defined for other relations coming from topology or computational geometry.

However, Varadharajulu et al. have proposed the following rule for checking road length against road type by using SWRL (remark that the length is given as a data, not computed from computational geometry); more exactly, a road less than 200 m long must be called a "close":

$$\begin{aligned} NEWROAD(?R1), ROADSUFFIX(?R1, ?T1), \\ hasLength(?R1, ?200), SameAs(?T1, ?Close) \\ \rightarrow isAllowed(?R1, true) \end{aligned}$$

All those three examples illustrate the difficulties to incorporate issues coming from topology and computational geometry. Indeed, the governance of smart cities must be based on both artificial intelligence and collaborative human intelligence. New domains such as sensors networks, big data, deep learning, geovisualization, etc. must be mobilized together with knowledge engineering to provide more efficient systems for citizens. Do not forget that one of the scopes of urban big data is to discover novel patterns and rules.

A. About geographic rules

In Laurini et al. [10], several examples of geographic rules were given, and some important semantical aspects were extracted. Anyhow, geographic rules are a way to model what Shoorcheh [21] has called spatial causality.

B. Generalities about the model

Based on the previous descriptions, a general diagram for rules can be designed regrouping all aspects, their origin, components, temporal dimension, mathematic tools, their management, their usage and the various modes of implication (Figure 1).

- For physical rules, the implication is mandatory;
- For legal rules, the implication is also mandatory, but the sanctions may or may not exist; in this case, a second rule with the negative conditions will lead to the sanctions;
- For best practices, the implication is more or less a kind of recommendation; in other words, nobody is obliged to follow this kind of rules; perhaps some additional conditions could be considered; at a first approximation, random variates can help to select best practices if any.
- For rules coming from data mining, it is necessary to provide support and confidence. See Shekar et al. [20] for details.

Figure 1 explains the main characteristics of geospatial rules, namely, origin, mathematical tools, temporal dimension, semantics, modality, usage and management. Remark that an existing rule can be superseded by another novel rule, sometimes for a period or a narrower place.

Fig. 1. Main characteristics of geospatial rules.

Now that basic components of geospatial rules are given, let us try to design a mathematical model.

IV. A MATHEMATICAL MODEL

This model will be based not only on logics and set theory, but also on computational geometry, topology and fuzzy set theory. Before giving the grammar, some precisions must be done to explain the main components. Anyway, in this model, we assume that:

- a) a global geographic object called *Earth* includes all existing geographic objects (*GO*) and territories (*Terr*),
- b) an ontology will describe their types/classes (whatever is the concept) together with some specific attributes and generic relations between them,
- c) all objects will have valued attributes; in the case of new objects, the attributes are set to null,

- d) there exists a set named *Projects* which comprises all possible environmental and urban projects,
- e) there exists a gazetteer integrating all place names possibly with different variants and in different languages,
- f) it is assumed that all information is correct and consistent,
- g) it is assumed that there are no problems neither regarding geometry accuracy, nor multi-representation,
- h) there are no considerations for storing, implementation, optimization, etc.,
- i) this language is not a rigid language as in computer sciences, but a sketch in which everybody can add symbols, functions, etc.

A. Geographic sets and collections

The first elements are the *Earth* and the sets of Geographic Objects (*GO*). As some of them are well characterized (roads, buildings, islands, etc.), others need some clarifications about their definitions. The scope of this paper is not to contribute to those characterizations as works regarding geographic ontologies have revealed several difficulties of categorization. Smith and Varzi [19] have tried to clarify by distinguishing “Fiat” and “Bona Fide” boundaries. Many objects are known by their name (such Sahara), but the boundaries are not well defined, similarly for the Rocky Mountains.

B. About places

All places belong to *Earth* and can be described according to various solutions. The simplest ones are by a name, for instance Argentina, the second by a set of coordinates (polygons). Do not forget that a place can form a non-connected polygon (for instance, a country with its islands). More complex solutions can be defined, e.g. the group of countries in which people are driving on the left, a city district enclosed in a set of streets.

C. About geographic objects

All geographic objects (*GO*) have three types of attributes, for identifying them, for describing them from a geometric point of view and the semantic attributes. The dot notation can be used, for instance “*A.population*”.

Concerning identification, sometimes ID’s can be used, but it is the more common to use a place name or a toponym. As several places can have the same toponym (e.g. Mississippi), the location will resolve ambiguities. Concerning geometry, remember that sometimes an object can have several geometries (multi-representation) sometimes taken at different scales.

Semantic attributes can have different formats such as alphanumeric, Boolean, multimedia, etc. In some cases, fuzzy values can be used such as “near”, “far”, “low”, “high”, etc. See Kainz [7] for details and examples.

However, in urban and environmental planning, one needs to consider projects. Indeed, projects have different phases. First, they are designed (design phase) maybe with several sub-steps (preliminary, front-end, etc.). Then there is a legal phase (building permit) in which the project can be approved or rejected. The construction phase itself will followed, sometimes delayed for archaeological or financial reasons. Finally, the projected object becomes really a geographic

feature, finished or not. In other words, it is necessary to consider a *Project* superclass including any geographic object-to be, for instance *Project.Road*. To simplify the problem, the ontology of those objects-to-be will similar to the ontology of existing geographic objects.

D. Relations, functions and procedures

Geographic relations are the basis of geographic reasoning. Those relations include Egenhofer et al. [4] 2D topological relations (Figure 2) and other relations. Some of them are defined for types, and some other for specific geographic objects. Remember that a relation is Boolean (true or false) and can be checked in conditions.

Fig. 2. 2D Topological relations. Egenhofer et al. [4].

Functions are also useful, some of them coming from computational geometry. To name a few, *Union*, *Intersection*, *Minus*, *Distance*, *Centroid*, *Buffer*, etc. They will be invoked when necessary. Anyone can add additional functions.

Two functions will be extensively used, *Geom* for the geometry and *Topo* for names. By definition *Geom* (*A*) is equivalent to *A.Geom*, and *Topo* (*A*) to *A.Toponym*. Although they are not indispensable, they will be useful in some cases.

Moreover, regarding existing models, the best solution is to encapsulate those mathematical models into procedures for using them when necessary. In addition, a special procedure can be used to call another rule.

E. Semantics of Symbols

In this model, the mainstream meaning of mathematical symbol will be accepted. However, due to this special context, some semantics must be fixed for some symbols. Let us explain some problems.

Implications

Three types of implications must be considered. The symbol $\Rightarrow$ will be used for logical assertions coming from physics or legal regulations. For best practices, the expression " $\Rightarrow$ [BP]" will be used, and for associative rules coming from data mining, this symbol will be accompanied with the support and the confidence levels.

About equalities

Remember that the very common sign $=$ has three meanings, (i) for defining something, (ii) for an assignment of value, and (iii) for comparisons. We will use the following symbols:

- a) $=$ as a comparator in Boolean conditions; so that the answer will be either true or false,
- b) $\equiv$ for definitions, especially of new variables,
- c) $:=$ for assigning a new value to a known variable.

The More, the Merrier

Often rules are written according to the style "the more of this, the more of that", or "the less of this, the more of that"; for instance: "the more of traffic, the more of pollution". To solve this problem, three solutions are possible.

- To compare two geographic points – 1 and 2 – and to show the evolution in space of some attributes such as if $A_1 > A_2$ then $F(A_1) > F(A_2)$, or $G(A_1) < G(A_2)$,
- A solution can be to write $+x \Rightarrow -y$, the semantics of which are when x increases, then y diminishes; for instance, when saying, the higher, the colder, this sentence can be roughly encoded by $+elevation \Rightarrow -temperature$,
- If a function between both variables is known, it can be used directly.

About homologies

When one needs to compare two geometric shapes, it is commonly accepted that there are some point coordinates which are slightly different. Moreover, for the same object, one person has measured 100 points, whereas another 500. Sometimes, the symbol $\approx$ is used for comparing numerical values. But when we need to compare types and toponyms of geographic objects, other semantics must be ascertained. Indeed, for strings of characters, the Levenshtein distance [12] is used. For instance, between Iraq and Iran the Levenshtein distance is 1, whereas they are very different geographic objects. Regarding types, one can consider that the concepts *street* and *road* are similar.

To solve this problem, I have proposed to use the symbol $\bowtie$ (Laurini, [9, 11]) for homologies. So that, considering two geographic objects, we can write:

- $Geom(A) \bowtie Geom(B)$ for comparing these geometric shapes (hence this is equivalent to $Geom(A) \approx Geom(B)$).
- "London" $\bowtie$ "Londres" (London, UK, is called Londres, both in French and Spanish languages; another example is "Washington D.C." $\bowtie$ "District of Columbia").
- $Type(A) \bowtie Type(B)$ when the concept names are different but are corresponding to the same concept, maybe coming from different ontologies.

Other symbols

In logic, the entailment symbol $\models$ is used to denote an affirmation taking the context into account. By writing $\models Contains(A, B)$, one claims that this relationship must be considered as True. Whereas by writing only $Contains(A, B)$, one declares that he is considering this relationship, but he has no hints whether it is True or False.

Regarding the other symbols coming from the set theory ($\forall, \exists, \cap, \cup, \supset, \subset, \in, \vee, \wedge, \oplus$, etc.), they have their common meaning.

Consequents

In logic, remember that a conjunctive list of antecedents implies a disjunctive list of consequents so that $A_1 \wedge A_2 \wedge A_3 \wedge \dots A_i \Rightarrow C_1 \vee C_2 \vee C_3 \vee \dots C_j$ in which both A and C are Boolean expressions. In the disjunctive list, the semantics are not clear: are all C 's true or only a subset?

Moreover, in our case, the consequents are not always Boolean (remember IF-THEN-Fact and IF-THEN-Action). To solve this problem, several decisions must be made: (i) discarding the symbol $\vee$, and separating consequents by the symbol $;$, meaning that all of them are independent and must be enabled; (ii) to use set bracket parentheses $\{\}$ to delimit several consequents if any.

About complex places

Two points of view can now be considered, according to the set theory and according to topology: we can write either $Ghana \in Earth$ or $Contains(Earth, Ghana)$. By extension, the set of left-driving countries can be defined by either $(UK, Ireland, Japan, \text{ etc.}) \in LD_Countries$, or $Geom(LD_Countries) \equiv Union(Geom(UK), Geom(Ireland), Geom(Japan), \text{ etc.})$. More generally, to define a place, all computational geometry functions can be used. Concerning streets, it could be important to delimit only a sector, for instance by giving civic numbers; for instance, $Geom(S) \equiv StreetSector(Street_name, Civic_number1, Civic_number2)$.

To delimit a city district by a set of surrounding streets, a special function can be invoked *SurroundedByStreet* (*Street1, Street2, Street3, Street4*, etc.).

Remarks

Here are some additional remarks. First, we need to split antecedents in context and conditions. Indeed, in the previous examples, several things were not explicit, i.e. the considered objects and sets were hinted or hidden: those aspects will be described in the context, maybe including definitions and constraints. Moreover, if one needs to consider a negative rule, only the conditions are denied, but not the constraints.

Secondly, let us enlighten the differences between $a \in b$ and $contains(c, d)$. In those expressions, b is a set, and a belongs to this set in the sense of the set theory, whereas c and d are geometric objects and d is inside c from a topological point of view.

V. FORMALISM

Any geographic rule will be designed as pictured in Figure 3, containing namely Context, Conditions, Implication and Consequents with the following separators : and $\blacksquare$. Context will describe the main variables; Conditions, one or more Boolean criteria to follow; Implication, the modality; and Consequents, all possible consequences that are to be run.

The grammar is presented by means of the Railroad Diagram Generator (Rademacher, [17]) which is an interesting way to present formally the syntactic structure of a language.

Fig. 3. Formal definition of the grammar.

The Context will be defined as usual from the set theory (Figure 4) by defining variables and their sets, possibly followed by Definitions.

Fig. 4. Context and Declarations.

In Figure 4, as previously explained the extended quantifiers (Salleb-Aoussi et al. [18]) will be used in which a stands either for void (i.e. 1) or for a number greater than 1, whereas b is a percentage. By definition, $\exists^1 = \exists$ and $\forall^{100\%} = \forall$.

Those Definitions (Figure 4) will be used for giving additional statements in the following manner (Figure 5) in which two kinds of statements will be accepted, either the assignment of a value to a variable already defined or the result of an algebraic expression. In this paper, the notion of algebraic expression will not be defined since it is very common in all mathematical languages. In addition, in the Definitions, it is possible to state a relationship between variables considered as a sort of constraint.

Fig. 5. Definitions and Definition.

Now that the Context is fully characterized, let's consider Conditions. Similarly, Conditions and Condition will be defined in Figure 6.

Regarding implications, as previously told, three modalities exist (Figure 7) in which BP means best practices. As usual, Support and Confidence are percentages for rules coming from data mining.

Now, regarding Consequents, the structure will be similar but recursive (Figure 8).

Fig. 6. Conditions and Condition.

Fig. 7. Implication with three modalities.

Fig. 8. Consequents and Consequent.

Now that the grammar is fully characterized, let us consider a few examples.

VI. EXAMPLES

Here are given a few examples of rules written with the above-mentioned language. For that, we suppose the existence of a town named "Smart Town".

A. Creating a new relation, a new zone

For defining a new relationship, it can be easily done, for instance for *North* (see Rule 1). An equivalent could be written for *South*. This rule is valid anywhere in the *Earth*.

$ \begin{aligned} &\forall p_1, p_2 \in \text{Earth}, \text{GeomType}(p_1) \equiv \text{Point}, \\ &\quad \text{GeomType}(p_2) \equiv \text{Point} \\ &\quad : \\ &\quad \text{Latitude}(p_1) > \text{Latitude}(p_2) \\ &\quad \Rightarrow \\ &\quad \models \text{North}(p_1, p_2) \blacksquare \end{aligned} $	Rule 1
--	--------

For defining *East* and *West*, the rule is a little more complex (Rule 2).

$ \begin{aligned} &\forall p_1, p_2 \in \text{Earth}, \\ &\quad \text{GeomType}(p_1) \equiv \text{Point}, \text{GeomType}(p_2) \equiv \text{Point} \\ &\quad : \\ &\quad (\text{Longitude}(p_1) > \text{Longitude}(p_2)) \\ &\quad \wedge (\text{Longitude}(p_1) - \text{Longitude}(p_2) < 180^\circ) \\ &\quad \Rightarrow \\ &\quad \models \text{West}(p_1, p_2) \blacksquare \end{aligned} $	Rule 2
--	--------

Suppose we want to create a new relation when a road is crossing a river. We need to consider a road, a river and their intersection. If there is an intersection between the road and the river, there is a *cross* relationship between them, and it is commutative (Rule 3).

$ \begin{aligned} &\forall Ro \in \text{Road}, \forall Ri \in \text{River} \\ &\quad : \\ &\quad \text{Area}(\text{Intersection}(\text{Geom}(Ro), \text{Geom}(Ri))) \neq 0 \\ &\quad \Rightarrow \\ &\quad \{\models \text{Cross}(Ro, Ri); \models \text{Cross}(Ri, Ro)\} \blacksquare \end{aligned} $	Rule 3
--	--------

For the creation of a flood-risk area in Smart Town, let's supposed it to be defined within a 100 m buffer. Beware, the river can pass either through the town (*Overlap*) or be at the border (*Covers*). Here we must use a constructor, that is a procedure to create a novel geographic object (*CreateGO*). See Rule 4).

$ \begin{aligned} &\exists T \in \text{Town}, \forall R \in \text{River}, \text{Topo}(T) \equiv \text{"Smart Town"} \\ &\quad : \\ &\quad \text{Overlap}(R, T) \vee \text{Covers}(T, R) \\ &\quad \Rightarrow \\ &\quad \{\text{CreateGO}(F); \text{Type}(F) := \text{"Floodplain"}; \\ &\quad \quad \text{Geom}(F) := \text{Intersection}(\text{Geom}(T), \text{Buffer}(R, \\ &\quad \quad 100))\} \blacksquare \end{aligned} $	Rule 4
--	--------

Suppose that a law decides that when a brownfield site is depolluted, it is possible to accept a new building (Rule 5).

$ \begin{aligned} &\forall P \in \text{Parcel}, L \in \text{Plants}, B \in \text{Project.Building}, \\ &\quad \models \text{contains}(P, L), \models \text{contains}(P, B) \\ &\quad : \\ &\quad L.\text{Erased} \wedge P.\text{Depolluted} \\ &\quad \Rightarrow \\ &\quad \models B.\text{Authorized} \blacksquare \end{aligned} $	Rule 5
--	--------

Suppose now that it was decided not to build in a marsh: this rule can be written as follows (Rule 6):

$ \begin{aligned} &\exists C \in \text{County}, M \in \text{Marsh}, \forall B \in \text{Project}, \\ &\quad \models \text{Contains}(C, M) \\ &\quad : \\ &\quad \text{Contains}(M, B) \\ &\quad \Rightarrow \\ &\quad \text{Prohibit}(B) \blacksquare \end{aligned} $	Rule 6
---	--------

B. Again the More, the Merrier

Suppose that "Smart Beach" is a sea resort in which land prices diminish when the distance to the sea increases. This economic rule (Rule 7) can be written as follows when considering two parcels with the same area:

$ \begin{aligned} &\forall p_1, p_2 \in \text{Parcel}, \exists S \in \text{Sea}, C \in \text{County}, \\ &\quad \models \text{Touches}(Sea, C), \\ &\quad \models (\text{Area}(\text{Geom}(p_1)) \approx \text{Area}(\text{Geom}(p_2))) \\ &\quad : \\ &\quad \text{Contains}(C, p_1) \wedge \text{Contains}(C, p_2) \\ &\quad \wedge (\text{distance}(\text{centroid}(\text{geom}(p_1)), S) > \text{distance}(\text{centroid}(\text{geom}(p_2)), S)) \\ &\quad \Rightarrow \\ &\quad \models p_1.\text{Landprice} < p_2.\text{Landprice} \blacksquare \end{aligned} $	Rule 7
--	--------

With the second solution (Rule 8), we get:

$ \begin{aligned} &\forall p \in \text{Parcel}, S \in \text{Sea}, C \in \text{County}, \\ &\quad \models \text{Touches}(Sea, C) \\ &\quad : \\ &\quad \text{Contains}(C, p) \wedge ++\text{distance}(\text{centroid}(\text{Geom}(p)), S) \\ &\quad \Rightarrow \\ &\quad \models --p.\text{Landprice} \blacksquare \end{aligned} $	Rule 8
--	--------

Suppose that some economists have evaluated a function *F* for prices, provided that the area is more than 100 m² and the distance to the sea less than 10 km, the following can be written (Rule 9):

$ \begin{aligned} &\forall p \in \text{Parcel}, \exists S \in \text{Sea}, C \in \text{County}, \\ &\quad \models \text{Touches}(Sea, C), \\ &\quad d \equiv (\text{Centroid}(\text{Geom}(p)), S) \\ &\quad : \\ &\quad \text{Contains}(C, p) \\ &\quad \wedge \text{Area}(\text{Geom}(p)) < 100 \\ &\quad \wedge d < 10 \text{ km} \\ &\quad \Rightarrow \\ &\quad p.\text{Landprice} := F(\text{Area}(\text{geom}(p)), d) \blacksquare \end{aligned} $	Rule 9
---	--------

C. Rules with fuzzy attributes

In the previous case, another solution can be based on fuzzy values: the previous rule can be transformed into "near the sea, prices are high", and "far from the sea, prices are low" (Rule 10).

$ \begin{aligned} &\forall p \in \text{Parcel}, \exists S \in \text{Sea}, C \in \text{County}, \\ &\quad \models \text{Touches}(Sea, C), \\ &\quad d \equiv (\text{Centroid}(\text{Geom}(p)), S) \\ &\quad : \\ &\quad \text{Dist}(S, p) = \text{"near"} \\ &\quad \Rightarrow \\ &\quad p.\text{Landprice} := \text{"high"} \blacksquare \end{aligned} $	Rule 10
---	---------

An additional rule can be written, by respectively replacing "near" and "high" with "far" and "low".

D. Best practice rules

Suppose it is a best practice to assign a pollution sensor to each lamppost located in "Churchill Road"; the example is given in Rule 11.

$\exists T \in Town, \exists R \in Road, \forall L \in Lamppost,$ $\forall S \in Pollution.Sensor,$ $Topo(T) \equiv "Smart\ Town",$ $Topo(R) \equiv "Churchill\ Road"$ $:$ $Contains(Geom(T), Geom(R)) \wedge Contains(Geom(R),$ $Geom(L))$ $\Rightarrow [BP]$ $Assign(S, L) \blacksquare$	Rule 11
--	---------

E. Associative spatial rules

When studying road incidents in the city of Helsinki, Finland, Karasova et al. [8] have shown by spatial data mining that many incidents occur near bars and restaurants. More exactly, around each incident they have designed a 50 m buffer and see whether there were incidents in those zones (Support 1.7% and confidence 40.0%).

The fact that the support is weak means overall that in their database, there are many other objects, whereas the confidence level means that 40 % of incidents happen in the vicinity of bars or restaurant. This associative rule can be written as follows in which *Terr* is a territory (Rule 12):

$\exists C \in City, \forall B \in Bar, \forall R \in Restaurant,$ $\forall I \in Incident,$ $\exists RiskyZone \in Terr, Topo(C) \equiv "Helsinki",$ $\models Contains(Geom(C), Geom(B)),$ $\models Contains(Geom(C), Geom(R)),$ $\models Contains(Geom(C), Geom(I)),$ $Geom(RiskyZone) \equiv$ $Union(Buffer(Centroid(Geom(B), 50), Buffer$ $(Centroid(Geom(R), 50))$ $\Rightarrow [1.7\%, 40.0\%]$ $\models Contains(Geom(RiskyZone), Geom(I)) \blacksquare$	Rule 12
---	---------

In their study in the city of Antwerp, Belgium, Zhou et al. [23] have a lot of co-location association rules within 600 m buffers.

For instance, they discovered such a rule between kindergartens, playgrounds, but support and confidence were not mentioned (let write α and β for those unknown values in the Rule 13).

$\exists C \in City, \forall K \in Kindergarten, \exists P \in Playground,$ $Topo(C) \equiv "Antwerp"$ $:$ $Contains(Geom(C), Geom(K)),$ $Contains(Geom(C), Geom(P))$ $\Rightarrow [\alpha, \beta]$ $\models Contains(Geom(Centroid(Geom(K), 600), Geom$ $(Centroid(P))) \blacksquare$	Rule 13
---	---------

F. Back to previous rules

Let us now rewrite rules given in §III.

$\exists P \in Earth, \forall X \in LargeTown, \forall Y, Z \in Road,$ $Topo(P) \equiv "Italy"$ $:$ $Intersection(X, Y) \wedge Z \neq Y$ $\Rightarrow [45\%, 90\%]$ $\models Intersection(X, Z) \blacksquare$	Rule 14
--	---------

The example previously given from Malerba et al. [13] from an Italian census can be modeled by (Rule 14):

The first rule given by Assab et al. [18] ($Mines - \exists_{5k}^3 Faults \rightarrow True$) can be translated into (Rule 15):

$\exists SA \in Earth, \exists^3 F \in Fault,$ $Topo(SA) \equiv "South\ America"$ $:$ $Contains(Geom(SA), Geom(F))$ $\Rightarrow$ $\{\exists M \in Mines;$ $\models Contains(Geom(Buffer(Centroid(M), 5000),$ $Geom(F))\} \blacksquare$	Rule 15
--	---------

To finish this section, let's consider the ultimate rule designed by Varadharajulu et al. [22] (Rule 16):

$\forall R \in Project.Road$ $:$ $R.Length(R) < 200, Type(R) = "Close"$ $\Rightarrow$ $\models R.Allowed \blacksquare$	Rule 16
--	---------

H. Located rules

Sometimes rules are located in small places, for example for urban planning zones. Figure 9 illustrates the case in which there are 3 zones in which 5 rules can be applied, those zones being delimited by the list of surrounding streets. Two solutions are possible:

- 1 – for each zone, give the list of applicable rules,
- 2 – for each rule, give the list of zones in which they apply.

Fig. 9. Rules and zones.

With this rule-oriented language, let's detail take the option that, for each zone, will give the rules to be applied. For zone A, the following can be written (Rule 17).

$\begin{aligned} &\exists C \in \text{City}, \forall B \in \text{Project}, \exists \text{ZoneA} \in \text{Terr}, \\ &\text{Geom}(\text{ZoneA}) \equiv \text{SurroundedByStreet}(\text{A_Street}, \\ &\quad \text{B_Street}, \text{D_Street}, \text{F_Street}) \\ &\quad \vdots \\ &\text{Contains}(\text{Geom}(\text{ZoneA}), \text{Geom}(B)) \\ &\Rightarrow \\ &\{\text{AppliedRule}(101); \text{AppliedRule}(102)\} \blacksquare \end{aligned}$	Rule 17
--	---------

And similar rules can be written for *ZoneB* and *ZoneC*.

I. Superseded rules

As given in Figure 1, geographic rules can be superseded. But this expression has two meanings. Let us explain them.

The first means that a rule can be replaced by another rule or a set of new rules. For instance, in urban planning when a new plan is adopted. In this case, the previous rule is removed and a new one is enabled. However, geographic objects generated by means of the previous rule still continue to stay: suppose the previous rule imposes a limitation of 20 m for buildings and the new one only 15 m; it does not imply to demolish those 20m-high existing buildings. Let's call this, temporal superseding.

The second meaning is more complex: suppose that a new rule is now valid in a small portion of a territory as illustrated in Figure 10. Two solutions are possible; either to state that the new rule (in Z_2) has a priority over the rule in Z_1 ; or to modify the initial rule by replacing *Geom*(Z_1) by *Minus*(*Geom*(Z_1), *Geom*(Z_2)). Let's call this, spatial superseding.

Fig. 10. Example of spatial superseding.

VII. EXAMPLE IN URBAN PLANNING

Building permit delivery is a very complex task for local authorities. Indeed, a projected building must be compliant with two types of constraints, national laws and local rules. National laws which are valid everywhere in the country, generally consider structural aspects, such as electricity, insulation, access for handicapped people, etc. – issues which are outside the goal of this paper – whereas local rules consider the location of the building and its conformance to local planning rules, such as height, distance to neighboring parcels, floorspace ratios, etc.

TABLE I. PLANNING ZONES AND THEIR PARAMETERS

Zone ID	Max Height (in m)	Max Floorspace ratio	Max Footprint
Downtown	12	3	80 %
Suburban area	15	4	70 %
Rural area	12	0.5	30 %
Near airport (Bowtie)	8	2	50 %
Airstrip	0	0	0

Figure 11 depicts the map with the extension of planning zones, and Table I gives some limit parameters like maximum height, maximum floorspace ratio and maximum footprint of the building.

Fig. 11. Planning zones

A. Rules

$\begin{aligned} &\exists C \in \text{City}, \exists PZoneA \in \text{Project.Terr}, \\ &\text{Topo}(C) \equiv \text{"Smart Town"} \\ &\text{Geom}(PZoneA) \equiv \text{Polyg}(731, 128; 903, 133; 905, \\ &\quad 341; 839, 346; 814, 349) \\ &\quad \vdots \\ &\text{Approved}(PZone) \\ &\Rightarrow \\ &\text{AffectName}(PZoneA, \text{"Downtown"}) \blacksquare \end{aligned}$	Rule 18
---	---------

First let's suppose we have rules for dividing the territory and creating zones.

And so, for *ZoneB*, "Suburban area"; *ZoneC*, "Rural area"; *ZoneD*, "Near airport" and *ZoneE*, "Airstrip", with the appropriate coordinates, respectively numbered Rule 19, Rule 20 and Rule 21. For a projected building supposedly located in the Suburban Area, the rule will be as follows to be accepted.

In this toy-example, we will consider a projected building presented to "Smart Town" local authority, the urban planning department of which has decided to cut the town into three zones, namely downtown, suburban area and rural area.

B. Rules regarding parcels and construction

To simplify, we will not consider parcels in which construction is not possible, and parcels for which municipalities may trigger some pre-emptive rights (for instance for schools, parks, etc.).

Fig. 12. Different states for a parcel regarding planning.

When there is a pre-existing building, a demolition permit must be granted and when it was a plant (brownfield), depollution is need (Figure 12). In addition, at the vicinity of a historical monument or of an airport, special limitations apply.

Fig. 13. Any projected building must be compliant with planning rules.

C. Projected Buildings and Urban Planning Rules

Regarding planning zones, each has its own regulations. For example, campsites are not allowed in downtown. To simplify, three zones are defined, downtown, suburban area and rural area. However, historical monuments are located in downtown, and it is forbidden to construct new buildings round them (for instance, within 200 meters). In the rural area, there is an airport for which some limitations exist (Figure 11). See Rule 22.

$\exists C \in \text{City}, \exists \text{ZoneB} \in \text{Terr}, \forall B \in \text{Project.Building}, \forall P \in \text{Parcel},$ $\text{Topo}(C) \equiv \text{"Smart Town"},$ $\text{Topo}(\text{ZoneB}) \equiv \text{"Suburban Area"}$ $\models \text{Contains}(\text{Geom}(C), \text{Geom}(\text{ZoneB})),$ $\models \text{Contains}(\text{Geom}(\text{ZoneB}), \text{Geom}(P)),$ $\models \text{Contains}(\text{Geom}(P), \text{Geom}(B))$ $\vdots$ $B.\text{Height} \leq 15$ $\wedge \text{Area}(\text{Union}(\text{Geom}(\text{Floors}))/\text{Area}(\text{Geom}(P))) \leq 4$ $\wedge \text{Area}(B)/\text{Area}(\text{Geom}(P)) \leq 0.70$ $\Rightarrow$ $\models B.\text{ZoneB_Approved} \blacksquare$	Rule 22
--	---------

$\exists C \in \text{City}, \exists \text{ZoneA}, \text{ConservA} \in \text{Terr}, \forall B \in \text{Project.Building},$ $\forall P \in \text{Parcel}, \forall M \in \text{Monuments},$ $\text{Topo}(C) \equiv \text{"Smart Town"},$ $\text{Topo}(\text{ZoneA}) \equiv \text{"Downtown"},$ $\text{Geom}(\text{ConservA}) \equiv \text{Union}(\text{Buffer}(\text{Centroid}(\text{Geom}(M), 200))),$ $\models \text{Contains}(\text{Geom}(C), \text{Geom}(\text{ZoneB})),$ $\models \text{Contains}(\text{Minus}(\text{Geom}(\text{ZoneA}), \text{Geom}(\text{ConservA})), \text{Geom}(P)),$ $\models \text{Contains}(\text{Geom}(P), \text{Geom}(B))$ $\vdots$ $B.\text{Height} \leq 12$ $\wedge \text{Area}(\text{Union}(\text{Geom}(\text{Floors}))/\text{Area}(\text{Geom}(P))) \leq 3$ $\wedge \text{Area}(B)/\text{Area}(\text{Geom}(P)) \leq 0.80$ $\Rightarrow$ $\models B.\text{ZoneA_Approved} \blacksquare$	Rule 23
--	---------

But for Downtown and Rural Area, the situation is more complex. For Downtown, we have to take historical monuments into account, for instance within a 200 m buffer around its centroid. See Rule 23.

For the Rural Area, taking the airport into consideration, we need to consider three areas, the area outside the airport, the airstrip in which any building is forbidden, and finally in the “bowtie” with additional limits (Rule 24). Technically speaking, we have to use an “exclusive or”, noted $\oplus$ between those three possibilities.

$\exists C \in \text{City}, \exists \text{ZoneC}, \text{Bowtie}, \text{Airstrip} \in \text{Terr},$ $\forall B \in \text{Project.Building},$ $\forall P \in \text{Parcel},$ $\text{Topo}(C) \equiv \text{"Smart Town"},$ $\text{Topo}(\text{ZoneA}) \equiv \text{"Rural Area"},$ $\text{Geom}(\text{Bowtie}) = \text{Polyg}(640, 243; 657, 290; 748, 387; 796, 405; 743, 459; 729, 406; 636, 316; 580, 297),$ $\text{Geom}(\text{Airstrip}) = \text{Polyg}(670, 311; 724, 365; 707, 386; 650, 330),$ $\models \text{Contains}(\text{Geom}(C), \text{Geom}(\text{ZoneC})),$ $\models \text{Contains}(\text{Geom}(P), \text{Geom}(B))$ $\vdots$ $(\text{Contains}(\text{Minus}(\text{Geom}(\text{ZoneC}), \text{Geom}(\text{Bowtie})), \text{Geom}(B)),$ $\wedge B.\text{Height} \leq 12 \wedge \text{Area}(\text{Union}(\text{Geom}(\text{Floors}))/\text{Area}(\text{Geom}(P))) \leq 0.5$ $\wedge \text{Area}(B)/\text{Area}(\text{Geom}(P)) \leq 0.30$ $\oplus \text{Disjoint}(\text{Geom}(\text{Airstrip}), \text{Geom}(B))$ $\oplus (\text{Contains}(\text{Geom}(\text{Bowtie}), \text{Geom}(B))$ $\wedge B.\text{Height} \leq 8 \wedge \text{Area}(\text{Union}(\text{Geom}(B.\text{Floor}))/\text{Area}(\text{Geom}(P))) \leq 0.5$ $\wedge \text{Area}(B)/\text{Area}(\text{Geom}(P)) \leq 0.30$ $\Rightarrow$ $\models B.\text{ZoneC_Approved} \blacksquare$	Rule 24
--	---------

Now, to be fully approved by the local administration, whatever its location, a projected building to be approved must follow the subsequent rule; here again, exclusive-ors must be used (Rule 25).

$\exists C \in \text{City}, \forall B \in \text{Project.Building},$ $\text{Topo}(C) \equiv \text{"Smart Town"},$ $\models \text{Contains}(\text{Geom}(C), \text{Geom}(B))$ $\vdots$ $(\text{Contains}(\text{Geom}(\text{ZoneA}), \text{Geom}(B)) \wedge B.\text{ZoneA_Approved})$ $\oplus (\text{Contains}(\text{Geom}(\text{ZoneB}), \text{Geom}(B)) \wedge B.\text{ZoneB_Approved})$ $\oplus (\text{Contains}(\text{Geom}(\text{ZoneC}), \text{Geom}(B)) \wedge B.\text{ZoneC_Approved})$ $\Rightarrow$ $\models B.\text{FullyApproved} \blacksquare$	Rule 25
--	---------

Of course, those rules can be enriched to take depollution for possible brownfields into account as exemplified in Rule 5, flood restriction, etc. Moreover, additional rejection rules can be written.

VIII. CONCLUSIONS

The goal of this paper was to present a mathematical language for geographic static rule modeling, independent from any computer language and able to integrate all semantics. Thus, this language is based on several

mathematical domains such as logic, set theory, computational geometry, topology, etc.

At the moment, only several hundreds of 2D static rules are modeled with an interesting expressive power. For the complete modeling of any geographic rule, research must be carried out in several directions:

- integration of 3D, especially for terrain modeling and engineering networks;
- integration of temporal issues; this will lead to dynamic geospatial rules;
- integration of rules deriving from continuous fields, especially for dealing with meteorology, pollution, etc. and other aspects in physical geography;
- integration of additional clauses to extend its expressive power, overall to deal with networks whatsoever, electricity, sewerage, bus lines, etc.;
- looking for more issues in order to enrich semantics, especially for the automatic adaptation to special contexts; for instance, how to adapt a rule such as "when planning a metro, move underground engineering networks" to various street configurations;
- transformation of this mathematical language into a computer language;
- study of metadata relative to geographic rules (origin, etc.);
- design of an inference engine to reason with those rules;
- defining the organization of rules together for their access mechanisms taking temporal and spatial superseding mechanisms.

To conclude this paper, let me thanks the anonymous referees for their work and especially their fruitful comments.

REFERENCES

- [1] Boley H. (2006) "The RuleML Family of Web Rule Languages". International Workshop on Principles and Practice of Semantic Web Reasoning. Springer, Berlin, Heidelberg, pp. 1-17. Can be downloaded from <http://www.cs.unb.ca/~boley/papers/ruleml-family.pdf>
- [2] Boley, H., Paschke, A., Shafiq, O. (2010) "RuleML 1.0: The Overarching Specification of Web Rules". In Semantic Web Rules: International Symposium, RuleML 2010, Washington, DC, USA, October 21-23, 2010, Proceedings (Vol. 6403, p. 162). Springer Science & Business Media.
- [3] Dietz J.L.G. (2008) "On the Nature of Business Rules". In Advances in Enterprise Engineering, Springer Verlag Lecture Notes in Business Information Processing (10) pp. 1-15.
- [4] Egenhofer M. & Franzosa R.D. (1991) "Point-set topological spatial relations", International Journal of GIS, vol.5, no.2, pp. 161-174.
- [5] Espinasse B. (2017). "Introduction to Semantic Web Rule Language - SWRL". Can be downloaded from <http://www.lsis.org/espinasseb/Supports/ONTOWS/SWRL.pdf>
- [6] Graham I. (2006) Business Rules Management and Service Oriented Architecture: A Pattern Language. London, John Wiley.
- [7] Kainz W. (2002) Fuzzy Logic and GIS. Can be downloaded from https://homepage.univie.ac.at/Wolfgang.Kainz/Lehrveranstaltungen/E_SRI_Fuzzy_Logic/File_2_Kainz_Text.pdf
- [8] Karasova V., Krisp J.-M., Virrantaus K. (2005) "Application of Spatial Association Rules for Improvement of a Risk Model for Fire and Rescue Services". Proceedings on the 10th Scandinavian Research Conference on Geographical Information Science (ScanGIS), Stockholm, Sweden, pp. 183-193.
- [9] Laurini, R. (2014) "A Conceptual Framework for Geographic Knowledge Engineering", Journal of Visual Languages and Computing (2014), Volume 25, pp. 2-19.
- [10] Laurini R., Servigne S., Favetta F. (2016) "An Introduction to Geographic Rule Semantics". In Proceedings of the 22nd International Conference on Distributed Multimedia Systems (DMS 2016), Salerno, Italy, November 25-26, 2016. Published by Knowledge Systems Institute, ISBN: 1-891706-40-3, pp. 91-97.
- [11] Laurini, R. (2017) Geographic Knowledge Infrastructure for Territorial Intelligence and Smart Cities. ISTE-Wiley. 250 p.
- [12] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet Physics, doklady, volume 10, pp. 707-710.
- [13] Malerba D., Esposito F., Lisi, F., Appice A. (2003). "Mining Spatial Association Rules in Census Data. Research in Official Statistics. 5. Can be downloaded from https://www.researchgate.net/publication/2839474_Mining_Spatial_Association_Rules_in_Census_Data/download.
- [14] Morgan T. (2008) Business Rules and Information Systems: Aligning IT with Business Goals. Addison-Wesley.
- [15] Pittl B., Fill H.-G. (2018) "A Visual Modeling Approach for the Semantic Web Rule Language". Semantic Web Journal. September 2018. Can be downloaded from <http://www.semantic-web-journal.net/system/files/swj2023.pdf>
- [16] Ross R. G. (2011) "More on the If-Then Format for Expressing Business Rules: Questions and Answers", Business Rules Journal, Vol. 12, No. 4 (Apr. 2011), URL: <http://www.BRCommun2002ity.com/a2011/b588.html>.
- [17] Rademacher G. (2019) Railroad Diagram Generator. <https://bottlecaps.de/rr/ui>
- [18] Salleb-Aouissi A., Vrain C. & Cassard D. (2015) "Learning Characteristic Rules in Geographic Information Systems". N. Bassiliades, G. Gottlob, F. Sadri, A. Paschke, D. Roman (Eds.). Rule Technologies: Foundations, Tools, and Applications, 9th International Symposium, (RuleML 2015), Aug 2015, Berlin, Germany. Springer, 9202, 2015, Lecture Notes in Computer Science.
- [19] Smith B., Varzi A., (2000), "Fiat and Bona Fide Boundaries". Philosophy and Phenomenological Research Vol. LX, No. 2, March 2000, pp. 401-420.
- [20] Shekhar S., Huang Y., (2001) "Discovering Spatial Co-location Patterns: A Summary of Results", in Proc. of the 7th Int'l Symposium on Spatial and Temporal Databases, 2001, pp. 236-256.
- [21] Shoorcheh M. (2018) "On the spatiality of geographic knowledge". Asian Geographer. 36:1, pp. 63-80, DOI: 10.1080/10225706.2018.1463854.
- [22] Varadharajulu P., West G., Mcmeekin D., Moncrieff S. & Arnold L. (2016), "Automating Government Spatial Transactions". In Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM), held in April 2016, Rome, Italy, edited by Jorge Gustavo Rocha and Cédric Grueau, published by Scitepress, ISBN: 978-989-758-188-5, pp. 157-167.
- [23] Zhou C., Xiao W.D, Tang D.Q. (2016) "Mining Co-Location Patterns from Spatial Data". ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume III-2, 2016 XXIII ISPRS Congress, 12-19 July 2016, Prague, Czech Republic, pp. 85-90.

Metaphorical Parametric Model for Brand Mark Design

Towards a Universal Model of Computational Visual Communication Design

Po Choy Ng

School of Design
Hong Kong Polytechnic University
Hong Kong
ng.pchoy@connect.polyu.hk

Clifford Sze-Tsan Choy

School of Design
Hong Kong Polytechnic University
Hong Kong
mccliff@polyu.edu.hk

Abstract—Visual communication design involved with two aspects of visual representation, namely, information and expression. The idea of building a computational model for visual communication design has been developed for many years. However, this objective has only been partially fulfilled as most of the solutions are focused on showing information rather than expression. Lakoff and Johnson have claimed that “human thought processes are largely metaphorical.” Thus, a computational model for visual communication based on the concept of conceptual metaphor is proposed. As brand mark design is considered as a challenging design type that demands the representation of both information and expression, the first attempt of the development of this new model is targeted for this specific type of design. In this paper, the Metaphorical Parametric Model (MPM) is proposed as a computational model to represent brand mark design. The MPM is a type of model which represents the semantic structure rather than the synthetic structure of conceptual components. The components of an MPM is assembled based on the concepts of metaphorical relations which could be signified by spatial relations between the related objects. All objects and relations defined in the MPM are parametrically controllable which could be generated as different visual forms to represent both information and a wide range of metaphorical expressions.

Keywords – *visual language; visual communication design; conceptual metaphor; metaphorical expression;*

I. INTRODUCTION

The idea of formalizing visual communication design as a model of visual language has been carried out by art and design partitioners for many years. In general, the focus of these partitioners can be divided into two major groups—those focused on the communication of information [1, 2, 3, 4, 5] and on the communication of expressions [6, 7, 8]. Brinton [1] stated that: “The principles for a grammar of graphics presentation are so simple that a remarkably small number of rules would be sufficient to give a universal language.” With the advent of computing, his idea of making a universal visual language for graphic representation has been partially fulfilled. However, most of the previous research works are limited to the presentation of quantitative information such as variable data printing [9], automatic layout [10], font matching [11] or data visualisation [12]. Although there are some attempts on creating algorithms for generative art [13, 14] or generative design [15, 16], most of them are only focusing on the synthesis

of visual forms but not generate works based on the creative concepts or the concept behind the expressions. Yet, there is no well-established approach to represent both information and expressions.

As an attempt to create a universal computational model which can generate designs to present information as well as express emotion, this paper proposes the Metaphorical Parametric Model (MPM) for generating visual communication design. Since brand mark design is related to both information and expression, it has been considered as “one of the most difficult to perfect [17]”; it is chosen as the topic for an initial trial.

The paper is organized as follows: Section II describes the theoretical framework of the MPM. Section III presents the model of Metaphorical Parametric Model (MPM). The concept of how to represent visual objects for making metaphorical expressions will also be discussed. Section IV proposes the concepts of Metaphorical Relations (RM) and Metaphorical Expressions (ME). In Section V, an existing brand mark will be recreated to illustrate the process of generating designs with MPM and RM. Section VI further discusses the concept of MPM. Section VII concludes this paper and future development will be suggested.

II. BACKGROUND

In the following paragraphs, the theoretical framework of the MPM will be introduced and described. The major concepts include the theories of image schema, conceptual metaphor, visual metaphor and visual rhetorics.

A. Conceptual Metaphor as the foundation of language

The Conceptual Metaphor Theory (CMT) brought forward by Lakoff and Johnson in their seminal book “Metaphor We Live By” states that, “Our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature [18, p. 3].” CMT tries to investigate the fundamental cognitive problem and considers that it is the conceptual metaphor rather than language as the foundation of thoughts.

According to these authors, conceptual metaphors are different from the linguistic ones but specific cognitive concepts embedded in our bodily experience. By learning from the interactions with the environment, infant gains experience which structures the mind and formulates the “image schema”. Johnson [19] explained that the image schema is not an abstract, finite proposition, concept or concrete image but an evolving

pattern of our imaginatively structured experience. It is not only connected to perceptual phenomena but also our understanding of more abstract domains of our experience through imaginative metaphoric and metonymic projections. Image schema is now well accepted and the concept has been confirmed by research in cognitive sciences and developmental psychology [20, 21, 22] and as many as sixty-six image schemas are identified by Risch [23]. He explained that many information graphics are based on the principle of image schema whereas people's early perceptual experience of spatial relations such as "containment", "path-following" and "object dynamics" become generalized and formalized subconsciously. Eventually, these frameworks will be adapted and utilized in different types of information graphics.

Lakoff and Johnson [18] have differentiated three different types of conceptual metaphor based on the nature of the source domain of the mapping namely, the ontological, structural, and orientational metaphors.

Structural metaphor: this type of metaphor is grounded in systematic correlation within our experience. The concept is expressed with another different structured, clearly defined or well familiar concept such as "ARGUMENT IS WAR" in which all the corresponding concepts belong to the same structure, e.g. attack, indefensible, strategy, win, gain ground, etc., could be used as metaphorical expressions to form a systematic way of talking about the battling aspects of arguing.

Oriental metaphors: in this type of metaphor, the mapping is focused on adopting the structure of one concept rather than a whole system of concepts of the corresponding object. Most of these are related to spatial orientation such as up-down, in-out, front-back, on-off, deep-shallow, or central-peripheral related to the physical environment; hence, the orientational metaphor represents a spatial orientation concept such as HAPPY IS UP and resulted as an expression like "I'm feeling up today."

Ontological metaphors: the concept is based on the mapping of a source in the physical world to the target which could be an activity, emotion or idea. Take the "INFLATION IS AN ENTITY" as an example, by taking inflation as an entity, it allows us to quantify, identify it, take it as a cause and act upon it as the "Inflation is lowering our standard of living" or "Inflation is hacking us into a corner." Thus, this is physical to abstract mapping.

Grady [24] noticed that a wide range of conceptual metaphors had been recognized but he wondered why do some elements of domains get mapped, but not the others. Eventually, he distinguished two major types of metaphor namely, correlation-based metaphor and resemblance metaphors. The source domain of correlation-based metaphor is based on recurring sensory experience. He also noticed that the correlation-based metaphor evolved with the process of mapping the structures of the source domain to a target domain, so there is a "directionality" which is not symmetrical. Within the group of correlation metaphor, Grady described that there are primary source concepts which allow us to map the concrete experience such as "heaviness" to the abstract target concept "difficulty" or "brightness" to "happiness. Thus, the recurring correlated mapping concept between "up" and "good" is defined as primary metaphor "GOOD IS UP". According to Grady, primary metaphors are the basic structure that can be combined to form a larger structure called complex metaphors.

B. *Conceptual Metaphor Theory could be based on embodied experience, contextual issues, social and cultural conditions*

However, after many years of development, the original conceptual metaphor theory has been updated and transformed.

Szwedek [25] has criticized that structural and orientational metaphors are based on existing objects, thus metaphors are basically ontological. Moreover, Hernández and Pérez, [26] also argue that metaphor is not fundamentally ontological as explained by Lakoff and Johnson [18], image schema and primary metaphor are more fundamental.

In a special issue of the retrospective of conceptual metaphor after thirty years of its development, Riccardo and Morgagni [27] mentioned that some authors such as Leezenberg [28] suggested that embodied concepts do not come from an individual's experience only but largely shaped by the social, cultural and linguistic practice. Regarding the influence from society and culture, authors such as Brandt, Deignan and Cameron [29, 30] have pointed out that the conceptual metaphor is not all dependent on bodily experience but conditions affected by the context. All these lead to the direction that the CMT should concern with the conceptual patterns affected by cultural practices and contextual issues. Based on the experimental research and empirical work of different researchers, plenty of evidence [31, 32, 33] reveals that people understand certain domains through the other domains more often than through the language. It has been found that people talk and think about time in term of space and motion but not from the opposite direction.

C. *From Conceptual Metaphor Theory to Visual Metaphor and Visual Rhetoric*

Schilperoord, Maes, and Ferdinandusse [34] described that people tend to organize objects as cohesive configurations according to the century-old gestalt theory [35], hence, designers often deliberately utilize these grouping principles for their aesthetic and communication purposes. The principle involved with making visual expressions through the manipulation of metaphorical relations between concepts is defined as symmetric object alignment (SOA) in advertisements.

Through the integration of the concept of primary metaphor and the principle of SOA, Ortiz [36] explained that the formal configuration of visual metaphor in pictorial advertising could be distinguished as three types: the first type is the fusion of different objects into a single hybrid image; the second one resulted as one single object but the other is suggested by context; while the third one show at least two objects at once.

Regarding the nature between the aligned object chosen for the alignment, three different kinds have been suggested, including the alignment of different objects, the alignment of incompatible objects and the alignment of identical objects.

So, based on various combinations of the formal structures and the selected objects, the authors have analyzed a series of advertisement and identified five types of primary metaphors commonly found in advertisements which include "SIMILARITY IS ALIGNMENT", "SIMILARITY IS PROXIMITY", "THE NATURE OF AN ENTITY IS ITS SHAPE", "CONSTITUENTS ARE CONTENTS", and "CATEGORIES/SETS ARE BOUNDED SPATIAL REGIONS".

Ortiz summarised that the nature of SOA could be either metaphorical or literal. The SOA is metaphorical when the object conveys the idea of similarity and belongs to the same category; also, when the alignment, proximity, shape, inclusion, and space indicate the presence of primary metaphors which are grounded by experiences. The SOA is literal if the element presented does not seem similar but merely shows objects different from the others.

D. Metaphor in information visualization

Risch [23] made a comprehensive discussion on the use of image schema and metaphor for the visualisation of information. He identified two types of metaphor which could be extended to the direction of making metaphorical expressions. He argued that there are two distinctive types of graphics namely analogical graphics and metaphorical graphics. Analogical graphics depicts the inherent spatial relations form the source to the target of the same domain such as the mapping of the length of line to a dimension. On the hand, metaphorical graphics, expressing non-spatial concepts in spatial terms through the functional alignment with image schemas. In conclusion, he proposed that “the standard inventory of image schemas derived from linguistic and cognitive studies can serve as the basis for developing a kind of visualization “grammar.” Such a grammar would employ graphical analogues of image schematic patterns as syntactic elements that relate concepts expressed using conventional signifiers such as text, colour, and symbology”.

III. CORPUS ANALYSIS OF BRAND MARKS

Pragglejaz Group (2007) has proposed the “metaphor identification procedure” (MIP) for identifying the metaphorically used words in discourse. They described that different researchers will handle the materials and process differently based on their purpose whereas some may do a large corpus while the experimental psycholinguists may only verify or identify the target words or discourse in small sources.

For this study, due to the nature of the corpus is visual form rather than text, the corpus analysis demands the judgement of human rather than the processing of computers. Thus, under the constraint of time and resource, a flexible and pragmatic approach is adopted. The process focuses on two key issues including:

- The identification of the elements and structures of brand mark designs.
- The identification of the use of the theories of conceptual metaphor and visual rhetorics in brand mark design.

Two sets of books [37, 38] with collections of approximately 8000 brand marks are chosen. Though one of the publishers of these books is from Japan while the other is from Korea, the samples in the collections are selected from worldwide. In general, these type of books are used by designers for reference, most of the samples in the collections were well selected by editor with design expertise. Hence, the samples in these collections consist of the choice of two different groups of experts who would screen for professional grade design samples. Those digital files distributed with the book is another crucial factor to consider the choice of these collections. Thus, all the brand marks can be loaded into the database for the corpus analysis without extra preparation work beforehand.

Two fresh graduate design assistants have been assigned to identify the shapes and objects found in each sample. All the shapes found in the samples are deconstructed and identified

as one or more of the shapes listed in the code book which is compiled based on the literature review. During the analysis of the corpus, if any shape extracted from the mark cannot be classified as a shape listed in the code book, the new item will be added to the code book until all the unique shapes were recorded. The objective of this task is to identify the smallest essential set of shape required for brand mark design. Other than the basic geometric shapes, the assistants have to identify and record the object represented by these basic shapes. Here, the basic shapes are defined as the simple, primitive forms found in brand marks, they may or may not resemble any object found in reality. However, these forms mostly appear as geometric shapes or distinctive types of shapes which could be represented by mathematical equations. On the other hand, objects could be either physical entities or graphic representations, they could be either natural or manmade objects. Thus, two small circles on a circle may signify a human face as Fig. 1 when they are reading as a single object. On the other hand, the same configuration could be considered as two small circles overlay on the upper part of a larger circle if they are perceived as separated shapes. In such a case, both the circle and human face will be an entry as shape and object respectively.

Figure 1. Two dots enclosed in a circle, forming a human face

From the result of the analysis, certain distinctive patterns are noticeable, part of the findings are listed and discussed as follow:

TABLE I. SUMMARY OF THE BRAND MARK ANALYSIS

Item	Count
Total number of brand marks analysed	8095
Total Object count	23738
Unique object identified	1538
Geometric Shapes	11784 (49.65%)
Organic Shapes	1195 (50.35%)

As TABLE I shown above, nearly half of the shapes in the corpus are abstract forms and mostly can be considered as geometric shapes. These brand marks may consist of one or more components which may appear as abstract forms, iconic objects or a combination of both. The findings reflect that both geometry and iconic shapes are essential for brand mark design. Since geometric shapes do not depict specific objects but mostly perceived as abstract forms, they can be used as components of design to represent a wide range of objects or topics. In many cases, it is the composition and arrangement of these objects that constitute the meaning of the brand marks. The figure listed above indicates that without using complex forms or many objects, it is possible to generate a wide range of brand marks in the collections with a small set of objects using geometric shapes.

In the second phase, the metaphorical properties and metaphorical expressions prevail in these marks will be analysed. Since the first hundred of the most frequently used objects in the corpus accounts for 67% of total object count, this study will focus on these 100 objects for further analysis instead of going over thousands of objects for efficiency. As the metaphorical expressions of each object are affected by the conditions, context and culture, it is expected that the annotations done

by assistants may not be precise and comprehensive if judged from different perspectives. However, because of the analysis of this part is not aimed for making an exhaustive analysis of the metaphorical features of the corpus but rather use it as a strategy to identify how the theories of conceptual metaphor are utilised in brand mark design. So, even though we notice the limitation of the corpus analysis, we focus on how to represent the objects and their semantics identified in the analysis for the purpose of building the computational models. Nevertheless, we have considered that in-depth annotations of a wide range of metaphorical properties of many objects by participants of different cultural backgrounds are necessary in future.

The result of the corpus inventory and analysis indicates that the objects used in the brand mark designs consist of either iconic or abstract forms. These designs may consist of single or more items in which several basic components could be combined together as an abstract or iconic object. Thus, based on the background theories described above and the result of the analysis just mentioned, the current study proposes the MPM model for the purpose of generating brand mark designs.

Metaphorical Parametric Model (MPM) is a type of two-dimensional model specifically designed for making visual metaphorical expressions based on conceptual metaphor theory and visual rhetoric. Instead of representing the synthetic structure of a specific object, it represents the semantic components and the semantic relations in between. In applications, the synthetic process of MPM is guided by its semantic components. Thus, through different combinations of parametric settings of the MPM, different forms could be generated to signify a wide range of metaphorical expressions.

A. MPM Types

MPM consists of two different types of entity namely element and compound. An element is a single, simple and distinctive visual object while a compound is composed of two or more components which could be either an element or another compound.

B. Elements and Metaphorical Properties

Based on the analysis of the configurations of the samples, approximately sixty primary components are encoded as the “elements” of the MPM currently. An element is the most common and essential form found in a specific domain; it is a simple, abstract, and geometric-like two-dimensional object generated by certain geometric logic and equations while each form is controlled by one or more parameters. Elements can be classified as two types, namely line and shape. The major difference between line and shape is that a line is defined as an open path in the system; therefore, it has no enclosed area. However, both types could suggest different expressions according to the art and design theories introduced by Kandinsky and Klee [6, 7, 8].

Among all graphic forms, both line and shape could trigger cognitive responses. Though experiments with the imaging from fMRI, Larson, Aronoff, Sarinopoulos and Zhu [39] has recognized that the data support the idea that visual threat can be triggered by a simple downward V-shape with no reference of the other contextual or affective cues. Besides, the test of the associations of meanings between the downward-pointing and upward-pointing triangles also support the hypothesis that simple geometric forms convey emotions. These results signify that our brains could detect the stimulus of the geometric shapes with much less stimulus information than earlier studies.

Thus, as indicated in TABLE II, with different parameters, a single element such as the Archimedean Spiral or Trapezium could suggest a wide range of possible expressions as listed. Regarding the semantics of lines and shapes, many results and findings done by researchers such as the study of lines by Ibáñez and Delgado-Mata [40] and the study on shape by Blazhenkova and Kumar [41] can be adapted to guide the parametric settings of the MPM for signifying the intended expressions.

TABLE II. GENERATED FORMS OF VARIOUS MPM AND THEIR POSSIBLE METAPHORICAL EXPRESSIONS

Archimedean Spiral					
Regular		Offsetted		Outward

Possible Metaphorical Expressions (Respectively)					
regular	harmony	graceful	living	springy	dynamic
hypnotic	gentle	organic	biological	cheerful	changing
cryptic	intricate	nature	structural	extroverted	unexpected
mysterious	infinity	exquisite	dynamic	excited	energetic

Trapezium					
Asymmetric		Large width/ high ratio		Complex

Possible Metaphorical Expressions (Respectively)					
free spirited	distinct	structured	precise	opposing	focused
unrestricted	individual	geometric	absolute	conflicting	complicated
open	intentional	dignified	strong	against	difficult
independent	anarchic	historic	bold	contradict	agitated

C. Compounds

A “compound” of MPM is a two-dimensional visual object formed by two or more components assembled with some specific relations. A component of a compound can be either an element or another compound. A compound only provides two or more semantic placeholders for holding objects, it has no definite synthetic form other than encoded with a set of semantic relations controllable by parameters. In general, a set of default parametric settings which matches with the perceptions of most of the people will be assigned to the initial form of the compound.

When different objects are used to substitute the default sets or different parameters are assigned to the compound, the result could be quite distinct from the default settings. Fig. 2 is the conceptual diagram of the compound “sun”. The top right version in Fig. 3 is the “sun” generated with the default settings. With different parameters, even without changing the objects assigned to the placeholders, the other variants in Fig. 3 become quite departed from the default version.

Figure 2. The conceptual structure of a compound “Sun”

Figure 3. Sun generated with different parameters

As the components and the parameters in the compound “sun” can be switched according to the semantics required, it can be used to constructed different design as shown in Fig. 4 and 5. in which the concept of the sun is still retained but the visual form and expression are different.

Figure 4. Existing brand mark recreated using compound “sun” on the right, original design on the left

Figure 5. The other existing brand marks that could be represented by the compound “sun”

D. Deconstruction and Construction of Compounds

To create a compound, the target object should be deconstructed as the relevant units and put together as a conceptual structure matched with the perception of most of the people or the target audience. Depending on the background, cultures or the shared experience, different people may have a different perception of the structure of the object. In additions, as the focuses or viewing angles towards the same object could be coming from diverse perspectives, that object could be deconstructed and reconstructed in many different ways. For example, a bird is a simple object, but it is unlikely to provide one model for all as a bird can stand still or fly; it could be viewed from the bottom, the side or from the top. Unlike 3D modelling, it is necessary to build various 2D models to anticipate different conceptions or situations. Carlier, Leonard, Hahmann, Morin, and Collins [42] have done a study to collect information about how people decompose the structure of shapes. Their study covers 1200 shapes in 70 shape classes, and 2861 participants with 41,953 annotations have been collected in which at least 24 annotations per shape. The result clearly reflects that there are many ways to decompose shapes into diverse configurations according to different perceptions. Therefore, many compounds could be created to represent different perceptual models of the same object.

IV. METAPHORICAL RELATIONS, RELATIONAL OPERATIONS AND METAPHORICAL EXPRESSIONS

Unlike the other models, the MPM only provides placeholders for assembling objects together according to specific relations. Although many researchers have made tremendous efforts on the study of mereotopological relations, the focuses are mostly on the studies of physical conditions or spatial relations for engineering, mathematical or scientific purpose. In order to build a model to signify metaphorical expressions, metaphorical relations (MR) is proposed to facilitate the significations of metaphorical expressions with the MPM. For constructing the specific spatial relations to signify the MR as identified in the corpus analysis as described above, a number of relational operations (RO) are encoded in the MPM. The detailed discussion of the features and functionality of MR and RO will be discussed in the following sections.

A. Relational Operation (RO)

Relational operation (RO) is the operation which arranges the position of objects according to the specific spatial or visual relations. These relational operations could arrange the objects according to the concepts of different MRs and then signifying different metaphorical expressions.

Currently, fourteen ROs which are essential for the arrangement of the components are defined in the MPM. These relations include: alignTo, topOf, bottomOf, rightOf, leftOf, rotateAroundPoint, parallelTo, perpendicularTo, scaleBy, matchOneSide, matchExactSize, inside, meetAt and tipOf.

B. Metaphorical Relations and Metaphorical Expressions

A large proportion of conceptual metaphors belong to the orientational metaphor type is associated with spatial relations. While mathematicians and researchers have been investigating the related theories for many years [43, 44, 45], their efforts have seldom come across the notice of visual communication designers. Nonetheless, the diagrammatic renditions of their mathematics concepts provide great resources to signify meanings and make a wide range of metaphorical expressions for different types of visual communication designs. In fact, these spatial relations which could be considered as a kind of visual schema have been used to signify different meanings in the pictogram of Chinese oracle-bone inscriptions in ancient. Two distinctive examples in Fig. 6 showcased the principle of orientational metaphor by positioning the shorter strokes in various relative positions and suggest different semantics.

Figure 6. Ancient Chinese characters based on pictogram for “up” (left group) and “down” (right group) and their modern counterparts

The RO developed for the MPM is originally created for the synthesis of the visual composition; it is mainly aimed at managing the mereotopological relations between different components in the MPM with geometric transformations. However, in the later stage of the development of the MPM, it has been found that these merotopological relations could be constructed according to the principle of image schema to signify specific information and expression metaphorically. Therefore, other than simply arranging the composition of the design with the synthetic operations such as RO, another layer of meaning can be suggested by constructing the Metaphorical Relations (MRs) through the ROs as well.

TABLE III. PERCEPTUAL RELATIONS BETWEEN TWO OBJECTS AND THEIR POSSIBLE METAPHORICAL EXPRESSIONS

Relation	Description	Logo Examples			Possible Metaphorical Expressions			
Positional								
Top of	A is on top of B					authority climax dominant	high rank important hierarchy	superior good influential
Bottom of	A is under B					foundation earthy lower rank	base basic support	endorse subordinate inferior
Left of	A is on the left side of B					accompany assist backward	extend lead liberal	member past peer
Right of	A is on the right side of B					active advance assist	extend forward follow	future companion lateral
Front of	A is in front of B, hiding part of B					apparent credible dominant	exposed forward important	inviting noticeable open
Back of	A is behind B, hiding part of A					backing back up concealed	emerge hidden introverted	obscure dependant disguised
Proportional								
Larger than	Size of A is larger than size of B					authoritative dominant established	expending full generous	influential plentiful powerful
Smaller than	Size of A is smaller than size of B					delicate fine gentle	humble insignificant minor	modest progeny subdominant
Same Size as	A and B are of same size					accompany balance equality	even matching partner	symmetric competitor fair
Part-Whole Relational								
Inside	A is inside B without touching its boundaries					bounded protected secure	sheltered surrounded attention	cherished child comforting
Covered By	A is a component of B					belong to component included	incorporated integral involved	member combined dependent
Covers	B is a component of A					basis central contextual	embracing progenitor cooperation	dynamic essential functioning
Contains	A includes B without touching its boundaries					background backing environment	inclusive protection safeguard	take care of approve assistance
Meet	A and B shares boundaries at one point					agreement contact equality	touch associate closeness	gentle intimate link
Overlap	A and B are partially overlapping each other					affiliated associated bond	consensus equality friendly	harmonious linked share

In TABLE III a number of spatial relations are listed with brand mark examples to illustrate how specific arrangement of visual objects can potentially signify different expressions. TABLE III only indicated one-to-one relations; however, these relations could be applied to one-to-many or many-to-many situations while the expressions may show some minor variations. Based on these spatial relations, the MPM could suggest a kind of relations which may be called as metaphorical relations (MR). Through the MR, different kinds of metaphorical expressions (ME) could be signified through the process of symmetrical object alignment (SOA). Through the process of SOA, the abstract qualities will be represented as concrete images. By choosing a source object with specific metaphorical properties and combined with the relevant MR, the metaphorical properties could be mapped to the target object. Thus, not only the literal meanings but also the metaphorical expressions could be delivered to the receivers.

C. Setting the parameters based on semantics

As the whole idea of the MPM is dealing with semantic, especially those related to expression and emotion, the principle of semantic differential could be utilised to control the parameters. By mapping the value of a semantic differential scale relevant to a specific parameter of a metaphorical relation, the form of the model could be modified and express such meaning respectively. For example, the weight of “authority” can be mapped to the relation “top of” whereas the distance between the top and bottom objects indicates the level of the “authority”. The position of the top object could be placed at either a higher or lower position to signify different degrees of authority as shown in Fig. 7. At the same token, we may enhance the effect by concatenating the “larger than” relation to enhance the effect by mapping the ratio of large to the size of an object.

Other than passing the semantic value to the parameters based on the principle of symmetrical object alignment and semantic differential, it is expected that different types of machine learning methods could be applied to drive the model if the proper interface is provided. However, the experimentation of these ideas are out of the scope of this paper and future development should be done in future.

Figure 7. (a) Original brand mark, (b) increase value of parameter “yOffset” in TopOf and (c) increase value of parameter “xScale” of ScaleBy

D. Resolving problems of contradicting relations

As some of the samples such as TopOf/BottomOf, LeftOf/RightOf shown in TABLE III are contrary pairs, some relations will signify divergent meanings depends on which object is under attention. For example, the object “A” may carry meanings of “authority”, “high rank” or “good” while the object “B” may show the ideas of “support”, “low rank” or “inferior”. In some sense, the expressions of “authority” vs. “support”, “high rank” vs. “low rank” are complementary pairs. However, the expression of “good” and “bad” could be contradictive when a unified brand personality is preferable. In general, this type of conflict could be avoided if the MPM is assembling according

to the theory of visual rhetorics as only the appropriate object will be selected. Although the detail discussion of visual rhetoric is out of the scope of the current discussion, a brief description could elucidate how the MPM would work with semantic issues. In general, if a unified concept is required, the choice of the nature of both object “A” and “B” should complement each other as the samples show in the top row of TABLE III. So, the expression of the complementary pair such as “authority/support”, “high rank/low rank”, and “good” will be dominant while the contradicted meaning “bad” will be submerged. In fact, the metaphorical expressions of a design are affected by the nature of the objects as well as the relations between the objects found in the design.

When a visual rhetorical figure is applied to a brand mark design, if the objects appearing in the design carry distinctive meanings, the metaphorical expressions suggested by the metaphorical relations between the objects could be dominated by the properties of the objects. For example, the construction of the brand mark in Fig. 8, is based on the principle of the rhetorical figure “Antithesis”. The object “A” and “B”, which carry the “opposition of content” as defined by Durand [46], are combined together as a single compound. The composition of this design can be constructed by at least two different metaphorical relations namely, “LeftOf/ RightOf” and “Cover/ CoverBy”. In this case, metaphorical relations suggested by the “LeftOf/RightOf” or “Cover/ CoverBy” will become minor. Here the contradictive expressions suggested by the “happy face” object and “sad face” object on both sides are showing in parallel and suggested the expression of “Antithesis” of two emotions to represent a theatre group.

Figure 8. Brand mark based on the principle of rhetorical figure “Antithesis”

V. AN EXAMPLE OF THE PROCEDURE OF MAKING BRAND MARK DESIGN WITH MPM

In this section, a formal procedure of making brand mark design with MPM is described so that users can acquire a better understanding of how to utilize the model in real life applications.

- A designer or user should get the design specifications with information such as the nature of the business, the target group, the brand personalities or specific requirement collected in advance. Here, for the purpose of demonstration and explanation, an existing brand mark of Swiss Airline is chosen so that we can show how to create the same result with the MPM.
- In general, visual communication designers will seek visual metaphor to represent the design concept. The choice of the metaphor could come from analytical processes such as a visual inventory of the other existing work created for a similar business or lateral thinking such as brainstorming. As the current task is based on an existing design, the first thing to do is analysing and find out the metaphor used in the original design. Through the deconstruction of the original brand mark, two visual metaphors representing the concepts “Swiss” and “airline” are identified.

- Depends on the design requirements, a brand mark design can be created by either an element, a compound or different combinations of both. As the Swiss Airline consists of two metaphorical concepts, creating a compound with two components will be a logical approach to initiate the design. To create a compound to represent these two concepts with visual objects, the MPM consists of one placeholder to contain the concept “Swiss” and another placeholder for the concept “airline”. Then these two placeholders are connected by one relational operation “inside”.
- Then the metaphorical concepts are encoded as a compound as shown in TABLE IV. To recreate the form of the original design, two elements, including a trapezium and a cross have been assigned to the placeholders of the compound. By setting the relevant parameters, a replicate can be created.
- By modifying the parameters or element of the compound, design alternatives could be generated as shown in TABLE V. With different elements in the back, the concept of “airline” will be omitted and signify different semantics according to the metaphorical properties of the replaced element. Similarly, changing the front element will signify different meaning as well.

By using the RO to create different MRs between the elements of the original composition, new metaphorical expressions could be suggested. By referring to the relations shown in TABLE III, each of these new relations could signify a manifold of meaning diverged from the original meaning.

As MPM and its related operations are still under development currently, the use of the coding library is yet not released to public usage. However, a prototype built with FileMaker, which is a relational database supporting HTML and Javascript and convenient for rapid prototyping, is available for designers with no programming experience to do design or experimentation. Through the GUI (Fig. 9), users can enter the ROs to assemble different elements as compounds or more complex compound combined with the other compounds. By selecting different objects for the placeholder and modifying the parameters of the objects or the relations operations, users could create a wide range of design with the prototype.

Figure 9. Screenshot of Prototype with GUI Interface

TABLE IV. CODED SIMULATION OF EXISTING BRAND MARK DESIGN

Swiss Air Brand Mark Example	
Original Brand Mark	Simulated Result

Code Sample (Javascript)	
<pre>//create trapezium to signify airline var tail = Trapezium.createComposed({ length: 190, height: 170, angleLeft: 60, angleRight: 90.5 }) tail.gt("skew", "x", -13, "CM", "g") tail.style("fill", "red") //create cross to signify Switzerland var cross = CrossShape.createComposed({ width:200, height:200, thicknessX: 60, thicknessY: 60 }) cross.style("fill", "white") //establish MR between the tail and the cross cross.addRelation("alignTo",tail,"CR","CR",[15,0]) cross.addRelation("inside", tail, 0.85) //plot resulting brand mark onto screen var logo = Compo.createComposed({}) logo.addItem(tail, cross) logo.plot()</pre>	

TABLE V. VARIATIONS OF THE SIMULATED RESULT

Simulated Result	Description
Change of Object
By changing the shape of the back element, different designs are created. These shapes are not the visual metaphors to represent the concept “airline”; however, they could represent the other brands from the Swiss.
Change of Relation
By changing the shape in the front element, another group of design are created. As the visual metaphor of “Swiss” is replaced, this design could signify an airline associated with different places or institutions.
By changing the relations between the elements using Right Of, Part Of, Overlap and Top Of instead of Inside. The semantic meaning will change and signify different metaphorical expressions.

VI. DISCUSSION

In order to create a universal model to facilitate visual communication design, building a model for expressing emotion is as important as showing information. The MPM proposed in this paper is an initial attempt to develop a new type of computational model for visual communication design based on the conceptual metaphor theory and visual rhetoric. The model is a representation of the semantic and metaphorical structure rather than any specific physical structure of a target object. Unlike some conventional architectural models which represent the physical structures of architectural forms, it only represents the most dominant conceptual structure perceived by the target groups or the designers. Why utilize such configuration instead of using a more definite structure as most of the other models? The major reason is the component of this model is not limited to a definite ontological structure but depending on different levels of metaphorical concepts. These concepts are affected by our embodied experience, our knowledge prevailing in a specific group, society or culture as well as the practice enforced by institutions.

Based on the theory of symmetrical object alignment, we can map the metaphorical properties from a source object to the target object and signifying the intended metaphorical expressions. However, choosing the relevant components to construct a computational model requires specific domain knowledge. Other than the knowledge of computing and coding, defining an MPM for making professional grade design requires the understanding of the brand mark design as well as the understanding of the culture of the target audience.

When a visual form is displayed, its meaning not only affected by the objects perceived by the viewers but the effect of visual rhetorical treatments is also crucial. For the same object, the choice of colour, line quality, texture or particular style could affect the expressions of the outcome. Graphic attributes always carry another layer of metaphorical properties which could affect the expressions signified in the design. A detail discussion of the visual rhetorical effect of graphic attributes is out of the scope of this paper, however, the current MPM have already incorporated the concept that all graphics attributes can be used to signify metaphorical expressions as the examples shown in TABLE VI.

TABLE VI. VARIATIONS OF THE SIMULATED RESULT OF THE CBS BRAND MARK

<i>Change of Attribute</i>
Changes in actual vector point location
Changes in styling methods (vector points are not changed)

VII. CONCLUSION AND FUTURE DEVELOPMENT

It is obvious that the current MPM is created with vector graphics and could not represent photographic or raster images in the design. However, with the relevant adjustments, the MPM could incorporate the raster image processing functions to generate both vector and raster graphics at once.

This paper mainly focuses on how the conceptual metaphor theory is employed in establishing the structures of MPM. It explains the assembling of visual objects as conceptual metaphors for making metaphorical expressions through the manipulations of metaphorical relations. Though it has explained the concept of symmetric object alignment, there is no discussion on how to choose which object as the source object and how it is mapped to the target object. Also, there is only a brief description on how to arrange the objects to compose the final design based on the theory of visual rhetoric in section IV. D. In order to elucidate the design process such as how to select objects and control their parameters, it will require a more detail introduction and in-depth descriptions of visual rhetorical operation (VRO) which is out of the scope of this paper. Currently, a few of VROs based on the seminal paper of Durand [46] have been implemented to facilitate the construction of the MPM. By using the VRO together with the MPM, plenty of diversified designs could be generated and the results are promising. In future, besides the rhetorical figure listed by Durand, a much wider range of visual rhetoric figures which has been reviewed by Baruchello [47] as well as Huhmann and Albinsson [48] should be implemented as VRO for working with the MPM.

The MPM described in this paper is only the first trial of a new type of model for visual communication design. However, its potentials should not be limited to brand mark design. Based on some experiments conducted before, the same model can be used to simulate various types of pattern design or generative art as shown in Fig. 10. This model is flexible and the users can modify the components and relations based on the theory of conceptual metaphor or visual rhetoric for an instant. Furthermore, though this model is initially developed for visual communication designers dealing with two-dimensional works, it could be adapted in different design disciplines immediately. For example, this model could be an ideal creative supporting tool (CST) for architects who need to design the floor plans or facade layouts with symbolic meanings. As all the objects and relations in the MPM are encoded with the conceptual metaphor and visual rhetoric, it will be quite convenient for architects to explore and create new floor plans which is hard to be done with conventional CAD systems.

Figure 10. Three different patterns based on the same element “Regular Star Shape”

Last but not least, as Lakoff and Johnson [18] described a common conceptual metaphor found in human beings—ARGUMENT IS WAR. With such a concept, people use the structure of war to see an argument between different parties. People use the word and ideas such as fight, indefensible, weak point, target, demolished, shoot, strategy, opponent to deal with an argument. They suggested that if there is a culture which views argument as a dance, participants as performers, and the objective is to give a pleasing performance, then the result will be more constructive.

Now, consider that if each MPM is a visual vocabulary or means for the visualization of metaphorical concepts. By choosing different visual vocabularies to represent different matters, we may create powerful resources to facilitate different activities related to communication and creative thinking. Thus, whether a user is from engineering, architecture, fashion or communication design background, or even if that person is a musician or a poet, all could find some approaches to use the model to support one's own works—by switching the mind visions through the transformations of visual metaphors. Eventually, through the building of MPM database with the metaphorical concepts of different cultures embedded, we may be able to create a universal visual language to facilitate better communication and creativity for all.

VIII. REFERENCES

- [1] W. C. Brinton, *Graphic presentation*. New York: Brinton Associates, 1939.
- [2] O. Neurath, *International picture language: the first rules of isotype*. London: Kegan Paul, Trench, Trubner & Co. Ltd., 1936.
- [3] J. Bertin, *Semiology of graphics: diagrams, networks, maps*. Madison, WI: University of Wisconsin Press, 1983.
- [4] C. J. Richards, "Diagrammatics: an investigation aimed at providing a theoretical framework for studying diagrams and for establishing a taxonomy of their fundamental modes of graphic organization," London: Royal College of Art, 1984.
- [5] Y. Engelhardt, "Syntactic structures in graphics," *Computational Visualistics and Picture Morphology*, vol. 5, no. 1, 2007, pp. 23–36.
- [6] W. Kandinsky, *Point and line to plane*, NY: Solomon R. Guggenheim Foundation, 1947.
- [7] P. Klee, *Paul Klee notebooks Volume1 the thinking eye*, 1st ed., vol. 1. London: Lund Humphries Publishers Limited, 1973.
- [8] P. Klee, *Paul Klee notebooks volume 2 the nature of nature*, 1st ed., vol. 2. London: Lund Humphries Publishers Limited, 1973.
- [9] X. Lin, "Active layout engine: Algorithms and applications in variable data printing," *CAD Comput. Aided Des.*, vol. 38, no. 5, 2006, pp. 444–456.
- [10] A. Jahanian, J. Liu, Q. Lin, D. R. Tretter, E. O. Brien-strain, S. Lee, N. Lyons, J. P. Allebach, and W. Lafayette, "Recommendation system for automatic design of magazine covers," 2013, pp. 95–105.
- [11] T. J. Morris, "A Software Automation Framework For Image-Typeface Matching In Graphic Design." M.A. thesis, Massachusetts Institute of Technology, USA, 2011.
- [12] B. J. Fry, "Computational Information Design." Ph.D. thesis, Massachusetts Institute of Technology, USA, 2004.
- [13] L. Xiong and K. Zhang, "Generation of Miro's surrealism," in *Proc. VINCI*, 2016, pp. 130–137.
- [14] W. Shi, "A generative approach to Chinese shanshui painting," *IEEE Comput. Graph. Appl.*, vol. 37, no. 1, pp. 15–19, 2017.
- [15] M.-T. Chi, C.-C. Hu, and Y.-J. Jhan, "A sketch-based generation system for oriental cloud pattern design" in *Proc. SG*, 2014, pp. 27–38.
- [16] Y. Li, K. Zhang, and D. Li, "Rule-based automatic generation of logo designs," *Leonardo*, vol. 50, no. 2, pp. 177–181, 2017.
- [17] J. Zhang, J. Yu, K. Zhang, X. S. Zheng, and J. Zhang, "Computational aesthetic evaluation of logos," *ACM Trans. Appl. Percept.*, vol. 14, no. 3, pp. 1–21, 2017.
- [18] G. Lakoff and M. Johnson, *Metaphors we live by*. Chicago and London: The University of Chicago Press, 1980.
- [19] M. Johnson, "Image-schematic bases of meaning," *Semiotic inquiry*, vol. 9, pp. 109–118, 1989.
- [20] R. Gibbs, "The psychological status of image schemas," *Cognitive linguistics research*, vol. 29, pp. 113–136, 2005.
- [21] R. Gibbs, H. Colston, "The cognitive psychological reality of image schemas and their transformation," *Cognitive Linguistics*, vol. 6, no. 4, pp. 347–378, 1995.
- [22] T. Rohrer, "Image Schemata in the Brain," *Cognitive linguistics research*, vol. 29, pp. 165–196, 2005.
- [23] J. S. Risch, "On the role of metaphor in information visualization," *Inf. Vis.*, vol. 10, no. 4, pp. 356–371, Oct. 2011.
- [24] J. Grady, "Foundations of meaning: Primary metaphors and primary scenes," Ph.D. dissertation, University of California, Berkeley, USA, 1997.
- [25] A. Szwedek, "An alternative theory of metaphorisation," *Language and meaning: Cognitive and functional perspectives*, pp. 312–327, 2007.
- [26] F. J. R. de M. Hernández and I. L. Pérez, "The Contemporary Theory of Metaphor: Myths, Developments and Challenges," *Metaphor Symb.*, vol. 26, no. 3, pp. 161–185, Jul. 2011.
- [27] R. Fusaroli and S. Morgagni, "Conceptual metaphor theory: thirty years after," *Cognitive Semiotics*, vol. 5, no. 1–2, pp. 1–12, 2013.
- [28] M. Leezenberg, "From cognitive linguistics to social science: thirty years after Metaphors We Live By," *Journal of Cognitive Semiotics*, vol. 5, no. 1–2, pp. 140–152, 2013.
- [29] L. Brandt, "Metaphor and the communicative mind," *Journal of Cognitive Semiotics*, vol. 5, no. 1–2, pp. 37–72, 2013.
- [30] A. Deignan and L. Cameron, "A re-examination of UNDERSTANDING IS SEEING," *Journal of Cognitive Semiotics*, vol. 5 no. 1–2, pp. 220–243, 2013.
- [31] R.W. Gibbs, "Making good psychology out of blending theory," *Cognitive Linguistics*, vol. 11, no. 3–4, pp. 347–358, 2000.
- [32] R.W. Gibbs, "Prototypes in dynamic meaning construal," *Cognitive Poetics in Practice*, pp. 27–40, 2003.
- [33] R.W. Gibbs and M. Tendahl, "Cognitive effort and effects in metaphor comprehension: Relevance theory and psycholinguistics," *Mind & Language*, vol. 21, no. 3, pp. 379–403, 2006.
- [34] J. Schilperoord, A. Maes, and H. Ferdinandusse, "Perceptual and conceptual visual rhetoric: The case of symmetric object alignment," *Metaphor Symbol*, vol. 24, no. 3, pp. 155–173, Jul. 2009.
- [35] R. Arnheim, *Art and Visual Perception: A Psychology of the Creative Eye* (New version, expanded and rev. ed.). Berkeley, CA: University of California Press, 1974.
- [36] M. Ortiz, "Visual Rhetoric: Primary Metaphors and Symmetric Object Alignment," *Metaphor Symb.*, vol. 25, no. 3, pp. 162–180, Jul. 2010.
- [37] Y.-S. Jung, *The Best in World Trademarks: Corporate Identity, Brand Identity, Volume 1-2*, Korea: Millim, 2005.
- [38] *New logo: a collection of corporate identities*. Singapore: Page One, 2004.
- [39] C. L. Larson, J. Aronoff, I. C. Sarinopoulos, and D. C. Zhu, "Recognizing threat: A simple geometric shape activates neural circuitry for threat detection," *J. Cogn. Neurosci.*, vol. 21, no. 8, pp. 1523–1535, 2009.
- [40] J. Ibáñez and C. Delgado-Mata, "Emotional line: Showing emotions through the sharpness, height, width and speed of a series of concatenated digital shifting curves," in *Proc. SG*, 2014, vol. 8698 LNCS, pp. 98–111.
- [41] O. Blazhenkova and M. M. Kumar, "Angular versus curved shapes: correspondences and emotional processing," *Perception*, vol. 47, no. 1, pp. 67–89, 2018.
- [42] A. Carlier, K. Leonard, S. Hahmann, G. Morin, and M. Collins, "The 2D shape structure dataset: A user annotated open access database," *Comput. Graph.*, vol. 58, pp. 23–30, 2016.
- [43] E. J. Golin and S. P. Reiss, "The specification of visual language syntax," *J. Vis. Lang. Comput.*, vol. 1, no. 2, pp. 141–157, 1990.
- [44] M. J. Egenhofer and M. P. Dube, "Topological relations from metric refinements," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. - GIS*, 2009, p. 158.
- [45] J. M. Mandler, "The spatial foundations of the conceptual system," *Lang. Cogn.*, vol. 2, no. 1, pp. 21–44, 2010.
- [46] J. Durand, "Rhetorical figures in the advertising image," in *Marketing and semiotics: new directions in the study of signs for sale*, J. Umiker-Sebeok, Ed. New York: Mouton, 1987, pp. 295–318.
- [47] G. Baruchello, "A classification of classics. Gestalt psychology and the tropes of rhetoric," *New Ideas Psychol.*, vol. 36, pp. 10–24, 2015.
- [48] B. A. Huhmann and P. A. Albinsson, "Assessing the Usefulness of Taxonomies of Visual Rhetorical Figures," *J. Curr. Issues Res. Advert.*, pp. 1–25, 2018.

Approach to Context Computation for Implicit Context-sensitive Graph Grammars

Yang Zou, Xiaoqin Zeng, Yufeng Liu, and Huiyi Liu
Institute of Intelligence Science and Technology
Hohai University
Nanjing, P. R. China

Abstract—Context-sensitive graph grammars are appropriate formalisms for specifying visual programming languages, as they are intuitive, rigorous, and expressive. Nevertheless, some of the formalisms whose contexts are implicitly or incompletely represented in productions, called implicit context-sensitive graph grammars, suffer inherent weakness in intuitiveness or limitations in parsing efficiency. Context tends to be a conceivable way to address this issue. Based on the formalization of context, this paper proposes an approach to the computation of context for implicit context-sensitive graph grammars. Moreover, the complexity of the four algorithms comprising the approach is analyzed and the applicability of the approach is discussed. The proposed approach paves the way for the applications of context in implicit graph grammar formalisms, such as facilitating comprehension of graph grammars and improving efficiency of general parsing algorithms.

Keywords - visual languages; implicit context-sensitive graph grammars; context computation; algorithm; complexity

I. INTRODUCTION

Visual Programming Languages (VPLs), which usually handle objects that do not possess inherent visual representation [1], have been frequently utilized in many fields of computer science including software engineering. Various approaches have been proposed to formally specify and parsing VPLs [2-4]. As a natural extension of formal grammar theory, graph grammars offer the mechanisms for formal specification and parsing of VPLs [5], just like formal grammars do for string languages. However, the extension from one-dimensional string-based formal grammars to two-dimensional graph grammars brings about a few novel challenges, especially the embedding problem. Quite a few graph grammar formalisms have been proposed in the literature [5-9]. From the perspective of usability, there is still room for these formalisms to be ameliorated in expressive power or computing efficiency.

Most existing graph grammar formalisms are context-free or context-sensitive. The expressive power of a graph grammar lies on the type it belongs to as well as the embedding mechanism it chooses [10-11]. Among all the categories of embedding mechanisms that vary in complexity and power, invariant embedding is the least complex one and most commonly employed in graph grammar formalisms. Context-sensitive graph grammars tend to be more expressive than context-free ones, when confined to identical less complex embedding mechanisms and invariant embedding in particular. As context-free graph grammars have difficulty in specifying a

large portion of graphical VPLs [8-9], recent research in this field focus more on context-sensitive graph grammars [12-15].

A graph grammar consists of a set of productions, each of which is a pair of graphs, called left graph and right graph, together with an embedding expression. In context-sensitive graph grammars, the contexts pertaining to a production generally refer to the neighboring subgraphs of the rewritten portion of its left graph in potential host graphs [16], which describe the situations under which the production can be applied. A host graph is a graph that is being rewritten by some graph grammar in the process of derivation or parsing. However, the context portion of a production, i.e., the remainder of the left graph minus the rewritten portion, is commonly not a direct copy of the contexts for the sake of conciseness of productions and easiness of embedding.

The most representative context-sensitive graph grammar formalisms are Layered Graph Grammar (LGG) [8] and Reserved Graph Grammar (RGG) [9]. To solve the embedding problem, LGG identically involves in the left and right graphs of a production its immediate context and imposing a dangling edge condition on redex definition, which guarantees that dangling edges never occur in rewritten host graphs. Generally, a redex is a subgraph in a host graph that is isomorphic to the left or right graph of a production. RGG is commonly viewed as an improvement over LGG in respect of succinctness of specification and efficiency of parsing algorithm. Rather than directly involving contexts in productions just as LGG does, RGG formalism invents a particular two-level node structure coupled with a marking technique to indirectly specify the context of a production by identically distributing a set of marked vertices into the left and right graphs. The vertices establish a one-to-one correspondence between the two graphs in terms of their marks. Thus, the embedding problem is solved through this mechanism together with a dedicated embedding rule. Other context-sensitive formalisms include Edge-based Graph Grammar (EGG) [17-18], Context-Attributed Graph grammar (CAGG) [19], Contextual Layered Graph Grammar (CLGG) [20], and Spatial Graph Grammars (SGG) [12]. To tackle the embedding problem, EGG identically augment a set of marked dangling edges to both the left and right graphs of a production, whereas CAGG introduces attributes of nodes to establish a correspondence between the two graphs of a production. CLGG and SGG are extensions of LGG and RGG, respectively. Based on LGG, CLGG supports three extra

mechanisms, which can be employed to define more complex VPLs. SGG extends RGG by augmenting its productions with a spatial specification mechanism, with which it can explicitly describe both structural and spatial relationships for VPLs.

In terms of how the context portion of a production is dealt with, the preceding formalisms can be classified into two categories: explicit and implicit [21]. The former are those that directly enclose the complete immediate contexts as its context portion in a production, whereas the latter refers to the ones in which the context portion is expressed as specifically tailored (i.e., incomplete) immediate contexts, newly introduced notations or attributes adhered to the rewritten portion. Readily, LGG and RGG are the typical examples of the former and the latter, respectively.

An inherent weakness of implicit formalisms is that they are not intuitive, which arises from the fact that the context portion of a production is not the complete immediate contexts. In RGG, the context portion is a set of marked vertices. Vertices are explained to be connecting points of edges, but their exact meaning is left undefined. Therefore, the selection, arrangement and marking of vertices within a node become a challenge in the design of productions. Similar situations arise in other implicit formalisms. Moreover, as actual immediate contexts are absent in productions, it is rather difficult for users to exactly comprehend the language of a given graph grammar.

The exploitation of context is a conceivable way to address the above issues. Obviously, context is useful in several situations. Context can be used to make up the deficiency in intuitiveness so as to facilitate the comprehension and design of implicit graph grammars. Furthermore, context can be utilized to reduce backtracking in general parsing algorithms by context matching, thus improving the parsing efficiency. In the literature [21], a formal definition of context is proposed and the properties are characterized, which provide a solid theoretical foundation for the computation of context. Nevertheless, due to the complexity of context formalization, an explicit method for the computation of context is apparently a necessity for the purpose of practical usage.

To this end, on the basis of RGG formalism, an approach for the computation of context for implicit graph grammars is proposed in this paper. This is a subsequent research on context, and the technical contributions are as follows: It proposes a concrete approach for context computation, which consists of four partially ordered algorithms with each one being dependent on its predecessors. Moreover, it provides the time complexities of the algorithms and discusses the applicability of the approach. Besides, this method can be generalized to other implicit graph grammar formalisms. Therefore, it paves the way for the application of context in implicit graph grammar formalisms.

The remainder of the paper is organized as follows: Section 2 reviews the RGG formalism and excerpts the formal definition of context. Section 3 proposes an approach that consists of four algorithms to the computation of context. Section 4 addresses the complexities of the algorithms. Section 5 discusses the applicability of the algorithms. Finally, section 6 concludes the paper and proposes future research.

II. PRELIMINARIES

A. The RGG Formalism

A graph grammar consists of an initial graph and a collection of productions (graph rewriting rules). Each production has two graphs called left graph and right graph respectively, and can be applied to another graph called host graph. Every node in a production is either a terminal or a non-terminal node. A graph grammar defines a graph language composed of those graphs that can be derived from the initial graph by repeated applications of the productions and whose nodes are all terminal ones. A redex is a subgraph in a host graph that is isomorphic to the left or right graph of a production.

Figure 1. A graph grammar for process flow diagrams.

RGG is a context-sensitive graph grammar formalism [9]. It introduces a node-edge format to represent graphs in which each node is organized as a two-level structure, where the large surrounding rectangle is the first level, called super vertex, and other embedded small rectangles are the second level, called vertices. Either a vertex or a super vertex can be the connecting point of an edge. In addition to the two-level node structure, the RGG also introduces a marking technique that divides vertices into two categories: marked and unmarked ones. Each marked vertex of a production is identified by an integer that is unique in the left or right graph where the vertex lies. A production is properly marked if each marked vertex in the left graph has a counterpart marked by the same integer in the right graph, and vice versa. In the process of a production application, when a redex is matched in a host graph, each vertex that corresponds to a marked vertex in the left or right graph preserves its associated edges connected to nodes outside of the redex, which avoids the appearance of dangling edges during the subsequent subgraph replacement provided that an additional embedding rule is also enforced. The embedding rule states that if a vertex in the right (or left) graph of a production is unmarked and has an isomorphic vertex in the redex of a host graph, then all the edges connected to the vertex should be completely inside the redex.

As an example, an RGG specifying process flow diagrams, which is slightly adapted from [8], is depicted in Figure 1.

B. Partial and Total Precedence

In this subsection, we take the RGG as the representative of implicit context-sensitive graph grammar formalisms to present partial and total precedence relations between graph productions. For the sake of clarity and simplicity, some basic concepts and notations are listed below. Note that graphs are directed ones in the node-edge format and only vertices in productions might be marked.

RGG: A reserved graph grammar is a triple (A, P, Ω) , where A is an initial graph, P a set of graph grammar productions, Ω a finite label set consisting of two disjoint sets Ω^T and Ω^{NT} (called terminal label set and nonterminal label set, respectively). For any production $p := (L, R) \in P$, three conditions are satisfied: R is non-empty, L and R are both over Ω , and the size of R are not less than that of L .

$p := (L, R)$: A production with a pair of marked graphs: the left graph L and right graph R . The notations $p.L$ and $p.R$ represent the left and right graph of a production p , respectively. For any graph G , $G.N$ and $G.E$ denote the set of nodes and edges, respectively; $n.V$ and $n.v$ denote the set of vertices and some vertex v of a node n , respectively; and $G.V = \bigcup_{n \in G.N} n.V$ is the union of the sets of vertices of nodes in G ; for any edge e , $s(e)$ and $t(e)$ represent the source and target vertex of e , respectively, and $l(e)$ is the label on e . $P_L = \{p.L | p \in P\}$ and $P_R = \{p.R | p \in P\}$.

$G_1 \approx G_2$: G_1 is isomorphic to G_2 .

Redex: A subgraph $X \subseteq H$ is a redex of graph G , denoted as $X \in Rd(H, G)$, if $X \approx G$ under an isomorphic mapping f and any vertex in X that is isomorphic to an unmarked vertex in G keeps its edges completely inside X .

$Rd(H, G)$: A set of redexes of marked graph G , which are subgraphs of graph H .

Merger: A graph G is a merger of G_1 and G_2 , if both G_1 and G_2 are subgraphs of G and each node or edge in G is either from G_1 or from G_2 . $Mrg(G_1, G_2)$ is the set of mergers of G_1 and G_2 .

Mcc: A mapping from graphs to the sets of maximal connected components contained in these graphs. A maximal connected component in a graph is a connected component being maximal. $Mcc(P_L) = \{C | C \in Mcc(p.L) \wedge p \in P\}$, $Mcc(P_R) = \{C | C \in Mcc(p.R) \wedge p \in P\}$.

The following definitions excerpted from [21] are necessary to understand the notion of context and the approach to context computation.

Definition 1. Let $gg := (A, P, \Omega)$ be an RGG, $p_1, p_2 \in P$ be two productions, $C_1 \in Mcc(p_1.L)$ and $C_2 \in Mcc(p_2.R)$. If $\exists X \subseteq C_2$ such that $X \in Rd(C_2, C_1)$, then C_1 is matched with X in C_2 , denoted as $C_1 \approx X \sqsubseteq C_2$; or concisely C_1 is included in C_2 , denoted by $C_1 \sqsubseteq C_2$.

This definition introduces the notion of inclusion between the components of productions, or to be exact, to locate a redex

of a component of the left graph of one production in some component of the right graph of another production.

Let U be some set and $S = (B, m)$ a multiset, where B is the underlying set of elements and $m: B \rightarrow \mathbb{N}$ is a mapping from B to the set $\mathbb{N}$ of positive natural numbers. $S \subseteq^{\mathbb{N}} U$ if and only if $B \subseteq U$. In order to unambiguously reference to an element from a multiset, we stipulate that any two elements in a multiset S have distinct identities even if they are the same element from the point of view of the underlying set B , and the identities of elements are not explicitly represented in context for the sake of conciseness.

Definition 2. Let $gg := (A, P, \Omega)$ be an RGG, and P_L and P_R the sets of left and right graphs of productions in P , respectively. A set $S_1 \subseteq Mcc(P_L)$ is included in another multiset $S_2 \subseteq^{\mathbb{N}} Mcc(P_R)$, denoted as $S_1 \sqsubseteq S_2$, if there is a mapping $f: S_1 \rightarrow S_2$ such that:

- $\forall C \in S_1 (\exists X \subseteq f(C) (X \in Rd(f(C), C)))$, and
- $\forall C, C' \in S_1 (C \neq C' \wedge f(C) = f(C') \rightarrow \exists X, X' \subseteq f(C) (X \in Rd(f(C), C) \wedge X' \in Rd(f(C), C') \wedge X \cap X' = \emptyset))$.

The first condition states that for each component in S_1 , there is an image in S_2 under the mapping f that contains a redex of it; and the second expresses that if two different components in S_1 have the same image in S_2 , then the two corresponding redexes in it cannot overlap, which adheres to the redex definition in the RGG formalism.

Definition 3. Let $gg := (A, P, \Omega)$ be an RGG, and $p_1, p_2 \in P$ be two productions, p_1 directly partially precedes p_2 , denoted as $p_1 \preceq_d p_2$, if $\exists S \subseteq Mcc(p_2.L)$ such that $S \sqsubseteq Mcc(p_1.R)$. The direct partial precedence relation between them is denoted by the pair $\langle p_1, p_2 \rangle$. The direct partial precedence relation on the set P of productions is defined as $\preceq_P = \{\langle p_1, p_2 \rangle | p_1, p_2 \in P \wedge p_1 \preceq_d p_2\}$. The partial precedence relation between them is denoted by the pair $\langle p_1, p_2 \rangle$.

The partial precedence relation is the closure of the direct partial precedence relation on a set P . Partial precedence is a kind of relation between a pair of components chosen from two distinct productions, whereas total precedence describes the same relation between two sets of components from a subset of productions and a single production respectively.

Definition 4. Let $gg := (A, P, \Omega)$ be an RGG, $p \in P$, and a multiset $P' \subseteq^{\mathbb{N}} P$. P' directly totally precedes p , denoted as $P' \prec_d p$, if there is a surjective mapping $f: Mcc(p.L) \rightarrow St$ such that:

- $St \subseteq Mcc(P'_R)$;
- $Mcc(p.L) \sqsubseteq Mcc(P'_R)$ with respect to f ;
- $\forall p' \in P' (\exists C \in Mcc(p.L) (f(C) \in Mcc(p'.R)))$.

The corresponding direct total precedence relation is denoted by the pair $\langle P', p \rangle$, and p and P' are called the target production and preceding set, respectively. The direct total precedence relations on the set P of productions is defined as $\prec_P = \{\langle P', p \rangle | P' \subseteq^{\mathbb{N}} P \wedge p \in P \wedge P' \prec_d p\}$.

A direct total precedence relation specifies that a certain graph composed of the right graphs of a subset of productions

contains a redex of the left graph of another production. Note that the third constraint on f emphasizes that every production in P' takes part in f with at least one of its components in the right graph. If a subset P' of P forms such a relation with a production p , then it means that all the right graphs of P' must exactly comprise a redex of the left graph of p with each one containing at least one redex of its components.

A total precedence relation $\langle M, p \rangle$ is composed of a set of direct total precedence relations and a set of linking relations on it. A compound precedence set consists of three parts: a multiset T of productions from set P , a multiset E of direct total precedence relations, and a set R of linking relations on E . A compound precedence set M , together with a production p , forms a total precedence relation, on condition that a direct total precedence relation is established between the first part of M and the production p .

C. Definition of Context

The sets of partial or total precedence relations with respect to a graph grammar establish an order of production applications, which can be exploited to discover potential situations in which any of the productions is applicable for derivation. We refer to these situations as contexts. Given two productions p_1 and p_2 , if p_1 directly partially precedes p_2 , then $p_1.R$ contains a context of p_2 or merely a portion of a context, depending on whether $p_2.L$ consists of only one or at least two maximal connected components. As for the former case, $\{p_1\} <_d p_2$ readily holds and a context of p_2 immediately follows; whereas in the latter case, a subset of productions involving p_1 that directly totally precedes p_2 is pursued so as to form a complete context for p_2 . As a third case, a total precedence relation can be sought to build a rather deeper complete context.

Complete contexts of a production can be stratified in terms of the levels of corresponding total precedence relations from which they are generated. Roughly, a complete context that is built in the light of a precedence relation that corresponds to a rooted tree of depth i is called a level- i context, $i \geq 1$, and it degrades to a level-1 context when the relation is a direct one.

A complete context can be employed to extend the respective production to which it pertains. This is done by augmenting the context simultaneously to its both graphs and properly linking the two parts together respectively. A production p equipped with a level- i context is called a context- i p .

Definition 5. Let $gg := (A, P, \Omega)$ be an RGG, $p \in P$, $Mcc(p.L) = \{C_1, \dots, C_n\}$, and $P' \subseteq^N P$. If $P' <_d p$ with respect to some surjective mapping $f: Mcc(p.L) \rightarrow St = \{D_1, \dots, D_m\}$ and a set of redexes $X = \{X_i | X_i \in Rd(f(C_i), C_i), 1 \leq i \leq n\}$, then the pair (U, Z) is a level-1 context of p with respect to P' , f , and X , denoted as $ct_p(P', f, X)$, where $U = \bigcup_{1 \leq j \leq m} D_j'$, $D_j' = D_j \setminus \bigcup_{k_j \in K_j} X_{k_j}$, $K_j = \{l | f(C_l) = D_j \wedge 1 \leq l \leq n\}$, $Z = \bigcup_{1 \leq j \leq m} Z_j$, and $Z_j = \{e \in D_j.E | (s(e) \in X_{k_j} \vee t(e) \in D_j'.V) \vee (s(e) \in D_j'.V \wedge t(e) \in X_{k_j}.V) \wedge k_j \in K_j\}$. The sets U and Z are called the contextual graph and contextual connection, respectively.

Each component D_j' of the contextual graph is the rest graph of D_j , a component of the right graph of some production that

contains one or more redexes X_{k_j} of the components C_{k_j} , minus X_{k_j} , and each Z_j is the collection of edges in D_j that connect D_j' to all the redexes X_{k_j} .

Similar to Definition 5, the notion of level- i context can be recursively defined.

Figure 2. The contexts of a production and their extended productions. (a) A level-1 context of p_9 . (b) A level-2 context of p_8 . (c) A level-1 context equipped p_9 . (d) A level-2 context equipped p_8 .

Example 1. Two contexts of a production and their extended productions.

Two contexts at distinct levels of a production and their respective extended productions are demonstrated in Figure 2. A level-1 context of p_9 originated from the direct total precedence relation $\{p_2, p_9\} <_d p_9$ is shown in Figure 2(a), where the left component of the graph is R_2 (the right graph of production 2), the right one is R_9 , the subgraph enclosed by the green dashed ellipse is the redex of L_9 (the left graph of p_9), and the context consists of two parts U and Z : U is the rest of the whole graph minus the redex and Z the set of thick red edges that connect U to the redex. The respective context-1 p_9 , numbered as p_{10} , is depicted in (c), in which the two subgraphs surrounded by dashed green eclipses are the isomorphic image of the underlying production of p_{10} .

Figure 2(b) shows a level-2 context of p_8 , which is created on the basis of the direct total precedence relation $Ud(\{p_4, p_{10}\}) <_d p_8$, or equivalently, the total precedence relation $U(\{p_4, p_{10}\}) < p_8$. In this graph, the left and right component is R_{10} and R_4 , respectively, the subgraph enclosed by the purple dashed rectangle is the isomorphic image of $Ud(p_{10})$, and the one surrounded by the green dashed eclipse is

the redex of the left graph of $p8$. Similarly, the corresponding context-2 $p8$, numbered as $p11$, is illustrated in (d).

III. CONTEXT COMPUTATION

In this section, an approach is presented for the computation of contexts in the RGG formalism based on the theoretical foundation reviewed in the preceding section.

The approach consists of four partially ordered algorithms, of which the first three deal with the computation of the set of direct partial precedence relations, the set of direct total precedence relations, and the set of total precedence relations with respect to a set of productions, respectively, and the last handles the computation of the contexts of a single production as well as the corresponding extended productions.

A. Computation of Partial Precedence

Algorithm 1. Computation of partial precedence relations.

Input. A set P of productions.

Output. The direct partial precedence relation on P .

```

{
   $Cm_L = \bigcup_{p \in P} Mcc(p.L);$ 
   $Cm_R = \bigcup_{p \in P} Mcc(p.R);$ 
  Create a two-dimensional array  $M$  whose two indices range over  $Cm_L$ 
  and  $Cm_R$  respectively such that each element is initialized to an
  empty set  $\emptyset$ ;
  for each  $C \in Cm_L$  {
    for each  $C' \in Cm_R$  {
       $M[C, C'] = \text{FindRedex}(C', C);$ 
    }
  }
  return  $M$ ;
}
```

The first algorithm generates the partial direct precedence relation on a given set of productions. It consists of collecting all the components of the left and right graphs of the productions to create one set and another respectively, and then finding all the redexes of each component of the former in any one of the latter that is taken as the host graph. The output is a matrix such that each entry is assigned to a set of redexes (maybe empty) with respect to a pair of components from the two distinct sets that uniquely locate the entry in it.

B. Computation of Direct Total Precedence

The second algorithm figures out the direct total precedence relation on a set P of productions, on the basis of the output of the first algorithm. First, it creates three one-dimensional arrays Llt , Rlt , Tpd and Fun which share a same index that ranges over P , to store the upcoming data. Then, it arranges the components of the left graph of each production p into an ordered tuple form $Llt[p]$ in a certain order, and for each element in this tuple, it generates a corresponding set that gathers all the redexes involved in the components from Cm_R . Next, it conducts the Cartesian product $Rlt[p]$ of these sets of redexes in the same order as their counterparts in $Llt[p]$. As a result, every tuple in $Rlt[p]$ is a redex of $Llt[p]$, since they are of the same length and each constituent of the former is a redex of the element of the latter to which it corresponds in terms of the tuple order. After that, by replacing each redex in a tuple of $Rlt[p]$ with the underlying production whose right graph includes a

component that contains this redex, it acquires a direct total precedence relation regarding p ; this process repeats until all the relations regarding p , which comprise the set $Dtp[p]$, are collected. Meanwhile, it establishes a mapping h from $Dtp[p]$ to a partition of $Rlt[p]$ (i.e., a collection of disjoint nonempty subset of it that have it as their union) such that h maps each tuple to the subset of $Rlt[p]$, each of whose elements can be transformed to this tuple by the preceding replacement. Note that each element in $Dtp[p]$ is a list of a multiset of productions in a predefined order. Nevertheless, a direct total precedence relation refers to a relation between a multiset of productions without any order and a production. To bridge this gap, it performs a partition of $Dtp[p]$ in terms of such an equivalence relation that two lists are equivalent if both of them correspond to a same multiset, which produces $Dps[p]$ and l . In this way, it finally achieves the arrays Dps and $Fun2$ that can generate the direct total precedence relations regarding each production of P , together with the array $Fun1$ that can produce the set of redexes with respect to any of these relations. For example, for each $S \in Dps[p]$, $\langle S, p \rangle$ is a direct total precedence relation, and $\bigcup_{Q \in l(S)} h(Q)$ is the set of all the possible redexes.

Algorithm 2. Computation of direct total precedence relations.

Input. A set P of productions and the direct partial precedence relation on P .

Output. The direct total precedence relation on P .

```

{
  Let  $Llt$ ,  $Rlt$ ,  $Dtp$ ,  $Dps$ ,  $Fun1$  and  $Fun2$  be one-dimensional arrays
  that share the same index which ranges over  $P$ ;
  for each  $p \in P$  {
     $k = |Mcc(p.L)|;$ 
    Let  $Mcc(p.L) = \{C_1, \dots, C_k\};$ 
     $Llt[p] = (C_1, \dots, C_k);$ 
    for each  $C \in Mcc(p.L)$  {
       $Rdx(C) = \emptyset;$ 
      for each  $C' \in Cm_R$  {
         $Rdx(C) = Rdx(C) \cup M[C, C'];$ 
      }
    }
     $Rlt[p] = Rdx(C_1) \times \dots \times Rdx(C_k);$ 
     $Dtp[p] = \emptyset;$ 
    for each  $k$ -tuple  $(t_1, \dots, t_k) \in Rlt[p]$  {
      Generate a  $k$ -tuple  $(p_1, \dots, p_k)$  such that
       $t_i \in \text{FindRedex}(Mcc(p_i.R), C_i), 1 \leq i \leq k;$ 
      if  $((p_1, \dots, p_k) \notin Dtp[p])$  {
         $h((p_1, \dots, p_k)) = \{(t_1, \dots, t_k)\};$ 
         $Dtp[p] = Dtp[p] \cup \{(p_1, \dots, p_k)\};$ 
      } else
         $h((p_1, \dots, p_k)) = h((p_1, \dots, p_k)) \cup \{(t_1, \dots, t_k)\};$ 
    }
     $Fun1[p] = h;$ 
     $Dps[p] = \emptyset;$ 
    for each list  $(q_1, \dots, q_k) \in Dtp[p]$  {
      Create a multiset  $S = \{q_1, \dots, q_k\};$ 
      if  $(S \notin Dps[p])$  {
         $l(S) = \{(q_1, \dots, q_k)\};$ 
         $Dps[p] = Dps[p] \cup \{S\};$ 
      } else
         $l(S) = l(S) \cup \{(q_1, \dots, q_k)\};$ 
    }
     $Fun2[p] = l;$ 
  }
  return  $(Llt, Rlt, Dtp, Dps, Fun1, Fun2);$ 
}
```

An underlying assumption for the algorithm is that any component in Cm_L or Cm_R , as well as any redex of a component

from Cm_L in another from Cm_R , is uniquely identified. This is applicable from the perspective of algorithm implementation, as it can be achieved by assigning to each node of a production a unique number as its identity, and representing each redex as a triple (S, f, id) with S the involved subgraph, i.e., the redex itself, f the underlying mapping, and id the identifier of the right graph that contains the redex.

C. Computation of Total Precedence

The third algorithm describes the procedure of constructing all the total precedence relations of depth $k + 1$ based on those of depth no more than k , with respect to a set P of productions, where $k \geq 1$. Readily, the set of total precedence relations of any depth can be recursively generated from the fundamental set of direct total precedence relations by using it.

Algorithm 3. Computation of total precedence relations.

Input. A set P of productions, $\leq_p$ the direct total precedence relation on P , and the set Tpd of total precedence relations of depth no more than k , where $k \geq 1$.

Output. The set of total precedence relations of depth $k + 1$.

```

{
  Let  $Ptp$  and  $Tpr$  be one-dimensional arrays whose indexes range
  over  $P$ ;
  Let  $Tpd_k$  be the set of total precedence relations of depth  $k$ ;
  for each  $p \in P$  {
     $Ptp[p] = \{null\}$ ; // Initialize the elements of  $Ptp$ ;
  }
  for each  $ps \in Tpd$  {
    Let  $e = \langle P', p' \rangle$  be the root element of  $ps$ ;
     $Ptp[p'] = Ptp[p'] \cup \{ps\}$ ;
  }
  for each  $p \in P$  {
     $Ptr[p] = \emptyset$ ;
    for each  $e' = \langle P'', p' \rangle \in \leq_p$  {
      Let  $P'' = \{p_1, \dots, p_k\}$ ; //  $P''$  is a multiset;
       $Crt = Ptp(p_1) \times \dots \times Ptp(p_k)$ ;
       $E = \{e'\}$ ;
       $R = \emptyset$ ;
      for each  $(t_1, \dots, t_k) \in Crt$  such that
       $\exists i, j (t_i \neq null \wedge t_j \in Tpd_k)$  //  $1 \leq i \leq k$ ;
        Create a mapping  $f$  with domain  $P''$ ;
         $D = \emptyset$ ; //  $D$  is a multiset;
        for each  $t_i$  //  $1 \leq i \leq k$ ;
          if  $(t_i = null)$ 
             $f(p_i) = null$ ;
          else {
            Let  $t_i = (E_i, R_i)$ , and  $e_i$  be the root element;
             $f(p_i) = e_i$ ;
             $D = D \cup \{e_i\}$ ;
             $E = E \cup E_i$ ;
             $R = R \cup R_i$ ;
          }
        }
        Construct a linking relation  $r = (e', D, f)$ ;
         $R = R \cup \{r\}$ ;
         $Ptr[p''] = Ptr[p''] \cup \{(E, R)\}$ ;
      }
    }
  }
  return  $Tpr$ ;
}

```

The algorithm consists of two tasks. First, it partitions the set Tpd into $|P|$ distinct subsets in terms of the root node of the rooted tree that each total precedence relation (i.e., a precedence structure) forms, which comprise the set Ptp .

Algorithm 4. Computation of contexts and extended productions.

Input. A set P of productions, a total precedence relation $\langle M, p_0 \rangle = (E_0, R_0)$ of depth h .

Output. The level- h contexts of p_0 and corresponding extended productions.

```

{
   $Cps = \{(E_0, R_0)\}$ ;
  while  $(\exists ps \in Cps$  such that  $ps$  is not a direct total precedence
  relation) {
    Let  $ps = (E, R)$ ;
    Find an element  $e \in E$  such that  $e$  is not the head of any linking
    relation in  $R$ ;
    Let  $e = \langle P', p \rangle$  and  $P' = \{p_1, \dots, p_k\}$ ; //  $k \geq 1$ ;
    Let  $r = (e', D, f) \in R$  such that  $e \in D$ ;
    Let  $e' = \langle P'', p'' \rangle$ ;
    Let  $Fun1[p] = h$ ,  $Fun2[p] = l$ ;
    Let  $l(P') = \{Q_1, \dots, Q_m\}$ ; //  $m \geq 1$ ;
     $Cps = Cps \setminus \{ps\}$ ;
    for each  $(t_1, \dots, t_k) \in h(Q_1) \cup \dots \cup h(Q_m)$  {
      Construct a level- $i$  context  $(U, Z)$ ,  $i \geq 1$ ;
      Construct a context- $i$   $p$  with  $(U, Z)$ , called  $p'$ ;
       $P''' = (P'' \setminus \{p\}) \cup \{p'\}$ ;
       $e'' = \langle P''', p'' \rangle$ ;
       $D' = D \setminus \{e\}$ ;
      Create a linking mapping  $f': P''' \rightarrow D' \cup \{null\}$  such that
       $f'(p') = null$ , and for any other production  $p''' \in P'''$ ,
       $f'(p''') = f(p''')$ ;
       $r' = (e'', D', f')$ ;
       $E' = (E \setminus \{e, e'\}) \cup \{e''\}$ ;
       $R' = R \setminus \{r\} \cup \{r'\}$ ;
      Construct a precedence structure  $ps' = (E', R')$ ;
       $Cps = Cps \cup \{ps'\}$ ;
    }
     $Cps = Cps \setminus \{ps\}$ ;
  }
   $Cnt = \emptyset$ ;
   $Xdp = \emptyset$ ;
  for each  $ps \in Cps$  {
    Construct a level- $h$  context  $(U, Z)$ ;
     $Cnt = Cnt \cup \{(U, Z)\}$ ;
    Construct a context- $h$   $p_0$  with  $(U, Z)$ , called  $q$ ;
     $Xdp = Xdp \cup \{q\}$ ;
  }
  return  $(Cnt, Xdp)$ ;
}

```

Then, for any production p in P , it constructs the set of all the total precedence relations of depth $k + 1$ whose root elements share the same head p . More precisely, this set is the union of the subsets, each of which includes those relations whose root elements are a same direct total precedence relation with p as the head. Thus, in terms of each of these relations, say $\langle P'', p \rangle$, the algorithm constructs a corresponding subset of all the relations of depth $k + 1$ with $\langle P'', p \rangle$ as the root element. To this end, it first conducts the Cartesian product of $|P''|$ selected elements from Ptp whose indexes exactly comprises the multiset P'' . Second, it screens the ordered tuples in the product to make sure that each chosen one can be utilized to generate a proper linking mapping from P'' to the union of $\{null\}$ and the set of root elements of its constituents, and that the resulting precedence structure must be of depth $k + 1$. The latter is guaranteed when one constituent of the tuple is of depth k . Third, for each chosen tuple, it generates the linking relation, which is composed of the head $\langle P'', p \rangle$, the set of tails, i.e., the root elements of the constituents of the tuple, and the newly created linking mapping, and then constructs the consequent precedence structure (E, R) , where E and R comprise all the involved direct

total precedence relations and relevant linking relations, respectively.

D. Computation of Context

The fourth algorithm calculates all the level- i contexts of a production and respective extended productions in terms of a total precedence relation, i.e., a precedence structure, with respect to it. It is known that a total precedence relation, i.e., a precedence structure (E, R) , forms a rooted tree in such a way that each direct total precedence relation in E corresponds to a subtree of depth 1 and they are glued to each other in terms of the linking relations in R . The rooted tree structure offers a more easily comprehensible perspective of the algorithm.

The algorithm is composed of two consecutive tasks. The first task is to transform a total precedence relation (a rooted tree) into a set of direct total precedence relations (rooted trees of depth 1). To this end, the algorithm creates an empty set, adds the rooted tree to it, and repeats the subsequent four steps until all the elements in it become rooted trees of depth 1. First, it randomly selects a rooted tree from the set and locates an outmost subtree of length 1 (with the root node, say p); then, it figures out all the possible context- i p 's according to the subtree, where $i \geq 1$; next, for each of those extended productions, say p' , it constructs a new tree from the original one by pruning the subtree except the root (this node is preserved since it is also a leaf of another subtree to which this one is linked via a linking relation in R) from it and substituting p' for the root p , and puts it into the set; and finally, it deletes the originally selected tree from the set. After that, the set includes only rooted trees of depth 1, any of which corresponds to a direct total precedence relation. In the second task, it constructs, for each element in the set, a level- h context and an accompanying extended production as well.

IV. COMPLEXITY ANALYSIS

In this section, the complexities of the proposed algorithms in the preceding section are analyzed.

Algorithm 1 calls a procedure FindRedex(H, G) to generate the set of subgraphs of H that are redexes of the marked graph G . A procedure similar to the callee was proposed in [9] with time complexity $O(|H|^{|\mathcal{G}|})$, where $|H|$ or $|G|$ denotes the number of nodes involved in it. Then, the time complexity of the algorithm directly follows:

Proposition 1. *The time complexity of Algorithm 1 is $O(m^2 n^2 r^r)$, where m is the number of productions in P , n is the maximal number of components in the left or right graphs of productions in P , and r is the maximal number of nodes in any of the components.*

Proposition 2. *The time complexity of Algorithm 2 is $O((mn)^{n+1} r^{rn})$, where m , n and r are as in Proposition 1.*

Proof. The algorithm mainly consists of a for-loop that nests other three sequential for-loops. In the outmost loop, the first nested for-loop also nests another for-loop, which takes time $O(mn^2)$. The line next to it is the calculation of a k -ary Cartesian product over sets of redexes of components in $p.L$, each of which contains at most $mn \times r^r$ elements, where mn and r^r are the maximal number of elements in Cm_R and $M[C, C']$,

respectively. Thus, the time complexity is $O((mn r^r)^n)$, since k equals to n in the worst case. In the second nested for-loop, the first line takes $O(n)$, because the production to which each redex in the ordered tuple of Rlt corresponds is clearly indicated beforehand, according to the underlying assumption. So, this for-loop takes $O(n(mn r^r)^n)$. Moreover, the time complexity of the last nested for-loop is at most $O(n(mn r^r)^n)$, for the cardinality of $Dtp[p]$ is no more than that of $Rlt[p]$. In summary, the nested part of the outmost loop takes time $O(n(mn r^r)^n)$. Consequently, the time complexity of the whole procedure is $O((mn)^{n+1} r^{rn})$, which is the product of the time complexity of the outmost for-loop and that of its nested part. ■

Proposition 3. *The time complexity of Algorithm 3 is $O(m^{n+2} n l^n)$, where m and n are as in Proposition 1, and l is the cardinality of Tpd .*

Proof. The algorithm consists of three sequential for-loops. It is evident that the first two loops take $O(n)$ and $O(l)$, respectively. The last one is a four-layer nested for-loops. According to the structure, its time complexity can be expressed as $O(d_1 d_2 d_3 d_4)$, where d_1, d_2, d_3 and d_4 are the maximal number of times traversed in the outmost, second, third, and inmost for-loop, respectively. We proceed inwards from outside of the structure. First, d_1 is the number of productions in P , that is, $d_1 = m$. Next, d_2 equals to the cardinality of $\leq_p$, which is no more than $m \times m^n = m^{n+1}$. It is obvious that d_3 is actually the cardinality of the Cartesian product over k sets from Ptp , where k denotes the number of tails in a direct total precedence relation. Since Ptp is a partition of Tpd , each element of the former must be a subset of the latter. Thus, $d_3 < l^n$. As for the inmost loop, d_4 readily equals to n , which is exactly the same as the exponent appearing in the preceding inequation. Consequently, the last structure takes $O(m^{n+2} n l^n)$, i.e., the product of the complexities of the four constituents. Readily, it is also the time complexity of the algorithm. ■

Theorem 1. *The time complexity of Algorithm 4 is $O((n! r^{rn})^{2n^{h-1}})$, where n and r are as in Proposition 1, and h is the depth of the input total precedence relation.*

Proof. The main part of the algorithm is a two-layer structure: a for-loop nested within a while-loop, followed by another for-loop.

As to the former, the total number of times it is traversed is the product of that of the outmost while-loop and of the inmost for-loop. We consider the while-loop first. In the worst case, the input total precedence relation (E_0, R_0) corresponds to a complete n -ary rooted tree of depth h . Then, the cardinality of E_0 , i.e., the number of subtrees of depth 1 that compose it, can be expressed as:

$$1 + n + \dots + n^{h-1} = \frac{n^h - 1}{n - 1} \quad (1)$$

Suppose the number of times the inmost for-loop is traversed at the worst case be w . In each traversal of the while-loop, it takes a rooted tree out from Cps , and then puts as many as w revised ones that comprise one less subtrees back into it. This process is repeated until each element in Cps becomes a rooted tree of depth 1, i.e., it only involves a root element. Consequently,

the maximal number of times the loop is traversed can be expressed as:

$$1 + w + w^2 \dots + w^{\left(\frac{n^{h-1}}{w-1} - 1\right)} = \frac{w^{\frac{n^{h-1}}{w-1}} - 1}{w-1} \quad (2)$$

Further inference to Formula (2) can be done as follows:

$$\begin{aligned} \frac{w^{\frac{n^{h-1}}{w-1}} - 1}{w-1} &< \frac{w}{w-1} \cdot w^{\frac{n^{h-1}}{w-1} - 1} = \frac{w}{w-1} \cdot w^{\frac{n(n^{h-1}-1)}{n-1}} \\ &< \frac{w}{w-1} \cdot w^{2n^{h-1}} = O(w^{2n^{h-1}}) \end{aligned}$$

Note that w exactly equals to the number of extended context-1 productions that can be produced from a direct total precedence relation, by Definitions 4.3 and 4.6. In the worst case, the relation consists of one head and n tails, and each component of the latter's right graphs contains r^r redexes of any component of the former's left graph. To be exact, of the loop variable $(t_1, \dots, t_k)$ for the for-loop, each element t_i can be any of the r^r redexes of the corresponding component of the left graph in any component of any tail's right graph, where $1 \leq i \leq k$, and $k = n$. Thus, for each permutation of the tails, there are $(r^r)^n = r^{rn}$ possible ordered tuples, where n is the maximal number of choices for selecting one component from each tail, and r^r is the maximal number of redexes in each component. Furthermore, the number of permutations for the tails is $n!$. Therefore, $w = n! \cdot r^{rn}$. Substituting the result for w in Formula (2) yields $(n! \cdot r^{rn})^{2n^{h-1}}$.

As to the latter, the number of times it is traversed is $n^{h-1} \times n = n^h$.

Consequently, the time complexity of the algorithm is $O((n! \cdot r^{rn})^{2n^{h-1}})$. ■

V. DISCUSSION

From the perspective of time complexity, the above four algorithms seem rather complicated at first glance. Nevertheless, they are applicable in practical scenarios due to the following three causes.

First, the parameters that characterize a graph grammar are usually small constants, and cannot change in any computation. That is, the parameters are the nature of a graph grammar that will not vary with host graphs in parsing or derivation processes under different situations.

Second, the worst cases theoretically assumed in the analysis of the complexities can rarely happen in practice, and they are frequently quite small number in practical applications. Notice that the number of redexes with respect to a direct total precedence relation is surprisingly $n! \cdot r^{rn}$. However, this amount is merely a theoretical upper bound that accounts for all the possibly matched subgraphs, no matter in which situation all the redexes can simultaneously occur. An extreme situation in this case is a host graph where all the nodes in the host graph are labeled with the same symbol and the directed edges between them are completely connected. However, such host graphs can rarely be encountered in practice. As an example, consider the graph grammar depicted in Figure 1, the number of redexes with respect to a direct total precedence relation is theoretically

$2! \cdot 4^{2 \times 4}$, whereas in the practical computation it is less than 10 in most cases.

Third, the contexts of a graph grammar can be achieved as the output from merely one execution of the algorithms, and then they can be repeatedly utilized in the process of derivation and parsing of this grammar at any time afterwards.

Noticeably, the computation complexities for context can be directly applied to the practical graph grammars specifying real-world visual languages, e.g., BPMN (Business Process Model and Notation), ER diagrams, UML diagrams, WebML (Web Modeling Language), chemical diagrams, and so on, since these graph grammars are concrete examples of the underlying formalisms where the specification of nodes and edges in productions is entirely inherited from the formalisms.

VI. CONCLUSION

Taking RGG as a representative formalism, this paper has proposed an approach to the computation of context on the basis of the formal definition of context, and presented the time complexities of the involved four algorithms. The approach can facilitate the applicability of implicit graph grammars, as contexts of the productions are essential information for the comprehension of graph grammars and efficiency improvement of parsing algorithms. Besides, the approach can be generalized to other implicit context-sensitive graph grammars without much effort.

In future work, further investigation will be made to explore more application scenarios of context, and a support system for context computation and visualization will be developed as well.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation of China under grants 61170089 and 91318301.

REFERENCES

- [1] S. K. Chang, "Visual languages: a tutorial and survey," IEEE Software, 4(1), pp. 29–39, 1987.
- [2] G. Costagliola, V. Deufemia, and G. Polese, "Visual language implementation through standard compiler-compiler techniques," Journal of Visual Languages and Computing, 18(2), pp. 165–226, 2007.
- [3] K. Marriott, "Constraint multiset grammars," IEEE Symposium on Visual Languages, St. Louis, Missouri, pp. 118–125, 1994.
- [4] F. Ferrucci, G. Pacini, G. Satta, et al., "Symbol-relation grammars: a formalism for graphical languages," Information and Computation, 131(1), pp. 1–46, 1996.
- [5] G. Rozenberg (Ed.), Handbook on Graph Grammars and Computing by Graph Transformation, vol.1: Foundations, World Scientific, 1997.
- [6] G. Engels, H. J. Kreowski, G. Rozenberg (Eds.): Handbook of Graph Grammars and Computing by Graph Transformation, vol 2: Applications, Languages, and Tools, World Scientific, 1999.
- [7] H. Ehrig, H. J. Kreowski, U. Montanari, and G. Rozenberg (Eds.), Handbook of Graph Grammars and Computing by Graph Transformation, vol.3: Concurrency, Parallelism, and Distribution, World Scientific, 1999.
- [8] J. Rekers and A. Schürr, "Defining and parsing visual languages with layered graph grammars," Journal of Visual Languages and Computing, 8(1), pp. 27–55, 1997.
- [9] D. Zhang, K. Zhang, and J. Cao, "A context-sensitive graph grammar formalism for the specification of visual languages," The Computer Journal, 44(3), pp. 187–200, 2001.
- [10] M. Nagl, "A tutorial and bibliographical survey on graph grammars," International Workshop on Graph Grammars and Their Application to

- Computer Science and Biology, Lecture Notes in Computer Science, vol. 73, Springer Verlag, 1979, pp. 70–126.
- [11] M. Nagl, “Set theoretic approaches to graph grammars,” International Workshop on Graph Grammars and Their Application to Computer Science, Lecture Notes in Computer Science, vol. 291, Springer Verlag, 1987, pp. 41–54.
 - [12] J. Kong, K. Zhang, and X. Zeng, “Spatial graph grammars for graphical user interfaces,” ACM Transactions on Computer-Human Interaction, 13(2), pp. 268–307, 2006.
 - [13] C. Zhao, J. Kong, and K. Zhang, Program behavior discovery and verification: a graph grammar approach, IEEE Transactions on Software Engineering, 36(3), pp. 431–448, 2010.
 - [14] Y. Liu, X. Zeng, Y. Zou, and K. Zhang, “A graph grammar-based approach for graph layout,” Software: Practice and Experience, 49(8), pp. 1523–1535, 2018.
 - [15] Z. Shi, X. Zeng, Y. Zou, et al., A temporal graph grammar formalism. Journal of Visual Languages and Computing, vol. 47, pp. 62–76, 2018.
 - [16] J. L. Pfaltz, A. Rosefeld, “Web grammars,” International Joint Conference on Artificial Intelligence, pp. 609–619, 1969.
 - [17] X. Zeng, X. Han, and Y. Zou, “An edge-based context-sensitive graph grammar formalism,” Journal of Software, 19(8), pp. 1893–1901, 2008. (in Chinese)
 - [18] Y. Liu, Z. Shi, and Y. Wang, “An edge-based graph grammar formalism and its support system,” International DMS Conference on Visualization and Visual Languages, pp.101–108, 2018.
 - [19] Y. Zou, X. Zeng, and X. Han, “Context-attributed graph grammar framework for specifying visual languages,” Journal of Southeast University (English Edition), 24(4), pp. 455–461, 2008.
 - [20] P. Bottoni, G. Taentzer, and A. Schürr, “Efficient parsing of visual languages based on critical pair analysis and contextual layered graph transformation,” IEEE Symposium on Visual Languages, pp. 59–60, 2000.
 - [21] Y. Zou, J. Lü, and X. Tao, “Research on context of implicit context-sensitive graph grammars,” Journal of Computer Languages, in press, <https://doi.org/10.1016/j.cola.2019.01.002>

Multilevel Visualisation of Topic Dependency Models for Assessment Design and Delivery: A Hypergraph Based Approach

Kendra M.L. Cooper¹ and Hassan Khosravi²

¹Independent Scholar, Canada

²University of Queensland, Australia

Abstract

The effective design and delivery of assessments in a wide variety of evolving educational environments remains a challenging problem. Proposals have included the use of learning dashboards, peer learning environments, and grading support systems; these embrace visualisations to summarise and communicate results. In an on-going project, the investigation of graph based visualisation models for assessment design and delivery has yielded promising results. Here, an alternative graph foundation, a two-weighted hypergraph, is considered to represent the assessment material (e.g., questions) and their explicit mapping to one or more learning objective topics. The visualisation approach considers the hypergraph as a collection of levels; the content of these levels can be customized and presented according to user preferences. A case study on generating hypergraph models using commonly available assessment data and a flexible visualisation approach using historical data from an introductory programming course is presented.

1 Introduction

Assessment remains a core educational activity, even as environments continue to evolve beyond the traditional classroom. Blended, flipped, and massive open on-line courses are supported by a wide range of assessment tools and techniques [1]. Instructors have many options for assessing the required topics of a course in a particular offering [2]. More traditional material includes homework assignments and examinations;

emerging material includes question repositories and games. This rich variety also introduces new challenges to educational stakeholders (e.g., students, instructors, administrators, education researchers) with respect to evaluating the coverage of assessment material and communicating achievements. Students may find it challenging to infer their strengths and weaknesses with respect to the topics and their relationships, which can impede their studies. Outside a classroom program administrators, course designers, course co-ordinators, and researchers also face challenges. Administrators find it challenging to compare the content and difficulty of formal assessments as well as students' outcomes across different offerings of a course. Course designers and co-ordinators find it challenging to ensure the required topics and their relationships (e.g. questions with a combination of topics *a*, *b* and *c*) have been assessed. Educational researchers need to compare the achievement results between control and experimental groups.

To address these assessment challenges, a new field, known as “Learning Dashboards” has emerged, which embraces learning analytics and educational data mining [3, 4, 5]. These dashboards help users to interactively explore and understand data sets through analysis and visualisation techniques. A variety of traditional plots and charts have been adopted (e.g., pie, box, histogram, radial) to visualise the achievements of students. The research typically considers independent (stand-alone) topics. Additional discussion on the related work is presented in Section 3.

As part of an on-going project, the authors of this paper have explored a collection of topic dependency models (TDMs) for assessment in which the relationships among topics are considered [6, 7]. The TDMs

DOI reference number: 10.18293/DMSVIVA2019-018

use a two-weighted undirected graphs foundation to formally represent and visualise a wide variety of assessment data for one or more topics (i.e., topics and their dependency relationship) to meet the needs of diverse stakeholders. The collection consists of a course reference model (to establish the topics and dependencies covered in a course), in addition to classroom models, both static and dynamic. As the work presented in this article leverages these results, a background section on the original TDM collection is presented in Section 1.

In this work, an alternative graph foundation is explored for the TDM collection: a two weighted hypergraph. This graph can explicitly visualise the n-ary topic coverage and achievement inherent in questions (i.e., a question can assess 1:N topics). However, visualisations of their more general n-ary hyperedge relationships may be more difficult to understand in the broader community. Here, a method to generate a two-weighted hypergraph model using commonly available assessment data and a flexible, approach to address the complexity of the graphs are presented. The visualisation approach is a multilevel filtering approach that considers the hypergraph as a collection of levels. The content of these levels can be customised and presented according to user preferences. For example, instead of being presented with the complete hypergraph, the users can select topics, achievement or number of responses (e.g., maximum, minimum, range of values) to view part of it. In addition, the visualisations of the levels can be presented in either a cumulative or accumulative mode. The new models are presented using an illustrative example. A case study that illustrates how the model can be applied to provide insight for instructors is presented. This study uses historical data from a first-year undergraduate level offering of a course on programming and engineering design at The University of British Columbia.

2 Background: The Original TDM Collection

A collection of TDMs has been presented in the author’s previous works [6, 7] to address some of the challenges stakeholders face that relate to the design, delivery and analysis of assessments. An overview of the collection is presented in Figure 1.

Stakeholders and Scenarios of Use. The wide variety of stakeholders interacting with these models (e.g., students, instructors, and so on) are shown at the top of the figure; they have roles inside and outside of the classroom. The earlier results present a preliminary scenario analysis that identifies 40 questions for the

stakeholders. For example, instructors interact with the models to explore questions such as:

- What topics and their relationships do I need to assess?
- What topics are covered each of the assessments?
- How well are the students performing on the topics?
- How well are the students performing compared to other (current or previous) classes?
- Who in the class may be at risk of failing the course?
- How have all the topics and their relationships been assessed?
- How much has the class improved over time on the topics?

Input Data. The input assessment data, shown on the left of Figure 1, includes results from current or previous classes. The data can be from a variety of formative (e.g., assignments) or summative (e.g., quizzes) assessments; the course data includes a specification of the course topics that need to be assessed.

Topic Dependency Models. Based on the requests from stakeholders, the input data are selected and transformed into visual models. As illustrated in main block of Figure 1, the TDM collection consists of two kinds of models: Course Assessment Reference and Classroom Assessment Visualisation (static and dynamic). The Reference model establishes and communicates the required topics and their relationships for a course (e.g., a CS1 course is an introductory programming course covering variables, branching, loops, and so on). It provides a common foundation for all sections of a course offered over time (e.g., CS1 Section 001 Term 1 2019 Instructor A. Smith). The reference model helps to ensure the consistent coverage of topics by different instructors, clearly communicate the scope of the topics to students, and support administrative activities related to monitoring learning objective outcomes.

The Classroom Assessment Visualisation models present assessment data within a class and support comparisons of assessment data between classes at one point in time (static) and over time (dynamic). The assessment data are selectively visualised, e.g., for the whole class, individual students, specific topics, and so on. For example, the user can choose to visualise static models, in a side-by-side comparison, for a specific assessment (e.g., Assignment 3) for all of the students in two classes. Alternatively, the user can choose to visualise dynamic models to explore the progression of a class over time.

Figure 1: Original TDM model collection: two-weighted undirected graph foundation [6].

Graph Foundation of the TDMs The Reference and Classroom Assessment Visualisation models share a common graph foundation: a two-weighted undirected graph. The vertices represent topics in a course; the edges represent assessment material (e.g., questions) that address the topics the edges are related to. The weights are reflected in the visualisation using a colour palette (achievement) and width (coverage). As the graphs can only represent edges with up to two vertices, questions involving three or more topics must be redistributed in the models as combinations, in order to visualise them.

3 Related Work

The learning analytics community continues to actively investigate approaches that support the exploration of learning activities by different stakeholders. With the increase in the use of educational technologies and the advancements in the areas of learning analytics and educational data mining, a new field, commonly known as “Learning Dashboards” has emerged to help make sense of data sets in learning and ed-

ucation [3, 4, 5]. A variety of visualisations such as bar charts [8], pie charts [9], histograms [10], box plots [11], radar graphs [12], and skill meters [13] have been adopted to show the achievements of students for independent (stand-alone) topics. [3] presents a systematic literature review on the use of learning dashboards. Based on the findings of this literature review, the use of graph-based visualisations in learning dashboards has not received much attention. In addition, a number of studies provide strong evidence that opening the model to learners, leading to the notion of Open Learner Models (OLMs) [14], can be effective in helping students learn [15]. The OLMs commonly use a set of individual topics as their underlying structure for modeling learners’ knowledge state, which ignores relationships among topics. An emerging new field applied methods from process mining and sequential data mining to educational data [16] to facilitate better understanding of the educational process. Educational process mining methods predominantly use graphs, but their main focus is on using activity logs to visualise students’ learning process in terms of the time, place, path, pace of learning activities.

Hypergraphs have been adopted as a foundation for data analytics and visualisations in a wide variety of domains including data warehousing, communication network analysis, geospatial metadata, and cellular biology networks. To support interactive queries in data warehousing systems, a framework for developing dashboards called Dashboard-by-example has been proposed [17]. This framework adopts hypergraph-based techniques to transform dissimilar, heterogeneous data into a homogeneous knowledge space of clusters and partitions. The framework is flexible, as the hypergraph-guided data linkages support the exploration and aggregation of data from multiple perspectives. For the analysis of network traffic traces, available in massive communication logs, a hypergraph based visualisation is proposed in [18]. The network traffic visualization approach utilises a frequent item set mining method to identify interesting traffic patterns in the large amount of data. The patterns are visualised as hypergraphs with explicit, multi-attribute relationships. A framework to comprehensively address all of the available geospatial metadata standards (i.e., documents) is proposed in [19]. Geospatial metadata describe geographic digital data resources such as earth imagery, geospatial databases and catalogues, and Geographic Information System files. The framework integrates hypergraphs and topic maps, representing the elements and their dependency relationships. The potential for adopting hypergraphs in the domain of cellular biology is introduced in [20]; recently, for example, [21] proposes a framework that adopts hypergraphs and associated hypergraphs to describe, analyse, and identify metabolic network alignments at the full genome level. These alignments are used to discover important similarities and differences between distinct molecular networks: they reveal mappings between components (topological, biological functional) across different networks.

To the best of our knowledge, a modeling and visualisation approach based on hypergraphs for educational assessment is not available at this time. The TDM_{MH} helps to address this gap in the literature.

4. Methodology

This section define the graph foundation (hypergraph, level), scenarios of use, and an illustrative example. The graph generation and visualisation algorithm are presented in Section 5, using the example. A case study demonstrating TDM_{MH} s based on data collected from the final exam of a large introductory course on programming is presented in Section 6.

4.1 Hypergraph Foundation of the TDM_{MH}

In this section, the formal definition for a two-weighted hypergraph and the concept of a level are presented.

Definition 1 A two-weighted, undirected hypergraph $G = (V, H)$, where V is the set of vertices representing the topics, and H is the set of hyperedges. A hyperedge $h \in H$ is represented as $h = (C, c_1, c_2)$, where C is the subset of the vertices being connected, and c_1 and c_2 represent the two weighted values for an edge. An edge involving only one node represents a self-loop.

In this work, the nodes are used represent topics; the hyperedges represent the assessment material that covers the topics. The weight c_1 represents the number of learning objects that are tagged with the topics for the hyperedge; and c_2 represents the represents the performance (e.g., achievement) on these learning objects. A simple example of a TDM_{MH} visualisation is illustrated in Figure 2. There are four topics in this example (A, B, C, and D); each topic is represented by a node. Topic B and Topic C both have a 1-ary hyperedge (i.e., a self loop). For Topic B, the edge has a dark pink colour and a wide width, reflecting the poor average performance (33%) on the questions answered and the large number of responses. The hyperedge for Topic C is a medium green colour and has a medium width; this represents the good average performance (75%) and a moderate number of responses to questions. A 2-ary hyperedge exists between Topics A and D. This edge is dark green and thin, in representing the excellent performance (100%) on the related questions and a small number of responses. A 3-ary hyperedge exists between Topics A, B, and D. This edge is a brownish pink colour and has a moderate weight, which represents the moderate average performance (50%) and a moderate number of responses to questions on these three topics.

The concept of a level is used in this work to support the visualisation. For example, in the simple example above, there are three levels. $Level_1$ is a subset of the hyperedges with one vertex (i.e., h_1, h_2), $Level_2$ is a subset of the hyperedges with two vertices (i.e., h_3), and $Level_3$ is a subset of the hyperedges with three vertices (i.e., h_4). More generally, $Level_i$ is the subset of hyperedges involving i vertices.

4.2 Scenarios of Use

As with the original TDM models, the users can select the assessment data of interest. For example, they may need to explore data for formative or summative

Figure 2: Example of a two-weighted hypergraph: definition and visualisation

assessments (e.g., one or more assignments or examinations); the data can be from one or more classes. Here, scenarios of use include filtering by topic(s), or constraints involving achievement or coverage. Additional filters can be added in a straightforward way. Currently, the scenarios focus on using the static models; the reference and dynamic models for the TDM_{MH} are planned in future work. Two examples, derived from the scenario analysis in [6], are described below.

Topics. How are the students doing on a specific topic? One or more topics of interest can be selected in order to identify topics that may need additional attention. For example, an instructor can select one of many topics covered in a course; the levels are filtered to emphasize the topic(s) of interest.

Achievements. What topics have poor achievement? A constraint for visualising a particular achievement level can be selected. For example, an instructor can select to view the minimum achievement scores over all topics in order to identify those that need additional attention; the levels are filtered to emphasize the achievement scores of interest.

The users can choose to combine filters. In other words, they can filter the visualisations based on one or more constraints such as selecting both a topic and an achievement constraint, e.g. highest performance involving topic T_1 . The results are presented on a level-by-level basis (a level is described in Section 4.1).

When visualising the results, the user can also choose to view the results in either the cumulative or accumulative mode. In the accumulative mode, only the results for one level are presented. In the cumulative mode, the results up to and including the current level of interest are presented.

4.3 Illustrative Example

An example based on six students, three formative assignments (five questions per assignment), and six topics has been defined by the authors. The following constraints are considered for creating the illustrated example:

- The questions must span a range of one to four topics.
- There must be a range in the individual student achievements in the responses (low to high scores).
- There must be a range in the number of responses to the questions.
- There must be a range in the average scores for the questions (0%-100%).
- Assessments on groups of topics demonstrate a range of achievements (very poor to very strong); one or more questions address the groups of topics.
- The multilevel visualisation algorithm with a list of filtering options (e.g., selected topic(s), achievement, cumulative/accumulative mode) can be clearly illustrated.

Table 1 presents a sample data set meeting these constraints. As input to the TDM_{MH} generation algorithm, the data are formatted into two CSV files: (1) A student-question-answer (SQA) file that contains the student identifier, question identifier, and score (correct/incorrect), and (2) a question-topic (QT) file that contains the question identifier and topics (i.e., tags) it addresses.

5 Generating and Visualising TDM_{MH}

In this section, the approach to generating and filtering a TDM_{MH} is presented. Section 5.1 presents methods for generating the graph and Section 5.2 explores filtering mechanisms to address the complexity of hypergraph visualisations.

5.1 Generating TDM_{MH}

This section demonstrates how commonly available input data (student achievements/grades for specific questions and the mapping from the questions to the course topics) are transformed into a TDM_{MH} . High-level code and notation are presented in Algorithm 1. The algorithm consists of three high level steps: create working dictionaries and matrices; define the TDM graph elements (vertices and edges); and visualise (i.e., plot) the TDM graph.

Q	Topic List	Student ID					
		S_1	S_2	S_3	S_4	S_5	S_6
Assignment 1							
Q_1	T_1	-	1	-	-	0	1
Q_2	T_3	1	1	1	1	1	1
Q_3	T_4	0	1	0	1	1	1
Q_4	T_1, T_2	1	1	0	1	1	0
Q_5	T_1, T_4	1	-	-	0	1	0
Assignment 2							
Q_6	T_4	0	0	1	0	0	1
Q_7	T_1, T_4	-	0	1	1	0	0
Q_8	T_4, T_5	0	1	0	0	0	1
Q_9	T_1, T_4, T_5	0	0	0	0	-	1
Q_{10}	T_1, T_2, T_4, T_5	0	0	0	1	0	1
Assignment 3							
Q_{11}	T_1, T_4	1	-	0	1	-	1
Q_{12}	T_2, T_6	0	0	0	0	0	0
Q_{13}	T_1, T_2, T_6	-	-	1	1	-	1
Q_{14}	T_1, T_2, T_4, T_5	-	0	1	0	0	1
Q_{15}	T_2, T_4, T_5, T_6	-	0	0	1	1	1

Table 1: Illustrative example: questions, topic lists, and student responses. The value 1 indicates the question is answered correctly; 0 indicates it is answered incorrectly; and - indicates it is not attempted.

The first six steps of Algorithm 1 transform the data in the SQA.csv and QT.csv files into working dictionaries and matrices. In Steps 1, 2, and 3 of the algorithm, three dictionaries are created, $QDict$, $SDict$, $TDict$, to map array and matrix indices to question, student, and topic identifiers. In Steps 4, 5, and 6, three working matrices are created: T , A , and R . The information on topics assigned to each question is represented in matrix T , in which $t_{ij} = 0$ indicates that question i is not tagged with topic j and $t_{ij} = 1$ indicates that question i is tagged with topic j . The correctness of the answers provided by the users are represented in matrix A , where $a_{ij} = 0$ indicates that user i has answered question i incorrectly, $a_{ij} = 1$ indicates that user u has answered question i correctly, and $a_{ij} = n$ indicates that user u has not attempted question i . Matrix R is used to keep track of attempted questions, where $r_{ij} = 0$ if user i has not attempted question i and one otherwise.

Steps 7 and 8 in Algorithm 1 establish the graph vertices and hyperedges. $VList$ stores the list of the vertices of the TDM_{MH} graph and $HList$ stores the list of its hyperedges. The coverage and competency associated with an edge are both computed using T , A and R within the $ComputeH$ function. The coverage (Cov) associated with a hyperedge among a set of vertices $V = v_j, \dots, v_k$ is computed using the following formula:

Algorithm 1 Generating a TDM_{MH}

Require: $SQA.CSV$, $QT.CSV$, $filters$

Create dictionaries and matrices for efficient indexing

- 1: $QDict \leftarrow CreateQDict(SQA.CSV)$
- 2: $SDict \leftarrow CreateSDict(SQA.CSV)$
- 3: $TDict \leftarrow CreateTDict(QT.CSV)$
- 4: $T \leftarrow CreateT(QT.CSV, QDict, TDict)$
- 5: $A \leftarrow CreateA(SQA.CSV, SDict, QDict)$
- 6: $D \leftarrow CreateD(SQA.CSV, SDict, QDict)$

Compute the Graph Elements: Vertices and Edges

- 7: $VList \leftarrow ComputeV(TDict)$
- 8: $HList \leftarrow ComputeH(T, A, D, TDict)$

Create and visualise the Graph

- 9: $TDMStatic \leftarrow CreateTDM(VList, HList)$
 - 10: $Visualise(TDMStatic, filters)$
-

$$Cov(V) = \sum_{i \in QDict} t_{ij} \times \dots \times t_{ik} \sum_{u \in SDict} d_{ui} \quad (1)$$

The outer summation loops through all of the questions. A question i contributes to $cov(V)$ only if it is tagged with exactly the topics that are included in V (i.e., $t_{ij} \times \dots \times t_{ik} = 1$). For example, a question tagged with topics (T_1, T_2, T_3) attempted by nine learners contributes $\frac{9}{1+0} = 9$ to the coverage of $Cov(\{T_1, T_2, T_3\})$.

The achievement ($Achv$) associated with a hyper-edge between a set of vertices $V = v_j, \dots, v_k$ is computed using the following formula:

$$Achv(V) = \frac{\sum_{i \in QDict} t_{ij} \times \dots \times t_{ik} \sum_{u \in SDict} a_{ui}}{Cov(V)} \quad (2)$$

A question i contributes to $Achv(V)$ only if it is tagged with exactly the topics that are included in V (i.e., $t_{ij} \times \dots \times t_{ik} = 1$). The extent of the contribution is determined by the the number of learners who have correctly answered the question, which is computed by $\sum_{u \in SDict} a_{ui}$. The numerator of this formula computes the number of times questions tagged with exactly topics that are included in V . Dividing this number by $Cov(V)$ produces the probability of correctly answering questions on topics in V . Note that Cov and $Achv$ of self-loops can also be computed via the same two formulas by using $V = v_j$.

Figure 3 represents the data which are stored in dictionaries $QDict$, $SDict$ and $TDict$ and matrices T , A and R .

QDict:

{'Q₁':0, 'Q₂':1, 'Q₃':2, 'Q₄':3, 'Q₅':4, 'Q₆':5, 'Q₇':6, 'Q₈':7,
'Q₉':8, 'Q₁₀':9, 'Q₁₁':10, 'Q₁₂':11, 'Q₁₃':12, 'Q₁₄':13, 'Q₁₅':14}

SDict:

{'S₁':0, 'S₂':1, 'S₃':2, 'S₄':3, 'S₅':4, 'S₆':5}

TDict:

{'T₁':0, 'T₂':1, 'T₃':2, 'T₄':3, 'T₅':4, 'T₆':5}

Questions (15)		Questions (15)	
Questions (15)	Topics (6)	Students (6)	Questions (15)
	100000		n 1 0 1 1 0 n 0 0 0 1 0 n n n
	001000		1 1 1 1 n 0 0 1 0 0 n 0 n 0 0
	000100		n 1 0 0 n 1 1 0 0 0 0 0 1 1 0
	110000		n 1 1 1 0 0 1 0 0 1 1 0 1 0 1
	100100		0 1 1 1 1 0 0 0 n 0 n 0 n 0 1
	000100		1 1 1 0 0 1 0 1 1 1 1 0 1 1
	100100		
	000110		
	100110		
	110110		
	100100		
	010001		
	110001		
	110110		
	010111		
Matrix T		Matrix R	

Figure 3: The data represented in Table 1 are organized into two input files: SQA.csv and QT.csv. The data are loaded, transformed into three dictionaries: *QDict*, *SDict* and *TDict* (Steps 1, 2 and 3 in Algorithm 1). The dictionaries and the input files are used to create three working matrices: T, A, and R (Steps 4, 5 and 6 in Algorithm 1). These matrices are used to create the TDM_{MH} model.

Steps 9 and 10 of Algorithm 1 create and visualise the graph. Figure 4 illustrates the resulting model, without any filtering. It has six vertices as the questions in this data set are tagged with six topics: T_1 , T_2 , T_3 , T_4 , T_5 , and T_6 . The coverage of the hyperedges are computed using Formula 1. For example, the hyperedge with id h_5 has coverage of 13, which consists of contributions of 4 from question 5, 5 from question 7, and 4 from question 11. The achievement of the hyperedges are computed using Formula 2. For example the achievement of the hyperedge with id h_5 is computed as $\frac{7}{13} = 0.54$.

5.2 Visualising TDM_{MH} (static model)

In Step 10 of Algorithm 1, the TDM_{MH} can be presented in either whole or part according to the filters the user selects. To address the complexity of the model, the topology of the graph is used; the illustrative example (with six nodes) provides six lev-

$G = (V, H)$

$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$v_1: T_1, v_2: T_2, v_3: T_3, v_4: T_4, v_5: T_5, v_6: T_6$

$H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}\}$

one topic

$h_1: (\{v_1\}, 3, 0.67)$

$h_2: (\{v_3\}, 6, 1.00)$

$h_3: (\{v_4\}, 12, 0.50)$

two topics

$h_4: (\{v_1, v_2\}, 6, 0.83)$

$h_5: (\{v_1, v_4\}, 13, 0.54)$

$h_6: (\{v_4, v_5\}, 6, 0.33)$

$h_7: (\{v_2, v_6\}, 6, 0.00)$

three topics

$h_8: (\{v_1, v_4, v_5\}, 5, 0.20)$

$h_9: (\{v_1, v_2, v_6\}, 3, 1.00)$

four topics

$h_{10}: (\{v_1, v_2, v_4, v_5\}, 4, 0.36)$

$h_{11}: (\{v_2, v_4, v_5, v_6\}, 5, 0.60)$

Figure 4: Complete TDM_{MH} for the data set provided in Table 1 using Algorithm 1 (no filtering). The example does not have questions involving five or six topics.

els. Table 2 illustrates the levels before any filtering is done. For example, $Level_1$ is the subset of hyperedges $\{h_1, h_2, h_3\}$.

If the user does not select any filters, then all levels are displayed by default, including all topics, achievements, and coverage of the learning objects as illustrated in Figure 4. Alternatively, the selected filters are applied to each level. The selected elements are presented using the original TDM colour palette; other items are greyed out. $Level'_i \leftarrow filter(Level_i, t, a, n)$, where the parameters t , a , and n represent optional constraints for the topics, achievements, and number of learning objects. For example, if a user requests to filter with respect to topic T_1 , the resulting filtered levels are illustrated in Table 2. In this case, for example, the filtered $Level'_i$ is the subset of hyperedges $\{h_1\}$, as only h_1 contains T_1 .

After the levels are filtered, they are ready to be vi-

Figure 5: Multilevel visualisation: one topic is selected (T_1) as a filter (accumulative mode).

Level ID	Level Membership	Level' Membership (T_1)
$Level_1$	$\{h_1, h_2, h_3\}$	$\{h_1\}$
$Level_2$	$\{h_4, h_5, h_6, h_7\}$	$\{h_4, h_5\}$
$Level_3$	$\{h_8, h_9\}$	$\{h_8, h_9\}$
$Level_4$	$\{h_{10}, h_{11}\}$	$\{h_{10}\}$
$Level_5$	$\emptyset$	$\emptyset$
$Level_6$	$\emptyset$	$\emptyset$

Table 2: Levels in the illustrative example graph: before and after filtering (e.g., topic T_1).

sualised. The user can choose to present them in either an accumulative or cumulative mode. In the accumulative mode, only the hyperedges for one level are presented at a time. The first four levels are illustrated in 5 in the accumulative mode. In the cumulative mode, the presentation for a level includes the hyperedges for the current levels and the previous levels. A preliminary layout algorithm is used in the prototype tool at this time; however, adopting a more effective proposal from the literature is planned in the next step.

This straightforward visualisation example based on filtering the levels for one topic can be extended to combinations of one or more topics, achievements, and coverage. The options to highlight part of the graph, in either cumulative or accumulative mode, provide a flexible approach for users. However, to easily access more detailed information (e.g., exploring specific achievement values for hyperedges), additional research is needed to extend the TDM_{MH} visualisations.

6 Case Study

This section presents a case study that demonstrate the applications of multilevel hypergraph TDMs. This case study is based on data collected from a first-year undergraduate level offering of a course on programming and engineering design at The University of

British Columbia. This offering of the course had 377 students and was held during the Fall of 2016. The course covers many topics that are generally included in an introductory course on programming and engineering design in 8 modules: number conversions, programming fundamentals, conditionals, loops, file I/O, functions, arrays, strings, and DAQ systems. Functions, strings and DAQ systems received two weeks of lecture time; all of the other modules received roughly one week of lecture time.

We use data collected from the final exam of the course, which is captured via the Gradescope platform [22], a system for the on-line assessment of handwritten exams, are used. The final exam of this offering had 17 independent sub-questions that formed a total of eight main questions. The questions are tagged using 20 concept-level topics, defined by the instructors, based on the 8 modules that are covered in this course. For example, the Strings Module is further decomposed into three topics: String-Length, String-Copy, and String-Compare, providing a finer level of granularity.

Figure 6 shows the hypergraph TDM for this exam. It indicates that the exam adequately covered all of the modules of the course and that all modules except Conversions (including the Hex and Octal topics) were highly connected to one another. The exam included questions that had only a single topic (e.g., h_1 on Hex), two topics (e.g., h_{14} on While loops and DAQ-write), three topics (e.g., h_9 on Printing, File output and 2D arrays) and four topics (e.g., h_{17} on For loops, 1D arrays, Write functions and String-compare).

The further decomposition to concept-level tags provides insights which might have not been possible to gauge using module-level tags. For example it shows that students did well on questions on String-Copy and String-Compare but not so well on questions on String-Length. As another example, further decomposition

makes it scalable and efficient. The visualisation helps to address the complexity of a hypergraph through a flexible, multilevel approach. A case study that illustrates how TDM_{MH} can be applied to provide insight in the context of a large university course is presented.

A number of directions are planned for the next steps in this research, including exploring the TDM_{MH} dynamic model, refining the prototype tool, and additional validation studies, in particular a user study to compare the strengths and limitations of the original TDM and TDM_{MH} . In addition, highly interactive 3D visualisations of the TDM_{MH} may also be of great value to the diverse stakeholders who are involved with the design and delivery of assessments.

References

- [1] C. Englund, A. D. Olofsson, L. Price, Teaching with technology in higher education: understanding conceptual change and development in practice, *Higher Education Research & Development* 36 (1) (2017) 73–87.
- [2] P. Griffin, E. Care, *Assessment and Teaching of 21st Century Skills*, 2015.
- [3] B. A. Schwendimann, M. J. Rodriguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet, P. Dillenbourg, Perceiving learning at a glance: A systematic literature review of learning dashboard research, *IEEE Transactions on Learning Technologies* 10 (1) (2016) 30–41.
- [4] V. Argumentation, Software tools for collaborative and educational sense-making, Paul A. Kirschner, Simon J. Buckingham Shum and Chad S. Carr (Eds.) Springer-Verlag: London (2003) 117–135.
- [5] Y. Park, I.-H. Jo, Development of the learning analytics dashboard to support students’ learning performance, *Journal of Universal Computer Science* 21 (1) (2015) 110.
- [6] H. Khosravi, K. Cooper, Topic dependency models: Graph-based visual analytics for communicating assessment data, *Journal of Learning Analytics* 5 (3).
- [7] K. Cooper, H. Khosravi, Graph-based visual topic dependency models: Supporting assessment design and delivery at scale, in: *Proc of the 8th International Conference on Learning Analytics and Knowledge*, 2018, pp. 11–15.
- [8] S. Dawson, L. Macfadyen, E. F. Risko, T. Foulsham, A. Kingstone, Using technology to encourage self-directed learning: The collaborative lecture annotation system, in: *Proc of the conference of the Australian Society for Computers in Tertiary Education*, 2012.
- [9] A.-M. Tervakari, K. Silius, J. Koro, J. Paukkeri, O. Pirttila, Usefulness of information visualizations based on educational data, in: *Global Engineering Education Conference*, 2014, pp. 142–151.
- [10] D. Leony, A. Pardo, L. de la Fuente Valentín, D. S. de Castro, C. D. Kloos, Glass: a learning analytics visualization tool, in: *Proc of the 2nd International Conference on Learning Analytics and Knowledge*, 2012, pp. 162–163.
- [11] D. Albert, A. Nussbaumer, C. Steiner, Towards generic visualisation tools and techniques for adaptive e-learning, in: *Proc of the 18th International Conference in Computers in Education*, 2010, pp. 61–65.
- [12] M. May, S. George, P. Prevot, Travis to enhance online tutoring and learning activities: Real-time visualization of students tracking data, *Interactive Technology and Smart Education* 8 (1) (2011) 52–69.
- [13] S. Bull, B. Ginon, C. Boscolo, M. Johnson, Introduction of learning visualisations and metacognitive support in a persuadable open learner model, in: *Proc of the Sixth International Conference on Learning Analytics & Knowledge*, 2016, pp. 30–39.
- [14] S. Bull, J. Kay, Open learner models, in: *Advances in Intelligent Tutoring Systems*, 2010, pp. 301–322.
- [15] R. Bodily, J. Kay, V. Aleven, I. Jivet, D. Davis, F. Xhakaj, K. Verbert, Open learner models and learning analytics dashboards: a systematic review, in: *Proc of the 8th International Conference on Learning Analytics and Knowledge*, 2018, pp. 41–50.
- [16] A. Bogarín, R. Cerezo, C. Romero, A survey on educational process mining, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (1).
- [17] D. Hoang, B. Nguyen, M. Tjoa, Dashboard by-example: A hypergraph-based approach to on-demand data warehousing systems, in: *Proc of the 2012 IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 1853–1858.
- [18] E. Glatz, S. Mavromatidis, B. Ager, X. Dimitropoulos, Visualizing big network traffic data using frequent pattern mining and hypergraphs, *Computing* 96 (1) (2014) 27–38.
- [19] G. Rajaram, H. C. Karnatak, S. Venkatraman, K. R. Manjula, K. Krithivasan, A novel computational knowledge-base framework for visualization and quantification of geospatial metadata in spatial data infrastructures, *GeoInformatica* 22 (2) (2018) 269–305.
- [20] S. Klamt, U.-U. Haus, F. Theis, Hypergraphs and cellular networks, *PLoS computational biology* 5 (2009) e1000385.
- [21] T. Shen, Z. Zhang, Z. Chen, D. Gu, S. Liang, Y. Xu, R. Li, Y. Wei, Z. Liu, Y. Yi, X. Xie, A genome-scale metabolic network alignment method within a hypergraph-based framework using a rotational tensor-vector product, *Scientific Reports* 8.
- [22] A. Singh, S. Karayev, K. Gutowski, P. Abbeel, Grade-scope: a fast, flexible, and fair system for scalable assessment of handwritten work, in: *Proc of the 4th Conference on Learning@ Scale*, 2017, pp. 81–88.

CROSSIDE: A Cross-Surface Collaboration by Sketching Design Space

Jorge-Luis Pérez-Medina¹, Jean Vanderdonckt² and Santiago Villarreal-Narvaez²

¹ Intelligent & Interactive Systems Lab, Universidad de las Américas. Quito, Ecuador

² Université catholique de Louvain (UCL). Louvain-la-Neuve, Belgium

Abstract—This paper introduces, motivates, defines, and exemplifies CROSSIDE, a design space for representing capabilities of a software for collaborative sketching in a cross-surface setting, i.e., when stakeholders are interacting with and across multiple interaction surfaces, ranging from low-end devices such as smartwatches, mobile phones to high-end devices like wall displays. By determining the greatest common denominator in terms of system properties between forty-one references, the design space is structured according to seven dimensions: user configurations, surface configurations, input interaction techniques, work methods, tangibility, and device configurations. This design space is aimed at satisfying three virtues: *descriptive* (i.e., the ability to systematically describe any particular work in cross-surface interaction by sketching), *comparative* (i.e., the ability to consistently compare two or more works belonging to this area), and *generative* (i.e., the ability to generate new ideas by identifying potentially interesting, undercovered areas). A radar diagram graphically depicts the design space for these three virtues.

I. INTRODUCTION

In many domains of human activity, across them [31] and within [17], *sketching* is largely used as a quick, efficient, and cost-effective tool for expressing ideas, illustrating design concepts and solutions as well as for sharing them with diverse people [49]. During the early stages [25] of product development until its final completion, sketching immediately plays some role when co-design is a must [52]. Sketching facilitates the sharing of abstract ideas and the insights inside a team while offering a common ground for the discussion, especially when the participants come from different cultural and social backgrounds. Sketching usually uses basic drawings to foster everyone's participation [10].

Sketching, as one particular form of drawing, belongs to the first human intellectual skills and abilities that are acquired even before speaking, writing, and precise drawing. By one year, infants understand that a sequence of sounds form a word that symbolically represents an action, a relation, or an object they can point to rather than reaching it. Our human ability to perceive and to recognize elements in a graphical representation is in itself a statement of the expression power yielded by such form of expression.

A *sketch*, coming from the Greek word $\sigma\chi\epsilon\delta\iota\omicron\zeta$ (schedios - *done ex tempore*), consists in a quick provisional freehand drawing executed without any constraint in any medium. A sketch may convey a message, record or develop an abstraction or may be used as a mean for explaining something, for example an image. A sketch is considered as a very quick drawing aimed at communicating a message, which can be understood, misinterpreted, or even ignored. But there is always a message that is different from the one of a *drawing*.

For these reasons, within many professions such as industrial and architecture design [52], representing actions, objects, and their relations using different forms of sketches such as drafts, blueprints, and prototypes, has always been very important. They make visible the stakeholder's contribution, which could be different than expected.

The audience of stakeholders involved in sketching activities is now recognized to be wider. Consequently, the pool of software tools to support collaborative sketching, both commercial and from the academia, also becomes more filled. *How much sketching activities could be supported by collaborative tools* is therefore a key question addressed in this paper. Researchers should be informed about capabilities of existing tools to update their research agenda and to better understand similarities and differences. Practitioners should also be informed about which tool would match their requirements for conducting collaborative sketching.

For this purpose, the remainder of this paper is structured as follows: Section 2 reviews the concept of sketching across some disciplines, mainly those involving some form of co-design, and discusses selected works in these disciplines; based on these observations, Section 3 introduces, motivates, and defines CROSSIDE, a design space for representing the key capabilities of software for collaborative sketching, with a special attention paid to cross-surface collaboration; Section 4 exemplifies some case studies with this design space; Section 5 concludes this paper by explaining how this design space could be systematically used to describe, compare, and invent tools for collaborative sketching on multiple surfaces of interaction.

II. RELATED WORK

In Information and Communications Technologies (ICT) as well as in computer and software systems engineering, a significant amount of resources is devoted to designing a concept, a service, a solution which could be later on revealed as inadequate. This could happen when the functional requirements are not satisfied, when the user experience is not met, or when the concept is simply not technologically feasible or too expensive. When such a mismatch is discovered late in the development life cycle or after the deployment, adapting what is required to be changed represents a high cost.

Early sketching in the development life cycle helps creating alternative solutions and comparing them. It helps to ensure that the right approach to design the right solution is put in place. It keeps the design and development right. With a sketch, designers and other stakeholders are able to

identify figures, arrows, symbols and other elements that were deliberately chosen by a computer actor to communicate with stakeholders, to illustrate the requirements and share design ideas. It is more efficient and effective than any textual, graphical or formal specifications.

By using sketches, designers become more motivated, more creative, and perhaps more able to address the challenges of creating a successful design, and thus to produce a better design outcome [56]. By definition, prototypes are scaled-down versions of what will be built. Designers use them because they are faster and cheaper to create than the final blueprints. Sketching is a quick provisional drawing, it therefore matches the requirements of prototyping [49].

In Human-Computer Interaction (HCI) design, Collaborative User-Centered Design (CUCD) process suggests using sketch to understanding and designing all the aspects of a user interface (UI) design and for getting involved a wide community of stakeholders, such as, but not limited to: user researcher, information architect, user interface interaction designers, user testing specialists, software developers, marketing personnel and software products leaders. Early sketches inform the development of User interface conceptual, interaction style, and even other related material such training resources, support services, online help, etc. UI usability can be seen as a design problem of “wicked nature”, as a problem to be solved – the more you try to solve the problem, the more you discover the complexity of the UI usability and more you are able to suggest solutions in the early design phase of the CUCD life cycle. This is because usually original usability problem has implications/consequences that cannot be known in advance. Sketches have been shown very powerful to build a consensus and a trade-off when usability problems are conflicting with other major quality factors, such as security. A UI sketch reveals how much a system could be usable, secure.

Sketching is the practice of drawing a rough outline or rough draft version of a final piece of art. Sketching is an aid to thought. Sketches are used as a mean of designing. Design by sketching has its foundations on the participatory design approach [37] in which a person not trained, qualified or experienced is an active and essential participant in the design process. As a communication tool, sketching can be used as a way of graphically specifying abstract ideas. It is a message from the designer to stakeholders. It can be understood by the receiver, misinterpreted, or ignored, but there is definitely a message. This message must be validated when there is a consensus to be achieved between the designer and someone else, for which designers often use limited or scaled versions of what is being designed.

A common ground to disciplines relying on is that stakeholders, particularly designers and end-users, whatever their background and skills in sketching and design are, feel actively engaged [5]. End-users help in materializing some requirements such as usability [50]. Designers then annotate original sketches introduce by end-users, add illustrations, and further develop them. They usually proceed by iteratively sketching a concept at different levels of abstraction [45].

Ambler [1] defines UI Prototyping as an iterative analytical technique in which end users are actively involved, namely by providing feedback since the early development stages and continuously afterwards.

Sketching covers many domains of human activity and inside these domains, there are several works exploiting sketching for one or many sub-activities such as for example: problem analysis in general [44], computer science [17] (e.g., user experience support [8], [18], user interface design, prototyping, and recognition [23], [13], [14], [39], [3], [2], [47], cross-device UI design [25], user-centered design in agile projects [32], [7], system walkthrough [24]), system development (e.g., QUILL [16] for model-based design of web applications), flexible modelling [40] (e.g., FlexiSketch [54], [55] for model sketching), RAPIDO [35] for web API development, sketching UML models (e.g., TAHUTI for sketching UML Class diagrams [19] and SketchML for various UML diagrams [3]), distributed software design [29], [33], task modelling [41], notation creation [55]), computer-supported collaborative work [21] (e.g., stakeholders’ meetings [20], collocated tables for meetings [51] and interactive design spaces [6]), product and service design (e.g., sketching in design [21], extreme designing [15], industrial design [52], shape-changing products [42]), public displays [36], learning (e.g., classroom design studio [26], teaching geometry [11]), ideation [4] and concept generation [56], knowledge design, capture, and sharing [31], design in any area of engineering [56] (e.g., knot diagramming [43]). Many techniques have been successfully reported for expressing sketch grammars [10], with the need to take into account the context of use (i.e., the user, the devices/platforms, and the environment) [9] to get context-aware sketching [11].

III. A DESIGN SPACE FOR CROSS-SURFACE COLLABORATION BY SKETCHING

We determined the greatest common denominator in terms of properties between the 41 aforementioned references, independently of their domain, provided that collaborative sketching is involved to some extent. This identification resulted into CROSSIDE, a design space expressing collaborative sketching according seven dimensions represented clockwise in Fig. 1: user configurations, surface configurations, input interaction techniques, work methods, tangibility, layout, and device configurations. Each dimension is organized according to a progressive degree of sophistication: each step starts from the simplest value found in the literature until the most sophisticated degree. We considered this representation as adequate to satisfy three virtues: *descriptive* (the design space should be able to describe any work on collaborative sketching based on these seven dimensions), *comparative* (the design space should be able to compare two or more works on collaborative sketching to identify their similarities and differences) and *generative* (once a comparison is performed, the design space should enable to identify undercovered areas and generate new and interesting configurations).

Fig. 1. A design space for Cross-Surface Collaboration by Sketching.

These seven dimensions are not intended to be completely independent of each other. Rather, they are aimed at serving these three virtues. Moreover, a radar chart can be effectively used to graphically represent the values along these dimensions by representing them on axes starting from the same origin. The steps on each dimension are not intended to be aligned or to be corresponding by concentric level. Therefore, steps joined on a same circle are not necessarily dependent, they only represent some progression.

A. User Configurations

The *user configurations* provide multiple ways to carry out a task by different stakeholders. Since many people could be located in different places, possibly at different levels of various organisations, the design space should consider the group configuration. Fig. 2a depicts various configurations among stakeholders involved in a distributed task [5]: *individual* (one task is carried out by one person in a group), *within groups* (one task is distributed across persons of a same group in the organisational structure), *group as a whole* (one task is carried out by one group of persons, independently of its internal organisation), *among groups* (one task is passed from one group to another), *within organisation* (one task is distributed across entities of the organisational structure), *among organisations* (when one task is distributed across several different organisations, all having their own internal structure), and *between organisations and their environment* (when tasks are exchanged between organisations and their common environment). Collaborative sketching requires stakeholders within groups because sketching tasks are distributed across people of a same group of the same organisational structure or not.

Fig. 3 graphically depicts an evolutive scale of the user configuration for different user configurations with respect to number of sessions and scenarios: an isolated stakeholder working alone, an individual stakeholder working as a group

member, a single group of stakeholders, multiple groups of different stakeholders, multiple groups within the same organization (e.g., a group of representative end-users in a bank vs a group of interaction designers in the same bank), several groups across organizations (e.g., an association of interaction designers in a particular domain of human activity) or the whole world (e.g., a community of practice could be developed that gathers people involved in a particular family of designs).

B. Surface Configurations

The *surface configurations* express a hypothetical distance between a stakeholder working alone and multiple ones collaborating in a personal, social, public or remote environment, thus touching the notion of territoriality. Fig. 2c differentiates configurations depending on concentric spatial zones centered around the stakeholder. The distance at which users feel comfortable interacting with others in tabletop environments depend on age and culture [51]. This study has contemplated the type of task or activity in which the users are engaged to influence tabletop design. In a completely isolated or close intimate space, a user works with personal interaction surfaces [12].

For example, a collaborative sketching system enables users testing UI prototypes on any surface size if the user is completely isolated. When users require a close intimate space, tiny or small surfaces will be used instead because they facilitate face-to-face collaboration and eye-contact [52]. Stakeholders who are not designers or developers usually come to meetings with their own interaction surface (e.g., a smartphone, a tablet) and would like to see the UI prototype on their very right devices, even if they drew something on another, possibly larger, surface. Personal spaces offer more ample possibilities than intimate spaces because users sitting around a table or standing side-by-side can share work-spaces by using multiple displays, large, or extra large desktops, tabletops, and wall screens. The social space takes place in a business-based interaction. Stakeholders and final users could interacting in shared work-spaces by using large tabletop display or wall screens.

For example, a social space is adopted when some kind of multiple user's interactions in taking place in a space between 1.2–3.6 m. Public space allows the user to interact with a sizable audience. In this context, the users make use of multiple display. Small surfaces allow manipulation of the work-space, while large surfaces serve to display the information to the entire audience. Informational wall surfaces, like public displays [36], provide shared views to the users standing far away from the display. Interactive walls enable users to walk up to the display and interleave interaction and discussion among participants. Finally, in remote spaces, users share work-spaces with same or different times. A collaborative sketching system could offer most surface configurations. Stakeholders interact simultaneously with any device. In a collaborative session, users can sketch scenes in a private way or separated into small groups, perform a task privately and then communicate the results to all stakeholders [45].

Fig. 2. CROSSIDE dimensions: (a) the “user” dimension, (b) the “work method” dimension, (c) the “surface” dimension, and (d) the “device” dimension.

They can interact using integrated environments composed of horizontal or vertical displays regardless of the dimensions of these devices.

C. Input Interaction Techniques

The *input interaction techniques* express the sophistication degree with which sketching is supported, ranging from an indirect manipulation to full direct manipulation. This dimension takes into account the use of materials and the naturalness of the interaction. A classical indirect pointing device is the “mouse” because its locus of control (i.e., the physical space in which actions occur) is different from or outside the locus of application (i.e., where the actions are applied). “Pen-based” interaction enables the user to interact with the device by using a passive or active stylus rather than a mouse. While pen devices initially supported indirect manipulation, they now support direct manipulation, namely by touch technology, regardless the fingers or the pointers involved. The “glove” considers the family of lightweight and stretchable devices that combines hand posture sensing and tactile pressure. Manipulation is direct, by device intermediation, like with smartwatches, armbands, and rings. “Tangible” devices naturally offer direct manipulation by connecting objects and surfaces to digital information. Tangible UIs typically work on tabletop surface and embed the

tracking mechanism inside or outside the surface [46]. The “finger” considers situations where the end user interacts directly with the surface by finger tracking. This dimension could be extended to using the whole hand, but requires technologies for hand pose recognition. The last step, i.e. “full body”, involves full-body gesture recognition, but its benefits for sketching are yet to be demonstrated.

D. Work Methods

The *work methods* characterise how interaction surfaces are spatially arranged according to a territoriality in the environment and how these environments are connected together (e.g., through Wi-Fi, LAN, WAN). Interaction surfaces could be *tiled*, *coupled*, *uncoupled*, or *positioned side-by-side* [12]. This dimension subsumes the physical location of each surface and how it is positioned (i.e., vertically, horizontally, in an oblique way). The dimension also considers the devices configurations of the five categories of technologies for collocated collaborative work classified by Wang et al. [52] namely: horizontal displays, large vertical displays, multiple displays, tangible interfaces, and integrated environments. Fig. 2b depicts typical setups of single and multiple users. Users can work alone in their own organisations. Multiple users work in corporate environments [5]: *hallway* (when an environment consists of any informal place where users could meet), *individual in the office* (when an environment only accommodates one user at a time, although this user can change over time), *meeting* (when the environment accommodates several users at a time for conducting a meeting), *get together* (when the environment accommodates several users for collaboration in general), *ongoing interaction* (when different environments involve many different users).

E. Tangibility

The *tangibility* dimension expresses to what extent the materialization of collaboration space is digital, physical, or mixed, ranging from applications in digital environments to the use of shape-changing devices. Digital environments define most classical applications where UI can be 1D (based on lines), 2D (based on surfaces), 2D1/2 (based on a space projected onto a surface) or 3D (in space). 1D UIs are

Fig. 3. User configuration with sessions, stakeholders, and scenarios.

Fig. 4. Properties of physicality, such as plasticity and elasticity.

typically based on command lines or instructions. In digital environments, users interact with a minimal immersion degree. Information is presented as a stream of characters in a text terminal. 2D UIs are incorporated in an environment based on conventional bi-dimensional representations where the interaction techniques are supported by events. Their widgets and their composition enable creating complex interfaces. The 2D1/2 and 3D interfaces extend classic GUIs with the notion of overlap and depth perceptions.

Virtual describes real environments simulated by a computer where users are involved with a high immersion degree. Users interact with virtual objects in an infinite 3D space. On the opposite of the dimension, applications are developed in real environments and use of mechanisms to increase the perception of the user. *Augmented Virtual* are applications including virtual worlds generated by a computer. These applications incorporate virtual reality to replace the physical world and the virtual world predominates over the real. *Augmented Virtual* extends the physical reality perceived by incorporating virtual objects into the physical world, thus increasing the degree of immersion. In contrast, *Augmented Real* considers an otherwise real environment augmented by means of virtual objects [34]. In the *Augmented Real*, virtual objects increase the real world, the dominant medium.

Tangible UIs give physical form to digital information, employing physical artifacts both as representations and controls for computational media [48]. This is further refined into four steps depending how tangible objects are materialized: real tangibility, stable physicality, elastic physicality, and plastic physicality. Real tangibility attempts the reproduce the physical behavior of a real world object into the tangible object, or a sub-set of it. Stable physicality occurs when tangible objects never change their behavior, whether they are expected to mimic some real work or not. Elastic physicality groups all environments that use both physical materials with computational analysis and simulation. These environments are used to understand and represent the world. Plastic physicality occurs in the area of shape-changing UI, where the devices are artifacts whose surface and/or volume can be articulated and modulated with their spatial domain [42]. In physics and materials science, the physicality property is

Fig. 5. Quadrant of the four configurations.

one form of adaptation, which could be decomposed into three major properties (Fig. 4): *plasticity*, which describes the deformation of a material undergoing (non-reversible) changes of shape in response to applied forces; *elasticity*, which is the tendency of solid materials to return to their original shape after being deformed; and *viscosity*, which expresses to what extent a material is resistant with respect to deformation. Solid objects will deform when forces are applied on them; if the object is elastic, it will return to its initial shape and size when these forces are released. These two terms can be used for tangibility.

F. Device Configurations

The *device configurations* refer to the surface size of device independently of their density and orientation. Fig. 2d depicts the different dimensions for the device configurations. The sizes are categorized by generalised dimensions ranging from tiny to wall screen. A system could run on a variety of devices offering different sizes, performing scaling and resizing to accommodate variations induced by different screens. Four typical configurations based on stakeholders (or users) and their surfaces therefore emerge (Fig. 5):

- 1) *Single user-Single device* (SUSD): a single stakeholder, such as an end-user, is working in isolation on one device only to sketch a UI (see figure 5 - top left). This device could be the very right device on which the final UI should run or another one. Therefore, this configuration is appropriate for conducting the sketching, prototyping, and testing activities as defined.
- 2) *Single user-Multiple devices* (SUMD): a single stakeholder is sketching on multiple devices either simultaneously or asynchronously in order to assemble the various sketches into a coherent design scenario (see figure 5 - top right). Therefore, this configuration is appropriate for conducting the sketching, prototyping, and sharing/testing activities.
- 3) *Multiple user-Single Device* (MUSD): several stakeholders share a same device in order to interact together and to come up with an agreement on some design questions (see figure 5 - bottom left). This configuration is expected to mimic the classical whiteboard

configuration where different stakeholders are together in front of a whiteboard and sketching things all together. Therefore, this configuration is appropriate for supporting the sharing/testing and discussing/reflecting activities, while it could be also used for the sketching and prototyping activities, but not primarily.

- 4) *Multiple users-Multiple devices* (MUMD): several stakeholders exploit private and public devices to apply any decided modification on the design scenario (see figure 5 - bottom right). They could perform these actions first on their own private device (e.g., their tablet), and then propagate modifications to the whole scenario (displayed on a wall screen). They straightforwardly interact on the public device to collect immediate feedback from other stakeholders. Therefore, this configuration is appropriate for supporting the sharing/testing and discussing/reflecting activities.

G. Building process of the Design Space

The design space has been built by identifying the greatest common denominator of software features described from forty-one references in various domains. The process followed to build the design space was basically a middle-out approach that combines two sub-processes, while trying to satisfy the principle of separation of concerns regarding the three main aspects of a context of use (e.g., users and their interactive tasks, their platforms and devices, their physical and organisational environments [9]):

- 1) A *bottom-up approach* consisting of browsing each reference at a time, identifying any high-level factor supporting collaborative sketching by separating them for users, platforms, and environments. All values found for each factor were collected in the same set.
- 2) A *top-down approach* consisting of determining such high-level factors by looking after a theoretical definition in the literature.

For example, the “User configurations” factor was identified among several references, therefore encouraging us to find a taxonomy that is relevant enough to express values found for this factor. We discovered a similar concept in Mandviwalla and Olfman [27]. To make it suitable for our purpose, we expanded the possible values from values collected in the references and we adapted the definition accordingly to make it categorical. Fig. 2a graphically represents the resulting factor presented as a progressive dimension.

Another example is related to the “Surface configurations” factor. By extracting from each use case found in every reference, we wanted to categorise the surface projected on the ground that delineates the interaction space. When only one person is involved in a sketching, there are various comfort zones from social psychology. When several persons are involved in a collaborative sketching, the way they interact with their system depends on what they want to do with their sketch. The sketching activity itself is more frequently found in small surfaces, while sharing and discussing the sketching is more frequently found in medium to large surfaces. The scope of the input and the output determines the surface.

Fig. 6. CROSSIDE instantiation for GAMBIT.

IV. INSTANTIATION OF THE DESIGN SPACE

This section first instantiates CROSSIDE on GAMBIT, one of our systems for collaborative sketching, to check how the descriptive virtue is addressed. It then performs the instantiations for the external tools: BELONGINGS [38], FLEXISKETCH [54], [55], [53], CALICO [28], SKETCHML [3] and EVE [47]. For the instantiations of each of the tools, we superimpose the first drawing to address the comparative virtue. At the end, all instances are presented in the same design space allowing to see their similarities and differences.

A. The CROSSIDE instantiation for GAMBIT

GAMBIT¹ (Gatherings And Meetings with Beamers and Interactive Tables) [45] is a freely accessible² multi-platform HTML5 environment for collaborative design by sketching for UI design, but not only, by supporting (Fig. 6):

- *Multi-fidelity*: it should enable stakeholders to provide material at any level of fidelity and to easily switch from one level of fidelity to another.
- *Multi-format*: while sketching remains at the heart of the service, it should accommodate various formats of data both as input and as output.
- *Multi-session*: it should offer the possibility to create, edit, one or many sessions simultaneously.
- *Multi-scenario*: within a single session, it should be able to manage several alternate scenarios at once, namely by representing explicitly the different scenarios and

¹The word “gambit” is used in chess in reference to a movement in which the player sacrifices a piece, usually a pawn, in exchange to some future advantage; in user interface design the gambit would be made by the designer who sacrifices sketches in exchange to know in advance how the interface would function in the hands of the end-users, allowing her to learn from the eventual observation of this usage.

²Accessible at <http://gambitsketch.appspot.com/>

Fig. 7. CROSSIDE instantiation for BELONGINGS superimposed to the instantiation for GAMBIT.

- by linking them. For instance fork and join mechanisms could be used to map scenarios together and create different design paths for idea writing. Different stakeholders make explicit more design ideas so as to generate new design paths that were remained under explored before. This is very important to avoid the “Do not fall in love of your own design” observation: once a design alternative is selected as an official one, some stakeholders tend to keep it (they fall in love of their design) and to spend remaining time in refining this design as opposed to exploring alternate designs.
- **Multi-stakeholder:** a same session could be created to involve a wide community of stakeholders whose roles and responsibilities may evolve over time. When the role or the responsibility of a stakeholder changes depending on the context of the problem, the tool should accommodate these changes flexibly.
 - **Multi-access:** each stakeholder should contribute to a design session with her own devices. End-users as well as designers tend to create smaller sketches on small screen devices and larger sketches when the interaction surface increases, without necessarily adding more details. Multi-device is equally important to support cross-device interaction, e.g., when a stakeholder may switch from one device to another while participating in the same session, but with different roles.

B. The CROSSIDE instantiation for BELONGINGS

BELONGINGS [38] consists of an interactive tangible tabletop surrounded by multiple interactive surfaces and smaller devices for collaborative query by sketching. The table was installed in a museum and designed to communicate indigenous traditional knowledge and cultural values. Fig. 7

Fig. 8. CROSSIDE instantiation for FLEXISKETCH on GAMBIT.

depicts the design space instantiated for this environment superimposed on the previous one (Fig. 6) to compare the coverage of both tools. BELONGINGS can be operated in a fixed tabletop by two users working together in a social context. Unlike GAMBIT, BELONGINGS supports the use of tangible devices, thus being better with that respect on the dimension. Both software share a participatory design [37] approach as part of the development process: designers and final users work together to produce the final application.

C. The CROSSIDE instantiation for FLEXISKETCH

FLEXISKETCH [54], [55], [53] comes in two versions: the first uses tablets as sketching media and supports inexpensive, mobile sketching at any time and in any place, while the second version, known as “FlexiSketch Desktop”, runs on electronic whiteboards and provides a wide screen for collocated meetings. Multiple tablets can be connected to the desktop version over Wi-Fi, thus enabling end users to collaborate and simultaneously work in the same workspace regardless the multiple screens they are using. FLEXISKETCH enables participants to edit the workspace simultaneously, offers shared and private views, and allows group work and/or facilitating individual work. Fig. 8 depicts how FlexiSketch is instantiated on the design space. As a way for comparison, the instantiation of Gambit is also superimposed. FlexiSketch and Gambit have the similarities of considering the participatory design as part of the development process. Both tools allow multiple concurrent accesses through the simultaneous use of multiple devices.

D. The CROSSIDE instantiation for CALICO

CALICO [28] is a free hand rapid design tool supporting early software design activities to be used with touch screen interfaces, such as interactive whiteboards and tablet PCs.

Fig. 9. CROSSIDE instantiation for CALICO superimposed to the instantiation for GAMBIT.

CALICO enables designers creating designs on multiple canvases based on a client-server architecture, supporting up to 20 simultaneously active users [30]. A CALICO client is portable, supporting computers connected to electronic whiteboards, laptops, and tablets. Thus, CALICO supports collaborative work across multiple devices, allowing multiple designers to work synchronously on the same canvas or asynchronously on different canvases. This allows designers working in a group to branch off to their own canvas. Fig. 9 combines the instantiations of CALICO and GAMBIT.

E. The CROSSIDE instantiation for SKETCHML

SKETCHML [3] is a framework that offers the ability to define and recognize every kind of 2D graphical library, by using freehand drawing, to be used in the construction of user interfaces. The framework uses an empirical language based on XML. Its language allows the compatibility of SketchML with other applications and services through various devices. Fig. 10 combines the instantiations of SKETCHML and GAMBIT.

F. The CROSSIDE instantiation for EVE

EVE [47] is a Sketch-based prototyping workbench that facilitates end-users to define their design through a set of low-fidelity sketches. The Low-fidelity representations are recognized and translated in medium fidelity representations, as well as in high fidelity prototypes. End-users realize the representations in a canvas of two dimensions. End-users can navigate through the three levels of loyalty. At each level it is possible to make the desired changes. For each of the fidelity, end-users can operate three modes. The design functionality, the configuration of the interaction and the preview of the prototype. Fig. 12 combines the instantiations of EVE and GAMBIT.

Fig. 10. CROSSIDE instantiation for SKETCHML superimposed to the instantiation for GAMBIT.

G. The comparative virtue of CROSSIDE

Fig. 13 combines all the instantiations of tools studied insofar in a radar diagram to facilitate the visual comparison of the tools (provided that they are not too numerous), such as the similarities and differences. The goal is to satisfy the comparative virtue of the design space, not to promote an ideal tool being the best along all dimensions. Rather, some tools are more advanced along some dimensions, say the “Device configurations”, while others are targeting more flexibility for other dimensions, like “User configurations”. For example, the range of “Device configurations” is well covered by both GAMBIT and CALICO, up to the three first steps of “Work methods”. Fig. 11 illustrates the “Meeting” work method, where two designers share a tabletop for sketching a prototype by pen that is rendered in real-time on the smartphone of an end-user. Sketching is in direct manipulation: the designer is sketching directly on the tabletop and manipulating the sketch in the same way.

Fig. 11. Meeting work method for GAMBIT.

Fig. 12. CROSSIDE instantiation for EVE superimposed to the instantiation for GAMBIT.

H. The generative virtue of CROSSIDE

A close examination of Fig. 13 leads to the observation that some dimensions are pretty well covered, like “Device configurations” and “Surface configurations”, some others are moderately covered like “Layout”, “Work methods”, and “Input interaction techniques” while the remaining ones are limited, such as “User configurations” and “Tangibility”. This suggests that these dimensions are welcome to be explored in further activities of a research agenda. Indeed, tangibility is very limited in all tools examined so far: most of them represent digital solutions where sketching is achieved on a 2-D surface with limited beautification performed in this space. There are probably other tools for collaborative sketching in 3D exhibiting the capability to virtually augment the real world, but they do not belong to our initial list of references. For the practitioners, this means that no tool is available today to fulfill these needs.

The design space obtained so far only reflects some significant dimensions identified among a set of forty-one references considered as representative instances of collaborative sketching in various domains of human activity, ranging from learning to industrial design. While this set of references covers several domains, we do not argue that its coverage is complete or representative enough of the vast majority of tools of interest.

Therefore, our next step consists of conducting a Systematic Literature Review (SLR) [22] for identifying references relevant to collaborative sketching for multiple purposes:

- 1) To expand the coverage of reviewed works from our 41 selected references to a larger panel.
- 2) To address the reproducibility of the procedure for guaranteeing the coverage of the design space.
- 3) To address explicitly the generative virtue by dis-

Fig. 13. CROSSIDE instantiations for BELONGINGS, FLEXISKETCH, CALICO, SKETCHML, EVE and GAMBIT all combined at once.

covering the most promising configurations on this design space which may serve for a research agenda in the near future.

For the moment, we can superimpose the instantiations of the design space performed for the 41 references. On one hand, this superimposition enables us to identify portions of each dimension that are more or less frequently covered, or not covered at all. But this analysis considers only one dimension at a time, which may be considered as reductive. On the other hand, the superimposition also enables us to identify the configurations that are the most or the least frequently adopted by tools. This does not mean that they are appropriate or not, but simply the coverage could be discussed. We prefer to perform this analysis on a set of references resulting from the SLR instead of our initial set.

V. CONCLUSION AND FUTURE WORK

We presented CROSSIDE, a design space for representing capabilities of a software for collaborative sketching in a cross-surface setting. This design space consists of seven dimensions (i.e., user configurations, surface configurations, input interaction techniques, work methods, tangibility, layout, and device configurations) resulting from a comparative analysis of 41 references in the domain. Each instantiation of these 41 references on the design space is graphically depicted as a radar diagram, which visually supports three virtues: descriptive, comparative, and generative.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their constructive and patient comments on earlier versions of this manuscript.

REFERENCES

- [1] S. W. Ambler. Agile adoption rate survey results: March 2007, 2007. [Online; accessed 05-April-2019].
- [2] D. Avola, L. Cinque, and G. Placidi. Sketchspore: A sketch based domain separation and recognition system for interactive interfaces. In A. Petrosino, editor, *Image Analysis and Processing – ICIAP 2013*, pages 181–190, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [3] D. Avola, A. Del Buono, G. Gianforme, S. Paolozzi, and R. Wang. Sketchml a representation language for novel sketch recognition approach. In *Proceedings of the 2Nd International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '09*, pages 31:1–31:8, New York, NY, USA, 2009. ACM.
- [4] R. Bellamy, M. Desmond, J. Martino, P. Matchen, H. Ossher, J. Richards, and C. Swart. Sketching tools for ideation (nier track). In *33rd Int. Conference on Software Engineering, ICSE '11*, pages 808–811, New York, NY, USA, 2011. ACM.
- [5] E. Berglund and M. Bång. Requirements for distributed user interface in ubiquitous computing networks. In *Proc. of Conf. on Mobile and Ubiquitous MultiMedia*, 2002.
- [6] J. Bowen and A. Dittmar. A semi-formal framework for describing interaction design spaces. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '16*, pages 229–238, New York, NY, USA, 2016. ACM.
- [7] T. Buchmann. Towards tool support for agile modeling: Sketching equals modeling. In *Proceedings of the 2012 Extreme Modeling Workshop, XM '12*, pages 9–14, New York, NY, USA, 2012. ACM.
- [8] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
- [9] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, 06 2003.
- [10] G. Costagliola, V. Deufemia, and M. Risi. Sketch grammars: A formalism for describing and recognizing diagrammatic sketch languages. In *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea*, pages 1226–1231. IEEE Computer Society, 2005.
- [11] G. Costagliola, M. D. Rosa, and V. Fuccella. Local context-based recognition of sketched diagrams. *J. Vis. Lang. Comput.*, 25(6):955–962, 2014.
- [12] J. Coutaz, C. Lachenal, and S. Dupuy-Chessa. Ontology for multi-surface interaction. In M. Rauterberg, M. Menozzi, and J. Wesson, editors, *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*. IOS Press, 2003.
- [13] A. Coyette, S. Faulkner, M. Kolp, Q. Limbourg, and J. Vanderdonckt. SketchiXML: Towards a multi-agent design tool for sketching user interfaces based on UsiXML. In *Proc. of 3rd Int. Conf. on Task Models and Diagrams, TAMODIA '04*, pages 75–82, NY, 2004. ACM.
- [14] A. Coyette, S. Kieffer, and J. Vanderdonckt. Multi-fidelity prototyping of user interfaces. In *Proc. of 11th IFIP*.
- [15] B. S. da Silva, V. C. O. Aureliano, and S. D. J. Barbosa. Extreme designing: binding sketching to an interaction model in a streamlined HCI design approach. In *7th Brazilian symposium on human factors in cs*, pages 101–109. ACM, 2006.
- [16] V. Genaro Motti, D. Raggett, S. Van Cauwelaert, and J. Vanderdonckt. Simplifying the development of cross-platform web user interfaces by collaborative model-based design. In *Proceedings of the 31st ACM International Conference on Design of Communication, SIGDOC '13*, pages 55–64, New York, NY, USA, 2013. ACM.
- [17] C. Gonzalez-Perez. Filling the voids - from requirements to deployment with open/metis. In *Proc. of 5th Int. Conf. on Software and Data Technologies, Vol. 1, Athens, Greece, July 22-24, 2010*, page 19, 2010.
- [18] S. Greenberg, S. Carpendale, N. Marquardt, and B. Buxton. *Sketching User Experiences - The Workbook*. Academic Press, 2012.
- [19] T. Hammond and R. Davis. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *AAAI Spring Symposium on Sketch Understanding*, pages 59–68, 2002.
- [20] M. Johansson and M. Arvola. A case study of how user interface sketches, scenarios and computer prototypes structure stakeholder meetings. In *Proc. of the 21st British HCI Group Annual Conference on People and Computers, vol. 1, BCS-HCI '07*, pages 177–184, Swinton, UK, 2007. British Computer Society.
- [21] G. Johnson, M. D. Gross, J. Hong, and E. Y. Do. Computational support for sketching in design: A review. *Foundations and Trends in Human-Computer Interaction*, 2(1):1–93, 2009.
- [22] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman. Systematic literature reviews in software engineering: a tertiary study. *Information and Software Technology*, 52(8):792 – 805, 2010.
- [23] J. A. Landay and B. A. Myers. Interactive sketching for the early stages of user interface design. In *Proc. of the CHI'95*, pages 43–50.
- [24] C. Lewis, P. G. Polson, C. Wharton, and J. Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proc. of the ACM Conf. on Human Factors in Computing Systems, CHI '90*, pages 235–242, New York, USA, 1990. ACM.
- [25] J. Lin and J. A. Landay. Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, pages 1313–1322, New York, NY, USA, 2008. ACM.
- [26] D. Loksa, N. Mangano, T. D. LaToza, and A. v. d. Hoek. Enabling a classroom design studio with a collaborative sketch design tool. In *Proc. of the Int. Conf. on Software Engineering, ICSE '13*, pages 1073–1082, Piscataway, NJ, USA, 2013. IEEE Press.
- [27] M. Mandviwalla and L. Olfman. What do groups need? a proposed set of generic groupware requirements. *ACM Trans. Comput.-Hum. Interact.*, 1(3):245–268, Sept. 1994.
- [28] N. Mangano, A. Baker, and A. Van Der Hoek. Calico: a prototype sketching tool for modeling in early design. In *Proceedings of the 2008 international workshop on Models in software engineering*, pages 63–68. ACM, 2008.
- [29] N. Mangano, M. Dempsey, N. Lopez, and A. van der Hoek. A demonstration of a distributed software design sketching tool. In *Proc. of the 33rd Int. Conf. on Software Engineering, ICSE '11*, pages 1028–1030, New York, NY, USA, 2011. ACM.
- [30] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek. Supporting informal design with interactive whiteboards. In *Proc. of the ACM Conf. on Human Factors in Computing Systems*, pages 331–340, 2014.
- [31] D. McCrickard. *Making Claims: The Claim as a Knowledge Design, Capture, and Sharing Tool in HCI*. Morgan & Claypool, 2012.
- [32] P. McInerney and F. Maurer. Ucd in agile projects: dream team or odd couple? *interactions*, 12(6):19–23, 2005.
- [33] J. Melchior, J. Vanderdonckt, and P. Van Roy. A model-based approach for distributed user interfaces. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '11*, pages 11–20, New York, NY, USA, 2011. ACM.
- [34] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [35] R. Mitra. Rapido: A sketching tool for web api designers. In *24th Int. Conference on World Wide Web, WWW '15 Companion*, pages 1509–1514, Republic and Canton of Geneva, Switzerland, 2015. Int. World Wide Web Conf. Steering Committee.
- [36] J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and design space for interactive public displays. In *Proceedings of the 18th ACM International Conference on Multimedia, MM '10*, pages 1285–1294, New York, NY, USA, 2010. ACM.
- [37] M. J. Muller and S. Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, June 1993.
- [38] R. Muntean. Considering collaboration in ?elewkw—belonging. In *WorkShop Cross-Surface 2016: Third International Workshop on Interacting with Multi-Device ecologies" in the wild"*, 2016.
- [39] v. Obrenovic and J.-B. Martens. Sketching interactive systems with sketchify. *ACM Trans. Comp.-Hum. Interact.*, 18(1):1–38, May 2011.
- [40] H. Ossher, A. van der Hoek, M.-A. Storey, J. Grundy, and R. Bellamy. Flexible modeling tools (flexitools2010). In *32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pages 441–442. ACM, 2010.
- [41] J.-L. Pérez-Medina, S. Dupuy-Chessa, and A. Front. A survey of model driven engineering tools for user interface design. In M. Winckler, H. Johnson, and P. Palanque, editors, *Task Models and Diagrams for User Interface Design*, volume 4849 of *LNCS*, pages 84–97. Springer Berlin Heidelberg, 2007.
- [42] M. K. Rasmussen, G. M. Troiano, M. G. Petersen, J. G. Simonsen, and K. Hornbæk. Sketching shape-changing interfaces: Exploring vocabulary, metaphors use, and affordances. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16*, pages 2740–2751, New York, NY, USA, 2016. ACM.
- [43] M. D. Rosa, A. Fish, V. Fuccella, R. Saleh, S. Swartwood, and G. Costagliola. A toolkit for knot diagram sketching, encoding and re-generation. In G. Polese and V. Deufemia, editors, *The 22nd International Conference on Distributed Multimedia Systems, DMS 2016, Salerno, Italy, November 25-26, 2016.*, pages 16–25. KSI Research Inc. / Knowledge Systems Institute Graduate School, 2016.
- [44] P. Sachse, W. Hacker, and S. Leinert. External thoughtdoes sketching

- assist problem analysis? *Applied Cognitive Psychology*, 18(4):415–425, 2004.
- [45] U. B. Sangiorgi, F. Beuven, and J. Vanderdonckt. User interface design by collaborative sketching. In *Proceedings of the Designing Interactive Systems Conference, DIS '12*, pages 378–387, New York, NY, USA, 2012. ACM.
 - [46] O. Shaer and E. Hornecker. Tangible user interfaces: past, present, and future directions. *Foundations and Trends in Human-Computer Interaction*, 3(1–2):1–137, 2010.
 - [47] S. Suleri, V. P. Sermuga Pandian, S. Shishkovets, and M. Jarke. Eve: A sketch-based software prototyping workbench. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA '19*, pages LBW1410:1–LBW1410:6, New York, NY, USA, 2019. ACM.
 - [48] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM systems journal*, 39(3.4):915–931, 2000.
 - [49] R. van der Lugt. Functions of sketching in design idea generation meetings. In *Proc. of the 4th Conf. on Creativity & Cognition, C&C '02*, pages 72–79, New York, USA, 2002. ACM.
 - [50] J. Vanderdonckt and A. Beirekdar. Automated web evaluation by guideline review. *J. Web Eng.*, 4(2):102–117, June 2005.
 - [51] J. R. Wallace and S. D. Scott. Contextual design considerations for co-located, collaborative tables. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 57–64. IEEE, 2008.
 - [52] H. Wang and E. Blevis. Concepts that support collocated collaborative work inspired by the specific context of industrial designers. In *Proc. of the ACM Conf. on Computer Supported Cooperative Work, CSCW '04*, pages 546–549, New York, USA, 2004. ACM.
 - [53] D. Wüest, N. Seyff, and M. Glinz. Collaborative sketching and notation creation with FlexiSketch Team. *Software Engineering 2017*.
 - [54] D. Wüest, N. Seyff, and M. Glinz. Flexisketch: A mobile sketching tool for software modeling. In D. Uhler, K. Mehta, and J. Wong, editors, *Mobile Computing, Applications, and Services*, volume 110 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 225–244. Springer, 2013.
 - [55] D. Wuest, N. Seyff, and M. Glinz. Flexisketch team: Collaborative sketching and notation creation on the fly. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE Int. Conf. on*, volume 2, pages 685–688, May 2015.
 - [56] M. C. Yang. Observations on concept generation and sketching in engineering design. *Research in Engineering Design*, 20(1):1–11, 2009.

C-QWERTY: a text entry method for circular smartwatches

Gennaro Costagliola, Mattia De Rosa, Vittorio Fuccella
Dipartimento di Informatica, University of Salerno
Via Giovanni Paolo II, 84084 Fisciano (SA), Italy
{gencos, matderosa, vfuccella}@unisa.it

Raffaele D'Arco, Sabato De Gregorio, Daniele Lupo
University of Salerno
Via Giovanni Paolo II, 84084 Fisciano (SA), Italy

Abstract

We present a study aimed at comparing different circular layouts for entering text on smartwatches. In particular, we measured the extent through which the use of a QWERTY layout increases user performance in the earliest sessions of use. To this aim, we designed C-QWERTY, a soft keyboard designed for circular smartwatches in which the keys are arranged along the edge of the screen in a circular layout. In order to make the keyboard more familiar to users, the order of the keys is similar to the one of traditional rectangular QWERTY keyboards. The method supports two interaction modes: tapping and gesture, in which a whole word can be written with a single gesture. As an evaluation, we compared the C-QWERTY layout with Cirrin, another circular layout in which the order of the characters has been optimized to minimize the distance between successive keys during writing. The experimental results showed that the C-QWERTY, with a text entry speeds of 9.1 wpm and 7.7 wpm (tapping and gesture mode, respectively), outperformed the Cirrin layout with entry speeds of 6.6 wpm and 5.5 wpm. The increase in text entry speed due to the use of the QWERTY layout was 38 percent.

Keywords: Text entry; Smartwatch; Gestures; Touchscreen.

1. Introduction

Smartwatches have recently emerged as new wearable computing devices, but the small size of the screen may however, in some cases, make it difficult to interact with them. One challenge in this area is text entry, and researchers are introducing new text entry techniques for

smartwatches in order to increase efficiency [1, 5].

Interestingly, many of them are adopting the QWERTY layout. This is probably done to exploit users' previous knowledge, as most of them use the QWERTY keyboard habitually, and to avoid that users have to invest time to learn a new layout. However, when the full QWERTY layout is fitted on a smartwatch screen the keys become very small. To solve this problem various techniques have been developed, as described in the next section.

In any case, a rectangular layout is more suitable for screens of the same type, while in a circular display, as in smartwatches, it can waste a lot of screen space. In this case a circular layout could allow a more efficient use of the screen space, but it is obviously very different from the QWERTY layout already known by users.

The purpose of this paper is to present a study on circular layouts for text entry on smartwatches. For this purpose we have decided to measure how much a QWERTY-like circular layout can improve user performance, especially at the first sessions of use. For this reason we have designed C-QWERTY, a soft keyboard for circular smartwatches in which the keys are arranged along the edge of the screen in a circular layout, and the keys have similar order of the ones of a traditional rectangular QWERTY keyboards. C-QWERTY also supports two types of interaction, tapping and gesture, in which a whole word can be written with a single gesture. To evaluate the performance of the C-QWERTY layout we therefore decided to compare it, through a user study, with another circular layout called Cirrin, in which the order of the characters has been optimized to minimize the distance between successive keys during writing.

The paper is organized follows: Section 2 describes previous work on text entry on smartwatches and on circular layouts; Section 3 describes our layout and text entry method, Section 4 shows its experimental evaluation and

Section 5 the evaluation results. Finally, Sections 6 concludes the paper with a discussion on future work.

2. Related Work

As in the case of smartphones and tablets [13, 25, 19, 7, 6], numerous text entry techniques have been proposed to facilitate text entry on smartwatches. To solve the problem caused by the smartwatches small screens, a common approach is to introduce a further interaction step on soft keyboards. In SplitBoard [12] only a part of the keyboard is shown at any given time and a flick is used to change the displayed part. In ZoomBoard [18], instead, a zooming step is used to increase key sizes. In ZShift [14] a call-out is used to show a zoomed copy of the screen under the user's finger. Such interaction steps, however, increase the time required to select each key. Another approach to the problem is to use gestures to select keys. SwipeBoard [4] is a eyes-free text entry method in which two swipes are used to enter each character. DualKey [10] uses keys with two letters and tapping gestures with two different fingers to choose between them.

Other techniques instead use a word-based (e.g. Watch-Writer [9]) or a sentence-based (e.g. VelociTap [24]) text entry approach.

It can be noted that all of above mentioned techniques are based on rectangular (QWERTY) keyboards. A rectangular layout, however, is naturally more suitable for screens of the same type. On a circular screen, instead, this layout can cause a waste of screen space. In this case a circular layout could allow a more efficient use of the screen space. In fact, in addition to the common rectangular layouts, there are also some keyboards with circular layouts, that are used to exploit such layout advantages in scenarios such as rotational interfaces [20] or personal area on the tabletop [2].

One of the first circular layout to be proposed is Cirrin, a text entry method for pen input devices based on a soft keyboard proposed by Mankoff and Abowd in 1998 [17]. Here, the keys are arranged in a circle, as shown in Figure 1, and the layout of the letters in the circle is such as to minimize the average distance traveled by the pen to write a word. This is because the user, to insert a word, simply draws a path, starting from the inside of the circle, that crosses the circumference in points corresponding to the characters of the desired word, in the right order. A space character is automatically inserted when the pen is raised. Cirrin has shown to have the potential to be faster than the classic QWERTY virtual keyboards and is particularly suitable for pen based devices.

Some techniques use characters in alphabetical order. Shoemaker et al. [21] proposed a circular keyboard for entering text on large wall displays. The selection is performed by moving the pointer over the desired character and

Figure 1. Cirrin Layout.

by pressing a button to enter it. BubbleCircle [11] uses a ring arrangement of alphabetic characters on a tabletop display and magnifies the next most probable characters based on the user's input. TUP [20] is a text entry method for touch sensitive wheels. The characters are positioned on fixed positions on the wheel and the user presses a select key to enter the highlighted character.

3. C-QWERTY

Our layout, C-QWERTY, is a keyboard layout for circular smartwatches in which the alphanumeric keys are arranged along the edge of the screen in a circular layout. In order to make the keyboard more familiar to users, the keys are arranged in a similar way to that of traditional QWERTY keyboards, as shown in Figure 2.

In particular the keys corresponding to the first row of the QWERTY keyboard have been positioned at the top of the screen (from right to left: q, w, e, r, t, y, u, i, o, p), while those corresponding to the third keyboard row have been positioned at the bottom of the screen (from right to the left: z, x, c, v, b, n, m). In the remaining space on the sides, in the central part of the screen, the keys corresponding to the second keyboard row have been positioned partly on the left (from top to bottom: a, s, d, f) and partly on the right (from top to bottom: g, h, j, k, l).

In order to make the best use of the available screen space, our C-QWERTY soft keyboard implementation has on the center of the screen a text field that shows the transcribed text. Above this field there are the space and backspace keys, while below it there are up to 4 suggestion words in order to complete the current written word. This functionality uses a 20K words dictionary.

Figure 2. C-QWERTY Layout.

The keyboard has two interaction modes for entering text: tapping and gesture. In the first mode, by tapping on a key it is possible to enter the corresponding character. In gesture mode, in addition to tapping, it is also possible to enter an entire word with a single gesture. This is accomplished by dragging the finger in sequence over the alphanumeric keys corresponding to the characters of the desired word. As soon as the finger passes over a key, the system enters the corresponding character, and also automatically adds a space when the user raises his finger at the end of the gesture.

The space and backspace keys and the suggestion words can be used in both modes only by tapping.

4. Evaluation

We carried out a user-study aimed at evaluating the speed and accuracy of the C-QWERTY layout. To this aim we decided to compare it with the Cirrin layout described in Section 2 and shown in Figure 1.

In the experiment we asked the participants to transcribe some sentences using both layouts (C-QWERTY and Cirrin) and both interactions modes (tapping and gesture).

4.1. Participants

For the experiment, 12 participants (2 female) were recruited. They were all university students between 20 and

Figure 3. A participant during the experiment.

28 years old ($M = 23.4$, $SD = 2.4$) and chose to participate for free. All were usual users of computers and smartphones, while most of them had little experience with smartwatches. Most of them also declared a good English knowledge. Figure 3 shows a participant during the experiment.

4.2. Apparatus

The experiment was conducted on a Ticwatch Pro equipped with a Snapdragon Wear 2100 Quad Core 1.2 Ghz processor and running the Wear OS operating system. The device weighs 58.5 grams and has a circular display with a 1.39" diagonal and a resolution of 400×400 pixels.

The experimental software is a Wear OS application that implements the two layouts (C-QWERTY and Cirrin) and the two interaction modes (tapping and gesture).

At startup the application asks the user to choose the desired layout and interaction mode. After this selection the application shows for a few seconds the sentence to be transcribed, after which the keyboard that the participant must use to transcribe the sentence is shown. During this phase the sentence is also shown on a computer screen, in order to allow the participant to re-look at it if necessary. After writing the sentence the participant can confirm it by performing a long press over the text field at the center of the screen, after which the system shows the next sentence (or asks the user to transcribe again the same sentence, if he has exceeded the 15% of non corrected errors).

A personal computer with a 19" monitor was used to show the participant the current sentence.

4.3. Procedure

Before starting the experiment, the participants filled out a questionnaire with the following information: personal data (age, gender), dominant hand, the hand with which they actually performed the experiment, previous experiences with smartwatches and with text entry on smartwatches, level of proficiency with the English language.

The experiment was conducted in a well-lit laboratory. The participants were given the watch and told to wear it on the wrist where it was most convenient for them and possibly rest their arm on a desk. Moreover, during the entire experiment they remained seated. Then, they had a short practice session in which the text entry method was explained and tested (fixed example sentences were shown on the pc monitor). The participants were given all the recommendations related to the experiment, and in particular to:

- read and memorize the sentence before starting to copy it;
- balance speed and accuracy when writing;
- correct mistakes made while entering text. Since the only way to correct errors is by using the backspace key, they were also told to avoid correcting errors noticed only after having already written other words.

The measured tasks started after the participants understood the procedure. The task was to transcribe short text sentences. Each participant had to enter six sentences in each of the four test conditions (the first sentence was for training and therefore not measured). The sentences were randomly chosen from the MacKenzie and Soukoreff set [16] that do not include punctuation or numbers. The participants were allowed a rest period of a few minutes at the end of each test condition.

At the end of the experiment the participants were asked to complete a System Usability Scale (SUS) [3] questionnaire for each layout (C-QWERTY e Cirrin). SUS includes ten statements, to which respondents had to specify their level of agreement using a five-point Likert scale. The questions alternate between positive and negative (since they are in a rather standard form we do not include them here). Each SUS questionnaire has a score between 0 and 100, of which we then calculated the averages on all participants.

In addition to SUS, we also asked for the preferred layout (C-QWERTY or Cirrin), the preferred interaction method (tapping or gesture) and the reasons for these choices. We also collected further feedback through an open form and verbal interaction.

Participants	Orderings			
	QWERTY-TAPPING	QWERTY-GESTURE	CIRPIN-TAPPING	CIRPIN-GESTURE
1, 5, 9	QWERTY-TAPPING	QWERTY-GESTURE	CIRPIN-TAPPING	CIRPIN-GESTURE
2, 6, 10	CIRPIN-TAPPING	CIRPIN-GESTURE	QWERTY-TAPPING	QWERTY-GESTURE
3, 7, 11	QWERTY-GESTURE	QWERTY-TAPPING	CIRPIN-GESTURE	CIRPIN-TAPPING
4, 8, 12	CIRPIN-GESTURE	CIRPIN-TAPPING	QWERTY-GESTURE	QWERTY-TAPPING

Table 1. Counterbalancing used during the experiment.

4.4. Design

The experiment was a two-factor within-subjects design. The factors were the Layout and the Interaction mode. The Layout included two levels: C-QWERTY and Cirrin, while the Interaction mode included two levels: tapping and gesture.

As dependent variables we included:

- Speed: text entry speed measured in words per minute (wpm) as specified in [15].
- Accuracy: the text entry accuracy using both the Total Error Rate (TER) and the Non Corrected Error Rate (NCER), calculated as specified in [23].
- GPC (gestures per character): the average number of gestures needed to enter a character (each interaction with the touchscreen is counted as a gesture).

We counterbalanced the two factors, as shown in Table 1.

5. Results

All participants completed the experiment. One participant was unable to successfully use the gesture interaction and basically used tapping when performing the two gesture tasks. For each participant the experiment lasted about 30 minutes. We tested significance using an analysis of repeated variance measures (ANOVA) [8].

5.1. Speed

The text entry speeds are shown in Figure 4. The grand mean for it was 7.2 wpm. C-QWERTY was the fastest layout with an average of 8.4 wpm, outperforming Cirrin at 6.1 wpm. Regarding the interaction mode, tapping was fastest with an average of 7.9 wpm, outperforming gesture at 6.6 wpm. Accordingly, the highest speeds were obtained by C-QWERTY/tapping at 9.1 wpm and C-QWERTY/gesture

Figure 4. Speeds (in wpm). Error bars show the standard deviation.

Figure 6. Non corrected error rates. Error bars show the standard deviation.

Figure 5. Total error rates. Error bars show the standard deviation.

at 7.7 wpm, while the slowest speeds were obtained by *C-Cirrin/tapping* at 6.6 wpm and *Cirrin/gesture* at 5.5 wpm.

This is probably because in the C-QWERTY mode, the participants could immediately find the position of the letters, given the similarity with the well known QWERTY layout. Moreover the gesture mode is slower than the tapping mode, probably due the fact that some keys are hidden by the hand when performing gestures.

From the ANOVA resulted that the main effect of the layout on the speed was highly significant ($F_{1,11} = 36.546$, $p = .0001$). There was also a significant effect for input mode ($F_{1,11} = 22.865$, $p = .0006$). The interaction between layout and interaction mode was not statistically significant ($F_{1,11} = 0.310$, ns).

5.2. Accuracy

Average values for TER and NCER are shown in Figure 5 and Figure 6.

The grand mean for TER was 19.5%. There was little difference between layouts, with *C-QWERTY* at 19.6% and *Cirrin* at 19.4%. Regarding the interaction mode, *tapping* was more accurate with an average of 18.8%, while *gesture* reached 20.2%. The lowest TER was obtained by *C-QWERTY/tapping* at 18.6%, while the highest by *C-QWERTY/gesture* at 20.6%. However, from ANOVA resulted that there was not a statistically significant difference for the layout ($F_{1,11} = 0.010$, ns), the interaction mode ($F_{1,11} = 0.265$, ns), and the interaction between layout and interaction mode ($F_{1,11} = 0.208$, ns).

The grand mean for NCER was 1.9%. There was little difference between layouts, with *C-QWERTY* at 2.2% and *Cirrin* at 1.7%. Regarding the interaction mode, *tapping* was more accurate with an average of 1.6%, while *gesture* reached 2.3%. The lowest NCER was obtained by *Cirrin/tapping* at 1.5%, while the highest by *C-QWERTY/gesture* at 2.8%. However, from ANOVA resulted that there was not a statistically significant difference for the layout ($F_{1,11} = 1.315$, ns), the interaction mode ($F_{1,11} = 2.151$, ns), and the interaction between layout and interaction mode ($F_{1,11} = 0.455$, ns).

5.3. Gestures per Character

The GPCs are shown in Figure 7. The grand mean for it was 0.94. The *C-QWERTY* layout had the lower value at 0.88, with *Cirrin* at 0.99. Regarding the interaction mode, as expected *gesture* had the lower value at 0.77, with *tapping* at 1.10. Accordingly, the lowest value was obtained by *C-QWERTY/gesture* at 0.67, with the highest value obtained by *C-Cirrin/tapping* at 1.11.

From the ANOVA resulted that the main effect of the layout on GPC was not significant ($F_{1,11} = 1.117$, ns). There was instead a significant effect for input mode ($F_{1,11} = 6.071$, $p = .0315$). The interaction between layout and

Figure 7. Gestures per Character. Error bars show the standard deviation.

interaction mode was not statistically significant ($F_{1,11} = 0.593$, ns).

5.4. User Satisfaction and Free-form Comments

The average SUS score was 70.0 ($SD = 13.3$) for C-QWERTY and 55.83 ($SD = 12.7$) for Cirrin. A Wilcoxon matched-pairs signed-ranks test [22] performed on SUS scores revealed a statistical significance between the two techniques ($Z = 2.22$, $p < .05$).

The trend on such scores was confirmed by the choice of the preferred layout, with 11 participants that preferred C-QWERTY and only one that preferred Cirrin. Most of them motivated their choice by declaring that the C-QWERTY layout was more intuitive and familiar.

Regarding the interaction mode all participants declared their preference for tapping, stating that in gesture mode it is too difficult to see the keys.

5.5. Discussion

During the experiment, even if most participants had little experience with smartwatches, most participants immediately learned how the method works and showed a fast learning process. To this, it may have contributed the fact that participants were all usual users of smartphones and computers. Moreover, all of them were in the 20-29 age group and university students (some of them even in computer science). This, together with their number (12), may have influenced the obtained speeds in absolute terms, with respect to the general population. However, the relative difference between the different input modes should not have been affected by this.

Most participants preferred the C-QWERTY layout, stating its familiarity as main motivation. This confirms that the

QWERTY layout remains recognizable even after its adaptation to a circular layout. All the participants complained about the gesture mode stating that it is too difficult to see the keys and press them correctly. Some participants also complained about the Cirrin layout stating that the position of the characters confused them.

Most of them also appreciated the word suggestion functionality, stating that it was a fundamental help, since it made possible to transcribe the sentences more quickly.

Many of them have instead complained of the lack of a spell checker and of the fact that the entry of a single wrong character is enough to ensure that the desired word will never be shown as suggestion.

However, since the focus of this study is to compare the two layouts and the two interaction methods, we decided to use a simple suggestion feature and not to implement a spellchecker, so that the differences between them could be more directly assessed. Obviously the introduction of functionalities related to a more advanced language model would allow the increase of the writing speed, and for this reason it would be important to use it when comparing C-QWERTY with other text insertion methods that make use of it.

6. Conclusions and further works

In this paper we presented a study comparing different circular layouts for entering text on smartwatches in order to measure the extent through which the use of a circular QWERTY-like layout increases user performance. In particular, we compared our proposed layout C-QWERTY to Cirrin, a layout in which the order of the characters has been optimized to minimize the distance between successive keys during writing. The experimental results show that the C-QWERTY significantly outperforms the Cirrin layout in the first sessions of use.

Future work will focus on the possibility of improving text entry accuracy and speed on circular smartwatch devices by using a specific language model (providing auto-corrections). Further experiments will focus on measuring the performance when the user experience increases and on comparing C-QWERTY with other text entry methods.

References

- [1] A. S. Arif and A. Mazalek. A survey of text entry techniques for smartwatches. In *Proceedings, Part II, of the 18th International Conference on Human-Computer Interaction. Interaction Platforms and Techniques - Volume 9732*, pages 255–267, Berlin, Heidelberg, 2016. Springer-Verlag.
- [2] R. Blanch and M. Ortega. Rake cursor: Improving pointing performance with concurrent input channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing*

- Systems, CHI '09, pages 1415–1418, New York, NY, USA, 2009. ACM.
- [3] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
 - [4] X. A. Chen, T. Grossman, and G. Fitzmaurice. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 615–620, New York, NY, USA, 2014. ACM.
 - [5] G. Costagliola, M. De Rosa, and V. Fuccella. Handwriting on smartwatches: An empirical investigation. *IEEE Transactions on Human-Machine Systems*, 47(6):1100–1109, Dec 2017.
 - [6] G. Costagliola, M. De Rosa, and V. Fuccella. A technique for improving text editing on touchscreen devices. *Journal of Visual Languages & Computing*, 47:1 – 8, 2018.
 - [7] V. Fuccella, M. De Rosa, and G. Costagliola. Novice and expert performance of keyscratch: A gesture-based text entry method for touch-screens. *IEEE Transactions on Human-Machine Systems*, 44(4):511–523, Aug 2014.
 - [8] E. Girden. *ANOVA: Repeated Measures*. Number No. 84 in ANOVA: Repeated Measures. SAGE Publications, 1992.
 - [9] M. Gordon, T. Ouyang, and S. Zhai. Watchwriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3817–3821, New York, NY, USA, 2016. ACM.
 - [10] A. Gupta and R. Balakrishnan. Dualkey: Miniature screen text entry via finger identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 59–70, New York, NY, USA, 2016. ACM.
 - [11] U. Hinrichs, H. Schmidt, T. Isenberg, M. S. Hancock, and S. Carpendale. Bubbletype: Enabling text entry within a walk-up tabletop installation. 2008.
 - [12] J. Hong, S. Heo, P. Isokoski, and G. Lee. Splitboard: A simple split soft keyboard for wristwatch-sized touch screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1233–1236, New York, NY, USA, 2015. ACM.
 - [13] P. Isokoski. Performance of menu-augmented soft keyboards. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 423–430, New York, NY, USA, 2004. ACM.
 - [14] L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt. Text entry on tiny qwerty soft keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 669–678, New York, NY, USA, 2015. ACM.
 - [15] I. S. MacKenzie. A note on calculating text entry speed. last visit 2019. <http://www.yorku.ca/mack/RN-TextEntrySpeed.html>.
 - [16] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 754–755, New York, NY, USA, 2003. ACM.
 - [17] J. Mankoff and G. D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 213–214, New York, NY, USA, 1998. ACM.
 - [18] S. Oney, C. Harrison, A. Ogan, and J. Wiese. Zoomboard: A diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2799–2802, New York, NY, USA, 2013. ACM.
 - [19] A. Oulasvirta, A. Reichel, W. Li, Y. Zhang, M. Bachynskyi, K. Vertanen, and P. O. Kristensson. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2765–2774, New York, NY, USA, 2013. ACM.
 - [20] M. Proschowsky, N. Schultz, and N. E. Jacobsen. An intuitive text input method for touch wheels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 467–470, New York, NY, USA, 2006. ACM.
 - [21] G. Shoemaker, L. Findlater, J. Q. Dawson, and K. S. Booth. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface 2009*, GI '09, pages 231–238, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.
 - [22] S. Siegel. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, New York, NY, US, 1956.
 - [23] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 113–120, New York, NY, USA, 2003. ACM.
 - [24] K. Vertanen, H. Memmi, J. Emge, S. Reyat, and P. O. Kristensson. Velocitap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 659–668, New York, NY, USA, 2015. ACM.
 - [25] S. Zhai and P. O. Kristensson. The word-gesture keyboard: Reimagining keyboard interaction. *Commun. ACM*, 55(9):91–101, Sept. 2012.

Vector-based, Structure Preserving Stroke Gesture Recognition

Nathan Magrofuoco¹, Paolo Roselli², J.L. Pérez Medina^{1,3}, Jean Vanderdonckt¹, Santiago Villarreal¹

¹ Université catholique de Louvain (UCL). Louvain-la-Neuve, Belgium

² Università degli Studi di Roma “Tor Vergata”. Rome, Italy

³ Intelligent & Interactive Systems Lab, Universidad de las Américas. Quito, Ecuador

Abstract—Unistroke and multistroke gesture recognizers have always striven to reach some robustness with respect to all variations encountered when people issue gestures by hand on touch surfaces or with sensing devices. For this purpose, successful stroke recognizers rely on a gesture recognition algorithm that satisfies a series of invariance properties such as: stroke-order invariance, stroke-number invariance, stroke-direction invariance, position, scale, and rotation invariance. Before initiating any recognition activity, these algorithms ensure these properties by performing several pre-processing operations. These operations induce an additional computational cost to the recognition process, as well as a potential error bias. To cope with this problem, we introduce an algorithm that ensures all these properties analytically instead of statistically based on a vector algebra. Instead of points, the recognition algorithm works on vectors between vectors. We demonstrate that this approach not eliminates the need for these pre-processing operations but also satisfies an entire structure-preserving transformation.

I. INTRODUCTION

Gesture user interfaces [4], [6], [28] are User Interfaces (UIs) fostering a natural interaction by capturing gestures issued by the user and by letting the system to recognize them and turn them into commands. Such UIs are applicable to a wide range of activities, such as menu item selection by gesture [2], hand recognition [13], diagram sketching [7], electronic commerce [3], handwriting [10].

Recognizers exploit a large palette of techniques to recognize stroke gestures [28], such as machine learning [6], data mining [5], template-based matching [23], pattern recognition [11], or Nearest Neighbor Classification (NCC) [1], [9], [12], [13], [16], [19], [25], [24], [27] due to several demonstrated advantages: high recognition rate, low execution time, low memory consumption, reasonable computational complexity, and understandable geometric interpretation.

When a stroke is acquired by an end user on a particular device (e.g., a smartphone, a trackpad, a ring device), it is usually subject to several potential variations which could affect its recognition: variations due to the end user, the input device, the output device, the underlying operating system and its software. Any stroke gesture recognizer is perpetually challenged by coping with these variations and address them by ensuring some properties of invariance, like stroke-order invariance, stroke-number invariance, stroke-direction invariance, position, scale, and rotation invariance [11]. To establish and preserve these properties, recognizers perform

several pre-processing steps (e.g., re-sampling, re-rotation, normalization) before any recognition activity is actually launched, thus consuming some resources. A vector-based recognizer escapes from performing these pre-processing steps, thus saving important resources. Working directly with vectors enables us to be released from the physical coordinates space and calculations, which mix points, lines, angles, surfaces, volumes. To discuss these aspects, this paper proves some invariance properties, leading to a structure-preserving recognizer that is beyond articulation-invariant recognizers. This paper extends [22] as follows: it focuses on structure preservation by elaborating on invariance properties, by adding isotropicity, it provides the mathematical demonstrations of invariance properties, it expresses Rubine features in terms of vectors, and it introduces shape distance for unoriented and oriented curves.

The remainder of this paper is structured as follows: Section 2 discusses related work with a focus on invariance properties; Section 3 introduces a vector-based approach for stroke gesture recognition; Section 4 demonstrates how structure-preserving capability is achieved by ensuring invariance properties algebraically instead of statistically or heuristically; Section 5 concludes this work.

II. RELATED WORK

A stroke gesture recognizer is said to be *invariant* if it remains robust, stable, or independent enough with respect to variations of a gesture to be recognized. Such variations can be classified into two categories: a *global variation* concerns a transformation applied to a gesture considered as a whole (e.g., a scaling, a dilatation) while a *local variation* only concerns a specific part of a gesture (e.g., cropping).

A first invariance property is *training set-invariance*: when the training set is modified (e.g., for adding a new gesture, for removing, or modifying an existing one), the algorithm remains unmodified. For example, *synthetic gestures* [14] enrich a training set without requiring more participants to provide additional samples.

Sampling invariance refers to the independence of a recognizer with respect to the amount of points [11], and therefore strokes between these points. Experiments show that 8 points are sufficient to recognize a sample stroke, but best results are obtained with 32 points [25].

Direction invariance [11], or *isotropicity*, refers to the independence of a recognizer with respect to the direction and ordering of stroke composing a gesture. *Time invariance*

Fig. 1. Geometric interpretation of gesture recognition of a candidate gesture against a reference gesture: (a) 2D distance in SP, (b) 3D bowl of center p_i and radius ϵ in 3D, (c) 2D vector-based LSD, (d) 3D vector-based LSD.

[23] refers to the independence of a recognizer with respect to the time required by a end user to issue a gesture.

Position invariance refers to the independence of a recognizer to recognize a gesture wherever it is located on an interaction surface or in the air. Signer [18], imposed that a gesture should be issued in a given bounding box on a certain surface because recognition is conditioned by a partition of this space into regions.

Scale invariance refers to the independence of a recognizer with respect to size and scale of the gesture. People tend to issue larger gestures on a large surface [26]. UI sketches are larger on a tabletop than on a graphical tablet even if they represent the same concept [7]. Gesture size could also convey a particular meaning: a small, medium, or large arrow expresses a slow, medium, or fast navigation movement.

Rotation invariance refers to the independence of a recognizer to recognize the same gesture whatever its orientation is. People sitting around a circular tabletop need all orientations possible since no reference position is privileged. For instance, PennyPincher [19] in its early version was not rotation-invariant while its successor was [20]. Protractor3D [13] supports 3D gestures in a rotation-invariant way.

Articulation invariance groups the above properties together to refer to a recognizer that does not depend on how end users may produce gestures to give results.

Reflection invariance refers to the independence of a recognizer to recognize the same gesture whatever the face of the interactive surface is, such as on a mirror or a transparent window. Since end users are not generally aware of the influence of their location with respect to the display, they prefer to forget it: two users located on each part of a transparent display collaborate regardless the reflection.

Projection invariance expresses the same property in space: whether a gesture is issued in the vertical, horizontal, or sagittal plane, its recognition remains the same. Projection invariance is more complicated to satisfy because points acquired in a particular system of coordinates may not mesh well with a projection. Several pre-processing steps should be performed: resampling or reparameterization, edgification, normalization, rotation to a reference point.

These pre-processing steps satisfy their corresponding properties, but negatively affect the overall performance, its complexity, and could introduce some approximations and/or error rounding due to these calculations that have nothing to do with the recognition process.

III. VECTOR-BASED GESTURE RECOGNITION

A. Initial Definition

To characterize structure preservation, we refer to *isometricity* as the property of a gesture set to hold a set of n -equally distanced points: $\forall i \in \overline{n-1}, d(p_i, p_{i+1}) = \text{constant}$. A representative example exist when $\|p_{i+1} - p_i\| = \frac{1}{n-1} \sum_{i=1}^n \|p_{i+1} - p_i\|$. We hereby refer to *isochronicity* as the property of a gesture set to hold a set of n equally-timestamped points, i.e., $\forall i \in \overline{n-1}, \|t_i - t_{i+1}\| = \text{constant}$, e.g., $\|t_{i+1} - t_i\| = \frac{1}{n-1} \sum_{i=1}^n \|(t_{i+1} - t_i)\|$. We refer to *isotropicity* as the property of a gesture set to hold a set of stroke having the same orientation, such as left-to-right (L2R). We also define *isoparametrization* as the property of two or more gestures/sets to contain the same amount of points, i.e. $\forall G = \{p_i\}_{i=1,\dots,n}, H = \{q_j\}_{j=1,\dots,m} : m=n$. Two gesture sets could be isoparametrized whether they are isometric or not, isochronic or not, isotropic or not.

A gesture expresses a motion between an initial point and a final point based on features such as position, scale, direction, curvature, pressure, tangential acceleration, to be extracted and classified [5], [21]. A point-based representation of a gesture has the advantage of significantly simplifying a gesture to a series of points, thus reducing the gesture recognition to a comparison of two series of points. This approach works well for static shapes, like symbols, letters, simple commands, where shape is more important than motion.

B. Definition of a Basic Gesture

The continuous trace of two consecutive non-trivial translations of a point will be called *basic gesture*. A basic gesture in a finite dimensional affine space can then be formalized by a ordered couple $(\vec{u}, \vec{v})$ of two non-zero free vectors $\vec{u} \in \mathbb{R}^n$ and $\vec{v} \in \mathbb{R}^n$. A basic gesture $(\vec{u}, \vec{v})$ generates a precise oriented triangle, whose third oriented side is the free vector $-(\vec{u} + \vec{v})$. However, a well precise (possibly non trivial) triangle, having points A, B and C as distinct vertices can be generated by six possibly different basic gestures. If we denote $\vec{a} = C - B$, $\vec{b} = A - C$, $\vec{c} = B - A$, then we have six basic gestures: $(\vec{c}, \vec{a})$, $(\vec{a}, \vec{b})$, $(\vec{b}, \vec{c})$, $(-\vec{a}, -\vec{c})$, $(-\vec{c}, -\vec{b})$, and $(-\vec{b}, -\vec{a})$, respectively. In particular, basic gestures in an affine plane correspond to ordered couples $(\vec{u}, \vec{v})$ of two non-zero free vectors $\vec{u} = (u_1, u_2) \in \mathbb{R}^2$ and $\vec{v} = (v_1, v_2) \in \mathbb{R}^2$. There is a one-to-one correspondence between vectors in $\mathbb{R}^2$ and complex numbers in $\mathbb{C}$; more precisely, to each free vector $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$ corresponds the complex number

$\mathbf{x} = x_1 + ix_2 \in \mathbb{C}$, and vice versa (i is the imaginary unit such that $i^2 = -1$). The shape of an ordered triangle, traced by a basic gesture $(\vec{u}, \vec{v})$, can be encoded by the complex number [15] obtained as the quotient $\frac{\mathbf{u}}{\mathbf{v}} \in \mathbb{C}$ of the complex numbers $\mathbf{u}, \mathbf{v} \in \mathbb{C}$ corresponding to vectors $\vec{u}, \vec{v} \in \mathbb{R}^2$ of the basic gesture $(\vec{u}, \vec{v})$. Every triangle can be characterized by its *shape* [15].

C. Vector-based Features

Gesture features are traditionally defined based on Cartesian coordinates. A transformation can convert them into any other system of coordinates, like polar, or spherical. Conversely, when a gesture is captured by a non-Cartesian device, a transformation into Cartesian coordinates should be applied before computing classical features, such as Rubine's features [17]. Instead, our vector-based definition re-expresses these features in terms of vectors that are independent of any system coordinates. We vectorized Rubine's features [17] into a red formula as follows:

Notations: $\forall (x_p, y_p) = \vec{u}_p$ with $0 \leq p \leq P-1$

$$\begin{aligned} \Delta x_p &= x_{p+1} - x_p, \Delta y_p = y_{p+1} - y_p \\ (\Delta x_p, \Delta y_p) &= \Delta \vec{u}_p = \vec{u}_{p+1} - \vec{u}_p \text{ and } \Delta t_p = t_{p+1} - t_p \\ \theta_p &= \arctan \frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} + \Delta y_p \Delta y_{p-1}} \\ &= \arctan \frac{(\Delta \vec{u}_p \wedge \Delta \vec{u}_{p-1}) \cdot (\vec{e}_1 \wedge \vec{e}_2)}{\Delta \vec{u}_p \cdot \Delta \vec{u}_{p-1}} \end{aligned} \quad (1)$$

Let $\{\vec{e}_1, \vec{e}_2\}$ be an orthonormal basis in the two-dimensional Euclidean space $\mathbb{E}_2$. Then, Rubine's features become:

$$\begin{aligned} f_1 &= \cos \alpha = \frac{x_2 - x_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} = \frac{(\vec{u}_2 - \vec{u}_0) \cdot \vec{e}_1}{|\vec{u}_2 - \vec{u}_0|} \\ f_2 &= \sin \alpha = \frac{y_2 - y_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} = \frac{(\vec{u}_2 - \vec{u}_0) \cdot \vec{e}_2}{|\vec{u}_2 - \vec{u}_0|} \\ f_5 &= \sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2} = |\vec{u}_{P-1} - \vec{u}_0| \\ f_6 &= \cos \beta = \frac{x_{P-1} - x_0}{\sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2}} = \frac{(\vec{u}_{P-1} - \vec{u}_0) \cdot \vec{e}_1}{|\vec{u}_{P-1} - \vec{u}_0|} \\ f_7 &= \sin \beta = \frac{y_{P-1} - y_0}{\sqrt{(x_{P-1} - x_0)^2 + (y_{P-1} - y_0)^2}} = \frac{(\vec{u}_{P-1} - \vec{u}_0) \cdot \vec{e}_2}{|\vec{u}_{P-1} - \vec{u}_0|} \\ f_8 &= \sum_{p=0}^{P-2} \sqrt{\Delta x_p^2 + \Delta y_p^2} = \sum_{p=0}^{P-2} |\vec{u}_{p+1} - \vec{u}_p| = \sum_{p=0}^{P-2} |\Delta \vec{u}_p| \\ f_9 &= \sum_{p=1}^{P-2} \theta_p = \arctan \frac{(\Delta \vec{u}_{p-1} \wedge \Delta \vec{u}_1) \cdot (\vec{e}_1 \wedge \vec{e}_2)}{\Delta \vec{u}_{p-1} \cdot \Delta \vec{u}_1} = \gamma \\ f_{10} &= \sum_{p=1}^{P-2} |\theta_p| \text{ and } f_{11} = \sum_{p=1}^{P-2} |\theta_p|^2 \text{ and } f_{13} = t_{P-1} - t_0 \\ f_{12} &= \max_{p=0}^{P-2} \frac{\Delta x_p^2 + \Delta y_p^2}{\Delta t_p^2} = \max_{p=0}^{P-2} \frac{|\Delta \vec{u}_p|^2}{\Delta t_p^2} = \max_{p=0}^{P-2} \left| \frac{1}{\Delta t_p} \Delta \vec{u}_p \right|^2 \end{aligned} \quad (2)$$

D. Similarity Ratio

Quotient $\frac{\mathbf{u}}{\mathbf{v}} \in \mathbb{C}$ is defined as the *similarity ratio* of the basic gesture $(\vec{u}, \vec{v})$. Let us recall the properties and correspondences between the metrics in $\mathbb{R}^2$ and $\mathbb{C}$: the norm of a free vector $\vec{x} \in \mathbb{R}^2$ and the modulus of its corresponding complex number $\mathbf{x} \in \mathbb{C}$ coincide

$$|\vec{x}| = \sqrt{\vec{x} \cdot \vec{x}} = \sqrt{(x_1)^2 + (x_2)^2} = \sqrt{\mathbf{x} \mathbf{x}^*} = |\mathbf{x}|_{\mathbb{C}} \text{ where } \quad (3)$$

- $\vec{x} \cdot \vec{y} = (x_1 y_1) + (x_2 y_2)$ is the scalar product between $\vec{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\vec{y} = (y_1, y_2) \in \mathbb{R}^2$,
- $\mathbf{x}^* = x_1 - ix_2 \in \mathbb{C}$ is the complex conjugate of $\mathbf{x} = x_1 + ix_2 \in \mathbb{C}$.

Thus, also the distance between two vectors $\vec{x}, \vec{y} \in \mathbb{R}^2$ and the distance between the two corresponding complex numbers $\mathbf{x}, \mathbf{y} \in \mathbb{C}$ coincide: $|\vec{x} - \vec{y}| = |x - y|_{\mathbb{C}}$, that is, $\mathbb{R}^2$ and $\mathbb{C}$ are isometric.

E. Local Shape Distance between Gestures

We define the dissimilarity between two basic gestures $(\vec{a}, \vec{b})$ and $(\vec{u}, \vec{v})$, by the *Local Shape distance* [22], denoted by the symbol $LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$, is defined as the Euclidean distance between the similarity ratios of the basic gestures $(\vec{a}, \vec{b})$ and $(\vec{u}, \vec{v})$, i.e.,

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = \sqrt{\frac{|\vec{a}|^2 |\vec{v}|^2 + |\vec{b}|^2 |\vec{u}|^2 - 2[(\vec{a} \cdot \vec{b})(\vec{u} \cdot \vec{v}) - (\vec{a} \cdot \vec{v})(\vec{b} \cdot \vec{u}) + (\vec{a} \cdot \vec{u})(\vec{b} \cdot \vec{v})]}{|\vec{b}| |\vec{v}|}} \quad (4)$$

The Local Shape Distance also exhibits a geometric interpretation: a basic gesture correspond to a triangle whose two first sides are formed by two vectors and whose last side correspond to a vector joining its origin to its destination. The Local Shape distance therefore interprets the comparison of a candidature gesture to a training gesture by computing the similarity between the two corresponding triangles. This process is repeated until the end of points is reached.

$$\begin{aligned} \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}}^2 &= \left| \frac{\mathbf{av} - \mathbf{bu}}{\mathbf{bv}} \right|_{\mathbb{C}}^2 = \frac{|\mathbf{av} - \mathbf{bu}|_{\mathbb{C}}^2}{|\mathbf{bv}|_{\mathbb{C}}^2} = \frac{(\mathbf{av} - \mathbf{bu})(\mathbf{av} - \mathbf{bu})^*}{|\mathbf{bv}|_{\mathbb{C}}^2} \\ &= \frac{(\mathbf{av} - \mathbf{bu})(\mathbf{a}^* \mathbf{v}^* - \mathbf{b}^* \mathbf{u}^*)}{|\mathbf{b}|_{\mathbb{C}}^2 |\mathbf{v}|_{\mathbb{C}}^2} = \frac{\mathbf{aa}^* \mathbf{vv}^* + \mathbf{bb}^* \mathbf{uu}^* - \mathbf{ab}^* \mathbf{vu}^* - \mathbf{a}^* \mathbf{bv}^*}{|\mathbf{b}|_{\mathbb{C}}^2 |\mathbf{v}|_{\mathbb{C}}^2} \end{aligned}$$

Let us recall that if $\vec{x} = (x_1, x_2)$ and $\vec{y} = (y_1, y_2)$ are two free vectors in $\mathbb{R}^2$ and their corresponding complex numbers in $\mathbb{C}$ are $\mathbf{x} = x_1 + ix_2$ and $\mathbf{y} = y_1 + iy_2$ respectively, then

$$\begin{aligned} \vec{x} \cdot \vec{y} &= x_1 y_1 + x_2 y_2 = (x_1 + ix_2)(y_1 - iy_2)(x_1 - ix_2)(y_1 + iy_2) \\ &= \mathbf{xy}^* = \mathbf{x}^* \mathbf{y}. \end{aligned}$$

$$\begin{aligned} \text{Hence, we write } \mathbf{aa}^* \mathbf{vv}^* + \mathbf{bb}^* \mathbf{uu}^* &= |\vec{a}|^2 |\vec{v}|^2 + |\vec{b}|^2 |\vec{u}|^2 \text{ and} \\ \mathbf{ab}^* \mathbf{vu}^* + \mathbf{a}^* \mathbf{bv}^* &= 2\mathbf{ab}^* \mathbf{vu}^* - \mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{bv}^* - \mathbf{a}^* \mathbf{bv}^* \\ &= 2\mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{ubv}^* - \mathbf{ab}^* \mathbf{u}^* \mathbf{v} - \mathbf{ab}^* \mathbf{vu}^* \\ &= 2\mathbf{ab}^* \mathbf{vu}^* + 2\mathbf{a}^* \mathbf{ubv}^* - 2\mathbf{ab}^* \mathbf{u}^* \mathbf{v} \\ &= 2(\vec{a} \cdot \vec{b})(\vec{v} \cdot \vec{u}) + 2(\vec{a} \cdot \vec{u})(\vec{b} \cdot \vec{v}) - 2(\vec{a} \cdot \vec{b})(\vec{u} \cdot \vec{v}) \end{aligned}$$

Equation (5) shows that LSD can be defined without considering complex numbers. With respect to the Euclidean Clifford [8] vector algebra $\mathcal{C}\ell_{(2,0)}$, associated to the Euclidean vector space $\mathbb{R}^2$, one can recognize that

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \vec{a}(\vec{b})^{-1} - \vec{u}(\vec{v})^{-1} \right|_{\mathcal{C}\ell_{(2,0)}}$$

the Euclidean distance in $\mathcal{C}\ell_{(2,0)}$ between two ratios of vectors, where

$$\vec{x}(\vec{y})^{-1} = \frac{\vec{x} \cdot \vec{y}}{|\vec{y}|^2} + \frac{1}{|\vec{y}|^2} (\vec{x} \wedge \vec{y}) \in \mathcal{C}\ell_{(2,0)}$$

In general, the two free vectors $\vec{u}$ and $\vec{v}$ of a basic gesture $(\vec{u}, \vec{v})$, can belong to a finite n -dimensional Euclidean space $\mathbb{R}^n$. Moreover, the LSD defined in (5) makes sense and corresponds to the distance between the ratio multivectors in the corresponding Euclidean Clifford algebra $\mathcal{C}\ell_{(n,0)}$, i.e.,

$$LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v})) = \left| \vec{a}(\vec{b})^{-1} - \vec{u}(\vec{v})^{-1} \right|_{\mathcal{C}\ell_{(n,0)}}$$

F. Invariance Properties of the Local Shape Distance

In order to demonstrate the invariance properties of LSD, let us suppose to use function LSD to compare two basic gestures $(\vec{a}, \vec{b})$ and $(\vec{u}, \vec{v})$ laying on a same affine plane.

Stroke-number invariance. LSD is computed on a set of vectors that are either continuous (a basic gesture) or a series of continuous ones (a series of basic gestures). If no stroke exists between the ending point of a gesture stroke and the starting point of the next stroke, no vector is created and the LSD remains unaffected. Therefore, both single-stroke and multi-stroke gestures are supported. The sampling limit can be pushed to only 8 points [25], but we observe that the 32 points-sampling represents a viable compromise between recognition rate and execution time.

Sampling invariance. LSD is computed on vectors created from one or many series of points, provided that isoparametrization is ensured. How the points are considered and thus how vectors are defined for computing the LSD does not affect LSD. Thus, stroke-order invariance is supported.

Direction invariance. LSD is not symmetric: it can happen that $LSD(\vec{a}, \vec{b}, \vec{c}, \vec{d}) \neq LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})$. An oriented gesture can be transformed into its unoriented version and redirected into a reference vector in one direction. We want to avoid edgification. Instead, a simpler and more efficient approach is preferred: since only the denominator differs for both $LSD(\vec{a}, \vec{b}, \vec{c}, \vec{b})$ and $LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})$, we compute the symetrized version which is independent of the direction:

$$LSD_{sym}(\vec{a}, \vec{b}, \vec{c}, \vec{d}) = \frac{LSD(\vec{a}, \vec{b}, \vec{c}, \vec{d}) + LSD(\vec{b}, \vec{a}, \vec{d}, \vec{c})}{2}$$

One can also observe that both LSD and LSD_{sym} depend on the lengths of vectors $\vec{a}$, $\vec{b}$, $\vec{c}$, and $\vec{d}$. Such sensitivity to the length of each basic gesture is valuable when the points of the corresponding ordered triangle faithfully represents the isochrone sampling of a real gesture. The sampling offered by a real-world device is rarely isochrone. Equally spaced successive sampled points, i.e., isometric points, do not correspond to equally time-spaced points of the gesture, i.e., to isochrone points. In order to reduce this potential bias, we define a LSD independent on the lengths of the vectors in the basic gestures: $\forall(\vec{a}, \vec{b}), (\vec{c}, \vec{d})$, the Normalized Local Shape Distance (NLSD) is defined as follows:

$$NLSD((\vec{a}, \vec{b}), (\vec{c}, \vec{d})) = LSD\left(\left(\frac{\vec{a}}{|\vec{a}|}, \frac{\vec{b}}{|\vec{b}|}\right), \left(\frac{\vec{c}}{|\vec{c}|}, \frac{\vec{d}}{|\vec{d}|}\right)\right)$$

$NLSD$ is symmetric: $NLSD((\vec{a}, \vec{b}), (\vec{c}, \vec{d})) = NLSD((\vec{b}, \vec{a}), (\vec{d}, \vec{c}))$.

Position Invariance. This is ensured by proving that a translation preserves LSD. The translation performed by a free vector $\vec{t}$ of a basic gesture $(\vec{a}, \vec{b})$ (corresponding to a oriented triangle of vertices A , B and C) produces a basic gesture $(\vec{a}', \vec{b}')$ (corresponding to a oriented triangle of vertices $A'=A+\vec{t}$, $B'=B+\vec{t}$ and $C'=C+\vec{t}$). Any translation $\vec{t}$ does not affect the basic gesture, thanks to $\vec{a}'=\vec{a}$ and $\vec{b}'=\vec{b}$.

Scale Invariance. This is ensured by proving that a homogeneous dilation preserves LSD. $\forall \lambda \neq 0 \in \mathbb{R}$, we have that

$$LSD((\lambda \vec{a}, \lambda \vec{b}), (\vec{u}, \vec{v})) = \left| \frac{\lambda \vec{a}}{\lambda \vec{b}} - \frac{\vec{u}}{\vec{v}} \right|_{\mathbb{C}} = \left| \frac{\vec{a}}{\vec{b}} - \frac{\vec{u}}{\vec{v}} \right|_{\mathbb{C}} = LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$$

Rotation Invariance. This is ensured by proving that a rotation preserves LSD. One can note that

- 1) Every counter-clockwise rotation $\mathcal{R}_\alpha$ in the Euclidean plane $\mathbb{R}^2$ of a Radian angle $\alpha \in \mathbb{R}$ corresponds to a counter-clockwise rotation R_α in the isometric complex plane $\mathbb{C}$ (of equal angle).
- 2) Recalling the Euler's formulas $e^{i\alpha} = \cos \alpha + i \sin \alpha$, $e^{i(\alpha+\beta)} = e^{i\alpha} e^{i\beta}$, the complex number $R_\alpha(\mathbf{x})$ can be obtained throughout the complex multiplication $R_\alpha(\mathbf{x}) = e^{i\alpha} \mathbf{x}$.
- 3) The ratio of two complex numbers, both rotated by a same angle, is equal to their original ratio,

$$\frac{R_\alpha(\mathbf{x})}{R_\alpha(\mathbf{y})} = \frac{e^{i\alpha} \mathbf{x}}{e^{i\alpha} \mathbf{y}} = e^{i\alpha} \mathbf{x} (e^{i\alpha} \mathbf{y})^{-1} = e^{i\alpha} \mathbf{x} e^{-i\alpha} \mathbf{y}^{-1} = \underbrace{e^{i(\alpha-\alpha)}}_{=1} \mathbf{x} \mathbf{y}^{-1} = \frac{\mathbf{x}}{\mathbf{y}}$$

Hence, we have that $LSD((\mathcal{R}_\alpha(\vec{a}), \mathcal{R}_\alpha(\vec{b})), (\vec{u}, \vec{v}))$

$$= \left| \frac{R_\alpha(\mathbf{a})}{R_\alpha(\mathbf{b})} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = \left| \frac{\mathbf{a}}{\mathbf{b}} - \frac{\mathbf{u}}{\mathbf{v}} \right|_{\mathbb{C}} = LSD((\vec{a}, \vec{b}), (\vec{u}, \vec{v}))$$

One can verify that the general LSD defined in (5) has the same invariance properties in any finite dimensional non-degenerate quadratic space. In order to examine how LSD can also be used for 3D gestures and for the symetrized version, we introduce the following definitions.

G. Discrete Curves

A continuous *curve* is a continuous function $\vec{x}: I \rightarrow \mathbb{R}^n$ from a interval I to the d -dimensional Euclidean space $\mathbb{R}^d$. A *discrete curve* is a string (i.e. a sequence) $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ of L ordered points, Where L is called the *length* of the string $\vec{X}$, in the d -dimensional Euclidean space $\mathbb{R}^d$ laying on some curve. A *plane discrete curve* is a discrete curve whose points lay in the Euclidean plane $\mathbb{R}^2$. Using Cartesian coordinates, a plane discrete curve $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ will appear as a sequence of couples of real numbers:

$$\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\} = \{(x_{1,0}, x_{2,0}), \dots, (x_{1,L-1}, x_{2,L-1})\}$$

Two discrete curves $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$, $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{M-1}\}$ are *isoparametrized* iff their strings have the same length, i.e. $L=M$. A plane discrete curve $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ is *regular* if $\vec{x}_{i+1} - \vec{x}_i \neq \vec{0} = (0, 0), \forall i=0, \dots, L-2$.

The oriented shape distance between two discrete curves. Given two regular isoparametrized plane discrete curves $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$, $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{L-1}\}$, the function $d^+(\vec{X}, \vec{Y}) =$

$$\sum_{0 \leq i \leq L-3} LSD((\vec{x}_{i+1}-\vec{x}_i, \vec{x}_{i+2}-\vec{x}_{i+1}), (\vec{y}_{i+1}-\vec{y}_i, \vec{y}_{i+2}-\vec{y}_{i+1}))$$

is called the (positively oriented) *Shape Distance* between $\vec{X}$ and $\vec{Y}$. If $\vec{X}$ and $\vec{Y}$ are both basic gestures laying on a same plane, then $d(\vec{X}, \vec{Y}) = LSD((\vec{x}_1-\vec{x}_0, \vec{x}_2-\vec{x}_1), (\vec{y}_1-\vec{y}_0, \vec{y}_2-\vec{y}_1))$.

The unoriented shape distance between two discrete curves. Since LSD is sensible to the orientation of the basic gestures, d^+ is sensible to the orientation of the discrete curves. We therefore propose a shape distance that does not depend on the orientation on the two regular isoparametrized plane discrete curves. Given two regular isoparametrized plane discrete curves $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$, $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{L-1}\}$, the function $d^-(\vec{X}, \vec{Y}) = \sum_{0 \leq i \leq L-3} LSD((\vec{x}_{i+1}-\vec{x}_i, \vec{x}_{i+2}-\vec{x}_{i+1}), (\vec{y}_{L-(i+2)}-\vec{y}_{L-(i+1)}, \vec{y}_{L-(i+3)}-\vec{y}_{L-(i+2)}))$

is called the (negatively oriented) *Shape Distance* between $\vec{X}$ and $\vec{Y}$, and $d(\vec{X}, \vec{Y}) = \min \{d^+(\vec{X}, \vec{Y}), d^-(\vec{X}, \vec{Y})\}$ is called the (unoriented) *Shape Distance* between $\vec{X}$ and $\vec{Y}$.

The shape distance between two discrete curves (not necessarily isoparametrized). Two plane gestures, corresponding to two regular plane discrete curves $\vec{X}$ and $\vec{Y}$, $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$, $\vec{Y} = \{\vec{y}_0, \dots, \vec{y}_{M-1}\}$ are often not isoparametrized ($L \neq M$), especially when coming from different gesture sets. Here we propose a “linear” way to compare them through the above shape distance. If $L \neq M$, we can consider the points of each discrete curve $\vec{X}$ and $\vec{Y}$ as the vertices of polygonal (continuous) curves $\vec{x}$ and $\vec{y}$, respectively. We can fix whatever $P \in \mathbb{N}$ (with $P \geq 3$), interpolate P points on each polygonal continuous curve $\vec{x}$ and $\vec{y}$, and thus produce two regular isoparametrized plane discrete curve of the same length P : $\vec{X}^{(P)} = \{\vec{x}_0^{(P)}, \dots, \vec{x}_{P-1}^{(P)}\}$, $\vec{Y}^{(P)} = \{\vec{y}_0^{(P)}, \dots, \vec{y}_{P-1}^{(P)}\}$. Then we can calculate the distance depending on the new common length P : $d_P(\vec{X}, \vec{Y}) = d(\vec{X}^{(P)}, \vec{Y}^{(P)})$. There are different criteria to interpolate points on a continuous curve; here, we consider the linear criterion.

Definition of string $\vec{X}^{(P)}$ from the given string $\vec{X}$. Let $\vec{X}$ be a regular plane discrete curve $\vec{X} = \{\vec{x}_0, \dots, \vec{x}_{L-1}\}$ with $L \geq 2$. Let $P \in \mathbb{N}$ (with $P \geq 3$), we define $\forall j=0, \dots, P-1$,

$$\vec{x}_j^{(P)} = \frac{P-(j+1)}{P-1} \vec{x}_{\lfloor \frac{N-1}{P-1} j \rfloor} + \frac{j}{P-1} \vec{x}_{\lceil \frac{N-1}{P-1} j \rceil}$$

where $\lfloor \cdot \rfloor$ is the floor function, and $\lceil \cdot \rceil$ is the ceiling function.

IV. CONCLUSION AND FUTURE WORK

This paper explained how a Local Shape Distance (LSD and NLSD) could compute the dissimilarity between gestures represented as n -dimensional vectors with several properties: point-number and stroke-number invariance, stroke-order and stroke-direction invariance, position, scale, and rotation invariance. An instantiation of LSD (and NLSD) to $n=2$ gave rise to a 2D stroke gesture recognizer. This work will benefit practitioners by providing a new gesture recognizer satisfying

That is, have both length $L = 3$, the basic plane gesture corresponding to the plane discrete curve $\vec{X}$ being $(\vec{x}_1-\vec{x}_0, \vec{x}_2-\vec{x}_1)$, and the basic plane gesture associated to plane discrete curve $\vec{Y}$ being $(\vec{y}_1-\vec{y}_0, \vec{y}_2-\vec{y}_1)$.

Other ways could use non-linear splines to continuously connect the ordered points of a discrete curve.

the requirements in a comparable way while preserving the invariance properties. If for any reason a gesture should become for instance scale variant, the constraint can be imposed on the vectors without any loss of generality and without complexifying the recognizer with additional computations.

REFERENCES

- [1] L. Anthony and J. O. Wobbrock. A lightweight multistroke recognizer for user interface prototypes. In *Proc. of GI '10*, pages 245–252, 2010.
- [2] C. Appert and O. Bau. Scale detection for a priori gesture recognition. In *Proc. of CHI '10*, pages 879–882. ACM, 2010.
- [3] F. Beuvs and J. Vanderdonckt. Designing graphical user interfaces integrating gestures. In *Proc. of SIGDOC '12*, 2012.
- [4] A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola, Jr. Gesturebar: Improving the approachability of gesture-based interfaces. In *Proc. of CHI '09*, 2009.
- [5] S. h.-h. Chang, R. Blagojevic, and B. Plimmer. Rata.gesture: A gesture recognizer developed using data mining. *Artif. Intell. Eng. Des. Anal. Manuf.*, 26(3):351–366, Aug. 2012.
- [6] M. Cirelli and R. Nakamura. A survey on multi-touch gesture recognition and multi-touch frameworks. In *Proc. of ITS '14*.
- [7] A. Coyette, S. Schimke, J. Vanderdonckt, and C. Vielhauer. *Trainable Sketch Recognizer for Graphical User Interface Design*.
- [8] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2007.
- [9] V. Fuccella, M. De Rosa, and G. Costagliola. Novice and expert performance of keystretch: A gesture-based text entry method for touch-screens. *IEEE Trans. on HMI*, 44(4):511–523, Aug 2014.
- [10] J. Herold and T. F. Stahovich. The 1¢ recognizer: A fast, accurate, and easy-to-implement handwritten gesture recognition technique. In *Proc. of SBIM '12*, pages 39–46, 2012.
- [11] H. Hse, M. Shilman, and A. R. Newton. Robust sketched symbol fragmentation using templates. In *Proc. of IUI '04*.
- [12] K. Kim and H.-I. Choi. Online hand gesture recognition using enhanced $n recogniser based on a depth camera. *Int. J. Comput. Vision Robot.*, 6(3):214–222, Jan. 2016.
- [13] S. Kratz and M. Rohs. Protractor3d: A closed-form solution to rotation-invariant 3d gestures. In *Proc. of IUI '11*.
- [14] L. A. Leiva, D. Martín-Albo, and R. Plamondon. Gestures À go go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM TIS*.
- [15] J. A. Lester. Triangles i: Shapes. *aequationes mathematicae*, 52(1):30–54, Feb 1996.
- [16] C. Pittman, E. M. Taranta II, and J. J. LaViola, Jr. A \$-family friendly approach to prototype selection. In *Proc. of IUI '16*.
- [17] D. Rubine. Specifying gestures by example. In *Proc. of SIGGRAPH '91*, pages 329–337, New York, NY, USA, 1991. ACM.
- [18] B. Signer, U. Kurmann, and M. Norrie. igesture: A general gesture recognition framework. In *Proc. of ICDAR '07*, pages 954–958, 2007.
- [19] E. Taranta and J. LaViola. Penny pincher: a blazing fast, highly accurate \$-family recognizer. In *Proc. of GI 2015*.
- [20] E. M. Taranta, A. N. Vargas, and J. J. LaViola. Streamlined and accurate gesture recognition with penny pincher. *Computers Graphics*, 55:130 – 142, 2016.
- [21] H. Tu, X. Ren, and S. Zhai. Differences and similarities between finger and pen stroke gestures on stationary and mobile devices. *ACM Trans. Comput.-Hum. Interact.*, 22(5):1–39, Aug. 2015.
- [22] J. Vanderdonckt, P. Roselli, and J. L. Pérez-Medina. !FTL, an articulation-invariant stroke gesture recognizer with controllable position, scale, and rotation invariances. In *Proc. of ICMI '18*.
- [23] R.-D. Vatavu. The effect of sampling rate on the performance of template-based gesture recognizers. In *Proc. of ICMI '11*.
- [24] R.-D. Vatavu, L. Anthony, and J. Wobbrock. \$Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. In *Proc. of MobileHCI '18*, pages 623–635. ACM, 2018.
- [25] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock. Gestures as point clouds: A \$P recognizer for user interface prototypes. In *Proc. of ICMI '12*, pages 273–280, New York, NY, USA, 2012. ACM.
- [26] R.-D. Vatavu, G. Casiez, and L. Grisoni. Small, medium, or large?: Estimating the user-perceived scale of stroke gestures. In *Proc. of CHI '13*, pages 277–280, New York, NY, USA, 2013. ACM.
- [27] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. In *Proc. of UIST '07*, pages 159–168, New York, 2007. ACM.
- [28] S. Zhai, P. O. Kristensson, C. Appert, T. H. Anderson, and X. Cao. Foundational issues in touch-surface stroke gesture design: an integrative review. *Foundations and Trends in HCI*, 5(2):97–205, 2012.

A Systematic Mapping on Visual Solutions to Support the Comprehension of Software Architecture Evolution

Joao Werther¹, Glauco de Figueiredo Carneiro², Rita Suzana Pitangueira Maciel¹

¹Federal University of Bahia, BRA

²Universidade Salvador (UNIFACS), BRA

jwertherf@gmail.com, glauco.carneiro@unifacs.br, ritasuzana@dcc.ufba.br

Abstract

Context: Software visualization has the potential to support specialized stakeholders to understand the software architecture (SA) evolution. To the best of our knowledge, there is no guideline based on which visual solutions should be applied to support the SA evolution comprehension and how to use them. Goal: Analyze the use of visual solutions for the purpose of comprehension with respect to software architecture evolution from the point of view of software architects and developers in the context of both academia and industry. Method: We conducted a Systematic Mapping Study to achieve the stated goal. Results: The study identified 92 papers published from January 2000 to December 2018 as a result of the search strings execution. We selected 12 primary studies and identified a gap in terms of a taxonomy to assist specialists in the development or classification of solutions to support the comprehension of software architecture evolution using visual resources. Conclusion: We observed that despite the relevance of the use of visual solutions to support the comprehension of software architecture evolution, only 12 studies have reported these initiatives, suggesting that there is still room for the use of different visual metaphors to represent its components, relationships and evolution throughout the releases.

1. Introduction

Software evolution reflects changes undergone by the software during its lifespan [1]. The study of software evolution is essential to better support changes in software requirements over time, keeping its integrity at a lowest possible cost [2][3][4]. Software Architecture (SA) is the design model used to build and evolve a software system [5]. Throughout the analysis of the software architecture, in high level structure, it is possible to understand what dimensions along which a system is expected to evolve [1]. The importance of SA in software evolution process is that a software built without an adaptable architecture normally will degenerate sooner than others with a change-ready architecture [1]. The evolution of a SA can be the result of changes in the current SA to accommodate business demands, new technologies and/or platform or other reason that impacts

the SA [6]. The comprehension of SA is essential for development and evolution of software systems [7][8] and can be supported by software visualization (SV) resources to understand key SA characteristics regarding architectural models and design decisions [7] [9].

To the best of our knowledge, there is no guideline to support the use of visual solutions towards SA evolution comprehension. For this reason, we conducted a Systematic Mapping Study (SMS) to identify studies that focus on this issue based on evidence provided by papers published in peer-reviewed conferences and journals from January 2000 to December 2018. We initially found 92 papers as a result of the applied search strings in specific electronic databases, from which we considered 12 studies as relevant.

This paper is organized as follows. Section 2 discusses related works and Section 3 presents the protocol of this SM. The Section 4 presents the main results of this study, also introducing the taxonomy proposed for visual solutions to software architecture evolution and presenting the main characteristics and features for each visual solution. The Section 5 discusses about the answers to research questions. Finally, Section 6 gives the conclusions of this work and suggests future works.

2. Related Works

Software visualization has been used in different areas of software engineering such as software architecture, software evolution and software design [7][10]. In the following paragraphs we highlight the importance of specific secondary studies that discussed the use of software visualization to support the execution of activities targeting software architecture. This is not an exhaustive list, it is rather an illustrative set of relevant papers whose discussion motivated the conduction of this systematic mapping.

Shahin, Liang and Babar [7] conducted a systematic review to characterize how Visualization Techniques (VT) have been used to represent SA in different application domains. Results classified the VTs into four types, according to its popularity: graph-based, notation-based, matrix-based and metaphor-based, where graph-based and notation-based VTs appear to be the most popular ones. However, the graph-based was pointed as the most popular architecture visualization technique in industry, largely employed. They

concluded that VTs have been used to support SA activities for several purposes and highlighted the following as relevant: (i) the understanding of architecture evolution; (ii) the understanding of static characteristics of architecture; and (iii) search, navigation, and exploration of architecture design [7]. Additionally, conclusion highlighted that VTs have been applied to SA in a large range of domains, highlighting “graphics software” and “distributed system” as those that have been received the most attention in industry. Finally, in the conclusion section, the authors argued that SV is one of the increasingly popular ways to support the understanding the rationale behind design decisions that affect software architecture [7]. This paper focused on VT to represent SA, not concerning about the evolution of SA and its understanding.

Telea, Voinea and Sassenburg (2010) [10] performed a survey regarding the use of visual tools for SA understanding from the perspective of stakeholders. They analyzed the results using software architecture visualizations tools (AVTs) aiming to orientate industrial practitioners in the adoption of tools and techniques according requirements and capabilities of each type. For this survey, they defined three stakeholder types: a technical users (developers), project managers/lead architects, and consultants. They concluded that the AVTs were more adequate for technical users and less adequate for consultants, according to expectancy of each stakeholder [10]. Although this paper explored the visual solutions for SA understanding, it did not focus on SA evolution solutions.

Breivold, Crnkovic and Larson (2012) [1] conducted a systematic review focusing on software architecture evolution. The goal of the review was to get an overview at the architectural level about existing approaches in the analysis and improvement of software evolution, and also examine the impacts on research and industry. The authors identified five main categories of themes: (i) techniques supporting quality consideration during software architecture design, (ii) architectural quality evaluation, (iii) economic valuation, (iv) architectural knowledge management, and (v) modeling techniques [1]. The conclusion of this study emphasized the need of development of methods, process and/or tools to design architecture in a very large systems involving many platforms, organizations and processes, considering the amount and particularity of artifacts produced and used during its lifecycle. This study also presented several useful conclusions for researchers and practitioners, including considering the possibility to elaborate new ideas beyond Lehman’s laws (about software evolution) so long as it has practical value for evolution. Additionally, this paper also reported the existence of few works for this area and expected that happens more research works in the future. [1]. Despite this article has studied the evolution of SA, it did not emphasize on visual solutions to support the

comprehension of SA evolution.

Despite the large contribution of the aforementioned related works to the SA area, including SA evolution, they do not focus on visual solutions to support the comprehension of the SA evolution.

3. Research Design

We conducted a SMS to find evidence for the use of visual solutions to support the comprehension of SA evolution during the software lifespan. A SMS is a form of a systematic literature review (SLR) with more general research questions, aiming to provide an overview of the given research [11]. We decided to conduct a SMS due to the potential that this methodology has to reduce the analysis bias, through the establishment of selection procedures [12].

3.1. Planning

We conducted this SMS based on a protocol comprised of objectives of the review, criteria for considering papers, research questions, selected electronic databases and its search strings, selection procedures and exclusion, inclusion and quality criteria to select the studies from which we aim to answer the stated research questions [12]. The protocol of this SMS and related artifacts are available in a public Github repository ¹. The goal of this study is presented in Table 1 according to the GQM approach [13].

Table 1: The Goal of this SMS according to the GQM Approach

Analyze	the use of visual solutions
for the purpose of	comprehension
with respect to	software architecture evolution
from the point of view of	software architects and developers
in the context of	both academia and industry

The Research Question (RQ) is “*How have researchers and practitioners from the industry considered the use of software visual solutions to support the comprehension of software architecture evolution based on papers published in the peer-reviewed literature?*”. This research question is in line with the goal of this review, and has been derived into four specific research questions, as follows: Specific Research Question 1 (SRQ1): *What are the main visual solutions to support the software architecture evolution comprehension?* Specific Research Question 2 (SRQ2): *What are the purposes of each visual solution to support the software architecture evolution comprehension?* Specific Research Question 3 (SRQ3): *How the solutions designed to visually support comprehension of software architecture evolution can be classified?* Specific Research Question 4 (SRQ4):

¹<https://github.com/dmsviva19saevolution/submission>

Which visual forms are used to support comprehension of software architecture evolution?

The motivation behind RQ is justified by the acknowledgment that the comprehension of the software architecture evolution is required to tackle issues or improvements related to the software architecture and its evolution throughout releases [14] [5] [8] [7] [6] [2]. The specific research questions have the goal to gather evidence to support the answer of the stated RQ.

We considered the PICO criteria to define the search strings, as shown in Table 2. The search strings are based on this criteria for the selective process of papers for this review.

Table 2: PICO Criteria for Search Strings

(P)opulation	studies of software architecture in software engineering
(I)ntervention	visual solutions to support the comprehension of software architecture evolution
(C)omparison	not applicable
(O)utcomes	solutions (i.e., tools, techniques, environment, approaches, models or methodology) with focus on visual resources to support software architecture evolution; visual solutions to software architecture evolution; use of visual resources to comprehension of software architecture evolution

The formation of the search string applied in the electronic databases is shown in Tables 3 and 4. The Table 3 refers to major terms for the research objectives, built using the PICO criteria. We also considered the use of alternative terms and synonyms of these major terms. For example, the term *visualization* can be associated with terms such as *visual*, *visualizing* and *visualize*. These alternative terms, as shown in Table 4, are also included in the search string. We built the final search string by joining the major terms with the boolean “AND” and joining the alternative terms to the main terms with the boolean “OR”. The focus of the formed search strings is to focus on papers targeting the research questions of this systematic mapping.

Table 3: Major terms for the research objectives

Criteria	Major Terms
(P)opulation	AND “software engineering” AND “software architecture”
(I)ntervention	AND “comprehension” AND “evolution”
(C)omparison	Not Applicable
(O)utcomes	AND “visual” AND “solution”

Table 5 presents the electronic databases from which we retrieved the papers along with the respective search strings used to retrieve the papers. Table 6 presents the criteria for exclusion and inclusion of papers in this review. The *OR* connective adopted in the exclusion criteria, means that the exclusion criteria are independent, i.e., meeting only one criterion is enough to exclude the paper. On the other hand, the *AND* connective in the inclusion criteria, means that all

Table 4: Alternative terms from majors terms

Major Term	Alternative Terms
“evolution”	(“evolution” OR “evolve” OR “evolving”)
“comprehension”	(“comprehension” OR “understanding” OR “understand” OR “support” OR “analysis” OR “evaluation” OR “examination” OR “explore” OR “exploring”)
“solution”	(“tool” OR “environment” OR “technique” OR “approach” OR “model” OR “methodology” OR “solution”)
“visual”	(“visualization” OR “visualizing” OR “visualize” OR “visual”)

Table 5: Electronic Databases Selected for this SMS

Database and URL	Search Strings
Scopus www.scopus.com	(“software architecture” AND (“evolution” OR “evolve” OR “evolving”) AND (“comprehension” OR “understanding” OR “understand” OR “support” OR “analysis” OR “evaluation” OR “examination” OR “explore” OR “exploring”) AND (“tool” OR “environment” OR “technique” OR “approach” OR “model” OR “methodology” OR “solution”) AND (“visualization” OR “visualizing” OR “visualize” OR “visual”) AND “software engineering”)
ACM Digital Library portal.acm.org	(+“software architecture” + (“evolution” “evolve” “evolving”) + (“comprehension” “understanding” “understand” “support” “analysis” “evaluation” “examination” “explore” “exploring”) + (“tool” “environment” “technique” “approach” “model” “methodology” “solution”) + (“visualization” “visualizing” “visualize” “visual”) + “software engineering”)
Engineering Village (Ei Compendex) www.engineeringvillage.com	(“software architecture” AND (“evolution” OR “evolve” OR “evolving”) AND (“comprehension” OR “understanding” OR “understand” OR “support” OR “analysis” OR “evaluation” OR “examination” OR “explore” OR “exploring”) AND (“tool” OR “environment” OR “technique” OR “approach” OR “model” OR “methodology” OR “solution”) AND (“visualization” OR “visualizing” OR “visualize” OR “visual”) AND “software engineering”)
IEEE Xplore ieeexplore.ieee.org	(“software architecture” AND (“evolution” OR “evolve” OR “evolving”) AND (“comprehension” OR “understanding” OR “understand” OR “support” OR “analysis” OR “evaluation” OR “examination” OR “explore” OR “exploring”) AND (“tool” OR “environment” OR “technique” OR “approach” OR “model” OR “methodology” OR “solution”) AND (“visualization” OR “visualizing” OR “visualize” OR “visual”) AND “software engineering”)

inclusion criteria must met to select the paper under analysis. Table 6 also presents the quality criteria used for this review represented as questions that were adopted and adjusted from Dyba and Dingsoyr [15]. A critical examination following the quality criteria established in this table was performed in all remaining papers that passed the exclusion and inclusion criteria. All these criteria must met (i.e., the answer must be YES for each one) to permanently select the paper, otherwise the paper must be excluded. The exclusion, inclusion and quality criteria were used in the selection process as presented in Table 7. According to Table 8, at the end of the selection process, all the retrieved papers were classified in one of the three options: *Excluded*, *Not Selected* and *Selected*.

3.2. Execution

The quantitative evolution of papers throughout the execution of this SMS is summarized in Figure 1. The figure uses the PRISMA flow diagram [16] and shows the performed steps and the respective number of documents for each phase of the SMS, following the outline described in Subsection 3.1.

Table 9 presents the effectiveness of the the search strings showed in Table 5 considering the 92 retrieved pa-

Table 6: Exclusion, Inclusion and Quality Criteria

Type	Id	Description	Connective or Answer
Exclusion	E1	Published earlier than 2000	OR
Exclusion	E2	The paper was not published in a peer-reviewed journal or conference	OR
Exclusion	E3	The paper does not present a primary study	OR
Exclusion	E4	The paper is not written in English	OR
Exclusion	E5	The paper has less than 3 pages	OR
Inclusion	I1	The paper must present an approach in the usage of visual solution to support the comprehension of software architecture evolution	AND
Quality	Q1	Are the aims of the study clearly specified?	YES/NO
Quality	Q2	Is the context of the study clearly stated?	YES/NO
Quality	Q3	Does the research design support the aims of the study?	YES/NO
Quality	Q4	Has the study an adequate description of the visual solution?	YES/NO
Quality	Q5	Is there a clear statement of findings by applying the visual solution to support the comprehension of software architecture evolution?	YES/NO

Table 7: Steps of the Selection Process

Step	Description
1	Apply the search strings to obtain a list of candidate papers in specific electronic databases.
2	Remove replicated papers from the list.
3	Apply the exclusion criteria in the listed papers.
4	Apply the inclusion criteria after reading abstracts, introduction and conclusion in papers not excluded in step 3.
5	Apply quality criteria in selected papers in step 3.

Table 8: Classification Options for Each Retrieved Paper

Classification	Description
Excluded	Papers met the exclusion criteria.
Not Selected	Papers not excluded due to the exclusion criteria, but did not meet the inclusion or quality criteria.
Selected	Papers did not meet the exclusion criteria and met both the inclusion and quality criteria.

pers. The electronic database that more contributed with selected studies was the *IEEE Xplore* with five papers, corresponding to a search effectiveness of 18.5%. The twelve selected papers represented 13.0% of all 92 retrieved papers.

The Figure 2 presents a overview of contribution of each exclusion criterion in total of excluded papers. The exclusion criterion that had more contribution was the E1 criterion that says “*Published date less than 2000*”, accounting for 54% of excluded papers. Note that the E4 criterion (“*The paper is not written in english*”) did not have any occurrence.

The Figure 3 presents graphics that provide a overview of sources (electronic databases) distribution by papers status. The papers status is according to Table 8. The Figure

Figure 1: Procedures and its results in the papers selection process.

Table 9: Effectiveness of the Search Strings

Database	Selected Papers	Excluded Papers	Not Selected Papers	Replicated Papers	Total Search Results	Search Effectiveness
ACM Digital Library	2	2	13	1	18	11.1%
Engineering Village	3	4	6	14	27	11.1%
IEEE Xplore	5	4	14	4	27	18.5%
Scopus	2	3	3	12	20	10.0%
TOTAL	12	13	36	31	92	13.0%

Figure 2: Contribution of exclusion criteria in total of excluded papers.

3a shows the selected papers distribution by source. The “IEEE Xplore” had the major contribution for selected papers with 41% of occurrences. The “Engineering Village”

was the second with 25% of selected studies. The Figure 3b shows the not-selected papers distribution by source. The “IEEE Xplore” led with 39% of not-selected papers, close to “ACM Digital Library” with 36%. The Figure 3c presents the excluded papers distribution by source. The “Engineering Village” and “IEEE Xplore” databases had the majors contributions with 31% of excluded papers each one. The Figure 3d presents the distribution of all found papers by source. The “Engineering Village” and “IEEE Xplore” databases had the majors contributions, both with 29% of all papers found in its databases searches.

Figure 3: Overview of paper totalization by status and by electronic databases.

4. Results

The Table 10 shows the list of 12 selected papers of this systematic mapping. All papers are labeled as “S” followed by the paper reference number through which the paper can be reached at the end of this document. All the selected papers were published in conferences. Unfortunately, the retrieved papers from journals matched the exclusion criteria (Step 3 of Table 7) or did not matched the inclusion criteria (Step 4 of Table 7). For this reason, they were discarded from this systematic mapping.

The paper **S01** [3] describes a solution aimed at enhancing the comprehension of the software architectural evolution based on visual resources. This solution proposes the use of efficient navigation and visualization of the history of software architectural changes throughout releases, integrating the use of evolution metrics with software visualization techniques. This integration has the goal to support both tracking and analysis of architectural changes from past releases. The authors developed the so called *Origin Analysis* method to analyze software structural change. This method supports the identification of possible origin of function or file that appears to be new in a later release of the software system, if it already existed in the system elsewhere [3]. This method highlights the use of two techniques in its implementation: *Bertillonage Analysis* and *Dependency Analysis*. The paper also performs a study of evo-

Table 10: Summary List of Selected Papers

Ref. Label	Title	Conference	Year
S01	An integrated approach for studying architectural evolution [3]	IEEE Workshop on Program Comprehension	2002
S02	An Approach based on Bi-graphical Reactive Systems to Check Architectural Instance Conforming to its Style [17]	First Joint IEEE/IFIP Symp. Theoretical Aspects of Software Engineering (TASE)	2007
S03	Exploring inter-module relationships in evolving software systems [4]	European Conference on Software Maintenance and Reengineering (CSMR)	2007
S04	The SAVE Tool and Process Applied to Ground Software Development at JHU/APL: An Experience Report on Technology Infusion [18]	IEEE Software Engineering Workshop (SEW)	2007
S05	Development of a methodology, software-suite and service for supporting software architecture reconstruction [19]	European Conference on Software Maintenance and Reengineering (CSMR)	2010
S06	Evolve: tool support for architecture evolution [20]	International Conference on Software Engineering (ICSE)	2011
S07	Model-Based Software Architecture Evolution and Evaluation [21]	Asia-Pacific Software Engineering Conference (APSEC)	2012
S08	ECITY: A tool to track software structural changes using an evolving city [22]	International Conference on Software Maintenance (ICSM)	2013
S09	Run-time monitoring and real-time visualization of software architectures [23]	Asia-Pacific Software Engineering Conference (APSEC)	2013
S10	eCITY4: A Tool to Analyze Software Architectural Relations Through Interactive Visual Support [24]	European Conference on Software Architecture Workshops	2014
S11	Towards the understanding and evolution of monolithic applications as microservices [25]	Latin American Computing Conference (CLEI)	2017
S12	EVA: A Tool for Visualizing Software Architectural Evolution [2]	International Conference on Software Engineering (ICSE)	2018

lution of a real tool to demonstrate the use of *BEAGLE*, a prototype implementation of this solution that works as an integrated environment for studying software architecture evolution, as a validation form of the *Origin Analysis* [3]. This paper does not discuss explicitly its limitations, even though cites some of them.

The paper **S02** [17] proposes a graphical technical description of the architectural instance of a software system, and verify the compliance to its corresponding style. This solution is based on the *Bi-graphical Reactive Systems* (BRS) to perform the verification with formal methods, and uses an extended version of a Bigraph to describe the instance. Besides supplying a visual method to specify ar-

chitectural instances and styles, the solution proposed can enhance the ability to design evolving systems. Additionally, the paper shows two study cases in order to prove the effectiveness of this solution [17].

The paper **S03** [4] proposes an approach based on the visual representation of inter-module dependencies and relationships between SA components and modules throughout multiple versions of the software system. The *Semantic Dependency Matrix* is a visualization technique that shows dependencies between two modules with similar behavior classes. The *Edge Evolution Film Strip* is a visualization technique that presents the evolution of an inter-module relations in a software system along its multiples versions. These techniques were applied in two large open source software systems, in reverse engineering context, to exemplify them. The paper also purposes a pattern language for inter-module relationships. The studied examples are provided from an exploration prototype named *Softwareonaut* [4].

The paper **S04** [18] describes the NASA JHU/APL's experiences in using the *SAVE (Software Architecture Visualization and Evaluation)* tool and process. The *SAVE* tool addresses the understanding, maintenance and evolving issues, allowing software architects to navigate, visualize, analyze, compare, evaluate, and improve their software systems, all in only one environment. This tool can be also used to develop a new architecture, compare with the current one and still helps in change impact analysis, among others features. The architecture comparison can also occur between distinct software systems. The paper shows how the *SAVE* tool has been successfully applied to the Common Ground software, a shared software architecture used by NASA missions software systems, in order to avoid further SA maintenance and evolution problems [18]. However, this paper does not discuss the limitations of its study.

The paper **S05** [19] presents the description and goals of the project titled "*Development of a methodology, software-suite and service for supporting software architecture reconstruction*", intended to develop a methodology and a tool-set (environment) to do automatic architecture reconstruction of software systems through visual resources utilization. It also provides tracking of changes in architectural components during software evolution. At the time this paper was written (2010) the project was focused to systems that has been built using Java or .NET technologies and deal with SQL databases. This paper presents limitation of its study, but only in a summarized way. It also shows details of the current status (2010) of this project [19].

The paper **S06** [20] introduces *Evolve*, a graphical modeling tool that implements an ADL (*Architectural Description Language*) named *Backbone* that is focused on software architecture evolution. *Evolve* supports definition and evolution of SA using the *Backbone*, with particular atten-

tion to incremental change and unplanned change processing, very common activities in the software development and evolution process. *Backbone* provides constructs that allow changes that may result in architectural anomalies but *Evolve* is able to detect these anomalies. The paper shows the main characteristics of *Evolve* and how it deal with changes definition in SA, besides a brief use historic. The *Evolve* tool, at the time the paper was published (2011), was freely available for academic research and the production of open source software under the *GNU Affero General Public License version 3*.

The paper **S07** [21] presents and proposes the development of *ARAMIS (Architecture Analysis and Monitoring Infrastructure)*, an architecture meta-model based solution to extract run-time architecture information and provide data to generate new dynamic architecture views in real time. The solution provides visual representations of the monitored architecture at several abstraction levels, as well as the availability of methods to evaluate this architecture [21]. This study does not present any type of implementation (only technical description), despite presents some limitations of the solution.

The paper **S08** [22] introduces *eCITY*, a tool that helps software architects and developers to understand the software structure of their system. It allows the track of components' insertion, removal, or modification over system lifespan and provides an interactive visualization that provides an overview of changes. All of this implemented under a city metaphor using animations to represent the transitions of the architecture components and color coding to highlight the evolution and changes of these components (Figure 4). The *eCITY* provides an overview of the entire system at a desired point into its evolution process (life cycle), implementing it under a city metaphor, allowing the user an interactive way to understand and explore these changes. *eCITY* provides views to help the changes over time, like: *Timeline View*, an administrative view that uses charts and color to emphasize changes between software system versions; and *City View*, a city layout using animations to represent the transitions of the architecture components and color coding to highlight the evolution and changes of these components, as shown in Figure 4. The *eCITY* tool works with compile-time information, not providing dynamic views [22]. The *eCITY* was originally designed as an Eclipse plug-in. This paper also presents a summary of a conducted user study to emphasize its usefulness.

The paper **S09** [23] describes the implementation results of some core characteristics of *ARAMIS*, previously proposed in S07 [21]. *ARAMIS* is a approach for evolution and evaluation of software systems that relies on a infrastructure of run-time monitoring to manage the behavior of the system, in several abstraction levels. In this paper, a prototype of *ARAMIS* was developed focused only in reconstruc-

Figure 4: *eCITY*: a City View [22]

tion of object-level interactions. The prototype uses aspect-oriented techniques to extract and gather the run-time architectural information, and the *XMPP* (*Extensible Messaging and Presence Protocol*) to distribute the gathered information for visualization in real-time, through specialized components, as we can see in Figure 5. The evaluation of this process, according to prototype results, shows that *ARAMIS* can easily be used to demonstrate the behavior of the run-time monitored systems [23].

Figure 5: *ARAMIS*: a Prototype Overview [23]

The paper **S10** [24] presents *eCITY+*, an improved version of *eCITY*, presented in S08 [22], now combining the stable city layout and the *Hierarchical Edge Bundling* (*HEB*), an useful technique to help the implementation of 3D visualization with the use of animation. The *eCITY+* is a later version of *eCITY* tool presented in S08 [22], owning characteristics similar from its predecessor. The *eCITY+* tool primarily differs from its earlier version in the use of *HEB* to highlight changes in both the hierarchical structure as well as the inter-dependencies between software components. The *eCITY+* tool, as its predecessor, performs analysis of software architecture relationships through interactive visual support, using the city metaphor to provide an

overview of the entire system [24]. The *eCITY+* also was consciously developed as a plug-in to traditional SA maintenance tools.

The paper **S11** [25] describes a technical solution to modernize monolithic applications into microservices using software visualization to support the comprehension of evolution process. This conceptual solution can provide a modernization process that uses a legacy system and generates a set of visual diagrams that help architects and developers to understand the system, also suggesting ways of code partitions for transforming into micro-services. The paper has focus only in an understanding stage of modernization, not in transformation stage [25]. The authors analyzed a large Java EE application to validate this solution.

The paper **S12** [2] introduces *EVA* (*Evolution Visualization for Architectures*), a visual tool to help software architects understand the evolution of architecture and therefore track and analyze architectural changes. This tool can visualize and explore architectures of software systems with a long life cycle, including stages of its evolution. *EVA* provides three main views: *Single-Release Architecture View*, that shows the architecture of only one software system version, as shown in Figure 6a; *3-D Architecture-Evolution View*, that depicts architectures of multiple software system versions in a single compositional view, as shown in Figure 6b; and *Pairwise Architecture-Comparison View*, that presents the architectural differences between two software system versions, as shown in Figure 6c. *EVA* allows its users to assess the impact of design decisions, as well as its rationale, which have influenced in the software architecture. As we can see in Figure 6, *EVA* uses color coding to distinct packages or groups of code level entities. The work is currently (2018) focused on showing the explicit reasons behind the architectural changes, in order to assist the rationale of tracking design during the software lifespan [2]. This article presents limitations of its study, but does not discuss them explicitly. The *EVA* tool was developed in Python and is available in a GitHub repository.

Unfortunately, we could not find any taxonomy to classify solutions that support the comprehension of SA evolution using visual resources. Then, to improve the understanding of these visual solutions previously described, we propose a taxonomy with the goal to classify their main characteristics, properties and features. Next, we present the taxonomy and the corresponding classification of each of the selected visual solutions in this SMS.

4.1. Taxonomy for Visual Solutions to Support SA Evolution Comprehension

In this subsection, we present the resulting taxonomy of visual solutions to support the comprehension of software architecture evolution. We argue that the concepts presented in this taxonomy are key to understand the char-

Figure 6: Three Types of Visualization of EVA [2]

acteristics of visual solutions and therefore to answer the research questions of this secondary study. We used the software visualization taxonomy proposed by Price, Baecker and Small (1993) [26] and the framework purposed by Gallagher, Hatch and Munro (2008) [27] as references. The first focused on software visualization, whereas the latter focused on software architecture. For this reason, we adopted them as references to propose the new taxonomy. According to Sulr et al. [11], a taxonomy is consisted of a number of dimensions (e.g., visualization form) with their attributes (e.g., 2D Elements, Color Coding). Each visual solution from the selected papers from this SMS can pertain to one or more attributes from a specific dimension, as we will describe in the following paragraphs.

As can be seen in Table 11, the **Category** dimension is related to the type of proposed visual solution to support the comprehension of SA evolution. It is comprised of four attributes: *Description*, *Technique*, *Tool* and *Environment*. The *Description* attribute classifies the solution or its core idea as a main concept or a technical description of a visual solution. The *Technique* attribute informs whether the visual solution proposes the use of a technique, such as specialized procedures or processes. The *Tool* attribute means the solution presents or proposes a tool, or its development, to assist specialized users of software architecture to develop, or maintain software systems. The *Environment* attribute indicates that the solution explicitly presents or proposes an environment, i.e., an integrated set of tools to help software architecture specialized users in the development or maintenance of software systems. The value of this dimension for each solution is mandatory and it may have more than one attribute signalized.

The **Stage** dimension has four attributes. The *Conceptual* attribute indicates that the solution is still represented and referenced as a *concept*, not having any implementation. The *Project* attribute indicates that the solution has been under development as a project. The *Prototype* attribute means the solution is a prototype of the solution. Finally, the *Stable Release* attribute is related to the status of the solution as already in production as a stable release. The value of this dimension is also mandatory and only one attribute can be signalized for each solution.

The **Visualization Form** dimension defines the visual

characteristics of the output of the solution [26]. It is characterized by eight attributes as follows. The *2D Elements* attribute indicates that the solution uses 2D elements, such as 2D charts, diagrams, shapes, windows, figures and lines. The *3D Visualization* attribute indicates that the solution uses 3D resources for visualization. The *Animation* attribute means the solution uses resources of animation in the visual representations. The *Bigraphs* attribute marks the use of bigraphs in the visual solution. The *Visual Metaphor* attribute indicates the use of one or more visual metaphors in the visual representations to enhance the SA evolution understanding. The *Color Coding* attribute means the solution adopts a specific color system to represent the data. The *Tree-based* attribute is used to characterize solutions that use visual structures based on trees. The *UML-based* indicates that the solution uses UML diagrams as a form of visual representation.

The **Static Representation** dimension shows what architectural information can be extracted and represented before run-time [27]. It has two associated attributes: *Static Visualization* and *Recovery*. The *Static Visualization* attribute means the solution displays data exclusively related to static structure of the software system. The *Recovery* attribute indicates that the solution supports the retrieval of architectural data from specific sources. This dimension may have no attributes flagged.

The **Dynamic Representation** dimension shows what architectural information can be extracted and represented during run-time [27]. The *Dynamic Visualization* attribute means the solution displays data extracted during its execution (run-time). The *Events Monitoring* indicates if the solution perform the catch events during its execution. These events can be identified and associated with SA elements and thereby support the comprehension of specific scenarios of software architecture evolution. The *Live* attribute points out that the SA data is gathered in a *real time* fashion as the solution is executed [26] [27]. The *Post-mortem* attribute means the SA data to be gathered is produced in a *post-mortem* fashion by the solution, i.e., generated by its previous execution [26] [27]. This dimension may have no attributes flagged.

The **Architectural Tasks** dimension is related with features of the visual solution that support stakeholders to perform tasks that to some extent focuses on the software architecture and its evolution [27]. It has nine attributes as follows. The *Anomalies* attribute indicates that the solution supports the identification of anomalies, violations and inconsistencies occurrences related to SA. These occurrences can also influence the *Comprehension* attribute indicates the solution supports visual analysis tasks to improve the comprehension of SA and its evolution. Analysis tasks means tasks that generate results to facilitate the understanding the SA, its components, dependencies and relationships, as well

as its evolution. They should support top-down or bottom-up approaches [27]. The *Styles* attribute indicates that the solution is able to identify architectural styles and/or verify its compliance with a predefined reference. The *Show Evolution* attribute indicates that the solution provides facilities to exhibit evolution evidence of a SA, in a basic or advanced way [27]. The *Construction* attribute indicates that the solution provides resources to add, change or remove SA elements in the visual representation. The *Evaluation* attribute means the solution supports SA quality analysis and also compliance evaluation. The *Comparison* attribute points out that the solution performs visual comparison among releases of the software system under analysis. A typical use of this attribute is the comparison between the *as-is* with the *to-be* architectures or the *as-designed* with *as-implemented* software architecture [27]. The *Tracking* indicates that the the solution supports the tracking of SA changes throughout its releases. This is a key resource to, for example, identify and trace the architectural decay of a SA, which impairs the software lifespan [2]. Finally, the *Rationale* indicates that the solution presents and make available the rationale behind the design decisions that somehow influences the SA.

4.2. Visual Solutions According to Taxonomy

The Table 11 shows the main characteristics, properties and features identified in the visual solutions of the selected papers from the perspective of the proposed taxonomy. These characteristics were identified and collected exclusively based on the text provided by the selected studies listed in Table 10.

The column labeled *Category* indicates different categories of solutions found in the selected papers, according the description presented in Subsection 4.1. The column labeled *Stage* means the stage of the solution proposed by the paper at the time it was published. The column labeled *Visualization Form* means the summary description of fundamental characteristics related to what can be exhibited in the visual solution. The content of this column is based on the *Form* category proposed by the taxonomy of Price [26]. The columns labeled *Static Representation*, *Dynamic Representation*, *Architectural Tasks* are based on keys areas proposed in Gallagher’s framework [27]. The column *Others Features* shows complementary purposes, features and characteristics of the presented visual solution not listed before.

We decided to present a analysis about the visual solution named *EVA* to illustrate how the characteristics of the selected visual solutions presented in Table 11 can be determined. This analysis is shown in Table 12. The visual solution *EVA* was previously presented in the paper S12 [2] and its justifications come exclusively from this paper’s content.

The Figure 7 presents the overview of current stages for all visual solutions referenced by Table 11. Its worth re-

membering that current stages means the stage at the time its study was published. Note that most of solutions (5 in 12) were in “Stable Release” stage, whereas few of them (2 in 12) were in “Conceptual” stage.

Figure 7: Current Stages of Visual Solutions

The Figure 8 shows the categories of solution found in selected papers. The “Tool” category has the major preference in visual solutions, being present in 8 of 12 solutions found. The “Description” category is present in only two solutions.

Figure 8: Categories of Visual Solutions

Figure 9 shows that the tasks *Comprehension* and *Show Evolution* are adopted in all visual solutions from the selected studies. It means that all of them reported the support of analysis tasks to improve the SA comprehension. Moreover, they also reported the adoption of facilities to exhibit the SA evolution. These are in fact, minimum requirements of a visual solutions to support comprehension of SA evolution. The task *Comparison* is also representative in the analyzed solutions. They have reported the ability to perform visual comparison of SA characteristics among two or more releases of a specific software system, which corresponds to 83% of analyzed visual solutions. The visual solutions not related to the task *Comparison* is the ARAMIS solution (S07) [21] and (S09) [23]. These papers reported the focus in real-time monitoring. The tasks *Anomalies*, *Styles* and *Rationale* were not representative in the solutions, corresponding to only 17% . The task *Construction* explicitly appears in 33% of the visual solutions and half of them

Table 11: Using the Proposed Taxonomy to Classify the Visual Solutions to SA Evolution

Ref. Paper	Name of Visual Solution	Category	Stage	Visualization Form	Static Representation	Dynamic Representation	Architectural Tasks	Other Features
S01	Beagle	Environment, Technique	Prototype	2D Elements, Tree-based	Static Visualization, Recovery	N/A	Comprehension, Comparison, Show Evolution	Evolution metrics usage
S02	Not named	Description, Technique	Conceptual	2D Elements, Bigraph	Static Visualization	Dynamic Visualization, Live	Comprehension, Comparison, Construction, Show Evolution, Styles	BRS resources
S03	Film Strip and Dependency Matrix	Technique	Prototype	2D Elements	Static Visualization, Recovery	N/A	Comprehension, Comparison, Show Evolution	N/A
S04	SAVE	Tool, Environment	Stable Release	2D Elements	Static Visualization, Recovery	N/A	Comprehension, Anomalies, Comparison, Construction, Show Evolution, Rationale, Styles, Evaluation	Process-oriented, Products comparison
S05	GOP	Tool, Environment	Project	2D Elements, Color Coding	Static Visualization, Recovery	N/A	Comprehension, Comparison, Construction, Show Evolution, Tracking	Methodology-oriented
S06	Evolve	Tool	Stable Release	2D Elements, UML-based, Animation	Static Visualization	N/A	Comprehension, Anomalies, Comparison, Construction, Show Evolution	Model-driven, ADL implementation
S07	ARAMIS	Tool	Conceptual	N/A	Static Visualization	Dynamic Visualization, Events Monitoring	Comprehension, Show Evolution, Evaluation	Model-driven
S08	eCITY	Tool	Stable Release	2D Elements, Color Coding, Animation, Visual Metaphor	Static Visualization	N/A	Comprehension, Comparison, Show Evolution, Tracking	N/A
S09	ARAMIS	Tool, Technique	Prototype	2D Elements, UML-based	N/A	Dynamic Visualization, Events Monitoring, Live, Post-mortem	Comprehension, Show Evolution, Evaluation	Model-driven, Traceability with requirements, Views creation
S10	eCITY+	Tool, Technique	Stable Release	2D Elements, 3D Visualization, Color Coding, Animation, Visual Metaphor	Static Visualization	N/A	Comprehension, Comparison, Show Evolution, Tracking	As-plugin
S11	Not named	Description	Project	2D Elements, UML-based, Color Coding	Static Visualization	N/A	Comprehension, Comparison, Show Evolution	Model-driven, Modernization
S12	EVA	Tool	Stable Release	2D Elements, 3D Visualization, Color Coding	Static Visualization, Recovery	N/A	Comprehension, Comparison, Show Evolution, Tracking, Rationale	ADD Traceability

present specific resources to design the architecture, such as S02 solution [17] (using *Biagraphical Reactive System*) and S06 solution [20] (using an ADL visual modelling component).

5. Discussion

The specific research question SRQ1 is related to the main visual solutions to support the software architecture evolution comprehension. The answer is presented in Table 11, where it is possible to identify their main goals and characteristics of each solution. The specific research question SRQ2 is concerned to different purposes of using the visual solutions to support the software architecture evolution comprehension. The purposes are also presented in Table 11, identified primarily in the *Architectural Tasks* column and also in the *Static Representation*, *Dynamic Representation* and *Other Features* columns. The specific research question SRQ3 focuses on solutions designed to visually support comprehension of software architecture evolution can be classified. The solutions can be classified in the categories *Description*, *Technique*, *Tool* and *Environment*.

The Table 11 classify each solution and shows that the *Tool* category has more representants among the selected studies. The specific research question SRQ4 focuses on visual forms used to support comprehension of software architecture evolution. The Table 11 uses the *Visualization Form* column to list the visual forms adopted in the visual solutions discussed in the selected studies.

Finally, the main research question (RQ) focuses on the evaluation of the usage of visual solutions to support the comprehension of software architecture evolution based on papers published in the peer-reviewed literature. Table 11 presents an up-to-date overview of solutions used for the stated purpose with different characteristics and strategies as has been already explained for the specific research questions. We have identified that based on evidence from the selected studies, features of comprehension and SA evolution visualization are minimum requirements to support software architecture evolution comprehension, as shown in Figure 9. On the other hand, the low number of papers found in this systematic mapping study, suggests that visual solutions to support SA evolution comprehension is an area

Figure 9: Distribution of Architectural Tasks in the Selected Papers

Table 12: Sample Analysis - Visual Solution *EVA* [2]

Values	Justifications
Category = ("Tool")	Title of the paper is " <i>EVA: A Tool for Visualizing Software Architectural Evolution</i> ".
Stage = ("Stable Released")	The behavior of <i>EVA</i> presented in the paper always suggests that it is in production; <i>EVA</i> was already evaluated in use and is in the process of deploying to a large development organization.
Visualization Form = ("2D Elements", "3D Visualization", "Color Coding")	"2D Elements": suggested by figures displayed in the paper; "3D Visualization": <i>EVA</i> provides a view that shows architectures of multiple releases in a 3D visualization; "Color Coding": <i>EVA</i> differs code-level entities through color coding.
Static Representation = ("Static Visualization", "Recovery")	"Static Visualization": <i>EVA</i> only collects data at compile-time (static elements); "Recovery": <i>EVA</i> supports many SA recovery techniques.
Dynamic Representation = N/A	The paper does not present any characteristic or feature for dynamic representation.
Architectural Tasks = ("Comprehension", "Comparison", "Show Evolution", "Tracking", "Rationale")	"Comprehension": <i>EVA</i> provides a view that represents a single SA release, allowing users to interactively understand the functionality of each architecture component; "Comparison": <i>EVA</i> provides comparison view that shows the SA differences between two releases; "Show Evolution": <i>EVA</i> visualizes the SA evolution through a set of source codes across multiple releases; "Tracking": <i>EVA</i> provides a view that allows representation of change tracking of entities across multiple releases; "Rationale": <i>EVA</i> collects relevant data of design decisions from system repositories, displaying them together with the architecture evolution visualization.
Other Features = ("Add Traceability")	<i>EVA</i> shows the traceability of architecture design decisions over time

that needs expand in terms of studies and options available for practitioners and researchers.

6. Conclusion

The aim of this work is to report the design, execution and results of a systematic mapping study of visual solutions to support the comprehension of software architecture evolution. We performed a SMS according to the plan described in Section 3. Initially, the applied search strings retrieved 92 papers from the selected electronic databases. This number was reduced to 12 after applying all selection procedures criteria. From these 12 selected studies, the identified visual solutions to support the comprehension of SA evolution were classified as follows: 17% categorized as *Description*, 42% as *Technique*, 67% as *Tool* and 25% as *Environment*. All of them support the architectural tasks *Comprehension* and *Show Evolution*.

Besides the identification of visual solutions from the literature, another contribution of this study is a taxonomy to classify these solutions. The taxonomy contains six dimensions: category, stage, visualization form, static representation, dynamic representation and architectural tasks. These dimensions and their attributes was explained in Section 4, SubSection 4.1. Each visual solution was classified according to this taxonomy, generating a table of characterization of visual solutions to SA evolution, presented in Table 11. The assignment of the taxonomy attributes value to the visual solution characteristics shows a current overview of the available visual solutions in peer-reviewed literature.

Moreover, this study also shows that visual solutions to support SA evolution comprehension usually present features to support analysis tasks to improve the SA compre-

hension and also provide facilities to exhibit the SA evolution. This study also concludes that, due to a few number of papers found in this SMS, the studies may consider to allocate more research and development effort to provide effective visual solutions to support SA evolution comprehension, improving the acquired knowledge in this area.

As a future work, we recommend extending the research to establish a methodology or process, based on the taxonomy proposed, to define projects to build visual solutions of SA evolution comprehension. Another possibility for future work is the improvement of the proposed taxonomy, aiming to generate a new framework to objectively evaluate solutions like that ones discussed in this SMS.

References

- [1] H. P. Breivold, I. Crnkovic, and M. Larsson, "A systematic review of software architecture evolution research," *Information and Software Technology*, vol. 54, no. 1, pp. 16 – 40, 2012.
- [2] D. Nam, Y. K. Lee, and N. Medvidovic, "Eva: A tool for visualizing software architectural evolution," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, May 2018, pp. 53–56.
- [3] Q. Tu and M. W. Godfrey, "An integrated approach for studying architectural evolution," in *Proceedings 10th International Workshop on Program Comprehension*, 2002, pp. 127–136.
- [4] M. Lungu and M. Lanza, "Exploring inter-module relationships in evolving software systems," in *11th European Conference on Software Maintenance and Reengineering (CSMR'07)*, 2007, pp. 91–102.
- [5] R. Taylor, N. Medvidovic, and E. Dashofy, *Software Architecture: Foundations, Theory and Practice*, J. Wiley and Sons, Eds., Hoboken, New Jersey, 2009.
- [6] D. Garlan, J. M. Barnes, B. Schmerl, and O. Celiku, "Evolution styles: Foundations and tool support for software architecture evolution," in *2009 Joint Working IEEE/IFIP Conference on Software Architecture European Conference on Software Architecture*, 2009, pp. 131–140.
- [7] M. Shahin, P. Liang, and M. A. Babar, "A systematic review of software architecture visualization techniques," *Journal of Systems and Software*, vol. 94, pp. 161 – 185, 2014.
- [8] M. Shahin, P. Liang, and M. R. Khayyambashi, "Improving understandability of architecture design through visualization of architectural design decision," in *Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge*, ser. SHARK '10, 2010, pp. 88–95.
- [9] R. L. Novais and M. G. de Mendonca Neto, *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications*, ser. Chapter: Software Evolution Visualization: Status, Challenges, and Research Directions. IGI Global, 2018.
- [10] A. Telea, L. Voinea, and H. Sassenburg, "Visual tools for software architecture understanding: A stakeholder perspective," *IEEE Software*, vol. 27, no. 6, pp. 46–53, 2010.
- [11] M. Sulr, M. Bakov, S. Chodarev, and J. Porubn, "Visual augmentation of source code editors: A systematic mapping study," *Journal of Visual Languages Computing*, vol. 49, pp. 46 – 59, 2018.
- [12] C. Wohlin et al., *Experimentation in Software Engineering*. Springer-Verlag, 2012.
- [13] V. R. Basili and H. D. Rombach, "The tame project: towards improvement-oriented software environments," *IEEE Transactions on Software Engineering*, vol. 14, no. 6, pp. 758–773, 1988.
- [14] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, and M. Mirakhorli, "The twin peaks of requirements and architecture," *IEEE Software*, vol. 30, no. 2, pp. 24–29, 2013.
- [15] T. Dyb and T. Dingsyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9, pp. 833 – 859, 2008.
- [16] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, P. Group et al., "Preferred reporting items for systematic reviews and meta-analyses: the prisma statement," *PLoS medicine*, vol. 6, no. 7, p. e1000097, 2009.
- [17] Z. Chang, X. Mao, and Z. Qi, "An approach based on bigraphical reactive systems to check architectural instance conforming to its style," in *First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE '07)*, 2007, pp. 57–66.
- [18] W. C. Stratton, D. E. Sibol, M. Lindvall, and P. Costa, "The save tool and process applied to ground software development at jhu/apl: An experience report on technology infusion," in *31st IEEE Software Engineering Workshop (SEW 2007)*, 2007, pp. 187–193.
- [19] L. Schrettnner, P. Hegedus, R. Ferenc, L. J. Fulop, and T. Bakota, "Development of a methodology, software – suite and service for supporting software architecture reconstruction," in *2010 14th European Conference on Software Maintenance and Reengineering*, 2010, pp. 190–193.
- [20] A. McVeigh, J. Kramer, and J. Magee, "Evolve: Tool support for architecture evolution," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11, 2011, pp. 1040–1042.
- [21] A. Dragomir and H. Lichter, "Model-based software architecture evolution and evaluation," in *2012 19th Asia-Pacific Software Engineering Conference*, vol. 1, 2012, pp. 697–700.
- [22] T. Khan, H. Barthel, A. Ebert, and P. Liggesmeyer, "ecity: A tool to track software structural changes using an evolving city," in *2013 IEEE International Conference on Software Maintenance*, Sep. 2013, pp. 492–495.
- [23] A. Dragomir and H. Lichter, "Run-time monitoring and real-time visualization of software architectures," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, vol. 1, 2013, pp. 396–403.
- [24] T. Khan, S. R. Humayoun, K. Amrhein, H. Barthel, A. Ebert, and P. Liggesmeyer, "ecity+: A tool to analyze software architectural relations through interactive visual support," in *Proceedings of the 2014 European Conference on Software Architecture Workshops*, ser. ECSAW '14, 2014, pp. 36:1–36:4.
- [25] D. Escobar, D. Crdenas, R. Amarillo, E. Castro, K. Garcs, C. Parra, and R. Casallas, "Towards the understanding and evolution of monolithic applications as microservices," in *2016 XLII Latin American Computing Conference (CLEI)*, 2016, pp. 1–11.
- [26] B. A. Price, R. M. Baecker, and I. S. Small, "A principled taxonomy of software visualization," *Journal of Visual Languages Computing*, vol. 4, no. 3, pp. 211 – 266, 1993.
- [27] K. Gallagher, A. Hatch, and M. Munro, "Software architecture visualization: An evaluation framework and its application," *IEEE Transactions on Software Engineering*, vol. 34, no. 2, pp. 260–270, 2008.

Design of a Dashboard of Software Metrics for Adaptable, Energy Efficient Applications

Vladimir Ivanov, Daria Larionova, Dragos Strugar, Giancarlo Succi

Innopolis University

Innopolis, Russia

{v.ivanov, d.larionova, d.strugar, g.succi}@innopolis.ru

Abstract—With the proliferation of internet-based software solutions worldwide came the need to maintain these products as well as to monitor the metrics collected within the products themselves. It has been a major challenge throughout the years to come up with an effective User Interface (UI) design that could be tailored to employees with multiple roles in the organization, especially in rapidly changing modern environments. Developing a self-adjusting dashboard not only could increase managers' productivity, but could potentially grow into a fully-fledged Adaptable System. However, such a system would carry a broad set of additional requirements. This paper presents the synergy of the adaptable systems in production and effective dashboard design according to the industry standards.

Index Terms—Adaptable Systems, Software Metrics, Energy-saving Applications

I. INTRODUCTION

The goal of our ongoing research was to come up with a self-adjusting metric analysis system that would allow managers as well as engineers to make more informed decisions in the development of Adaptable, Energy Efficient solutions.

Developing a product first requires an understanding of the problem that is expected to be solved. Therefore, for this system a customizable dashboard can be used, which is a well-proven solution for this purpose.

Thus, we have created a web application to demonstrate our vision of the dashboard best-suited for this use case. The purpose of the global data collection system, to which the dashboard belongs to, is to present the obtained data in a specific structure and form, as well as to provide the users with an overview of the collected information.

Numerous academic papers emphasize the importance of the good design for the dashboard. In [4], Few points out that currently the entire purpose of dashboards is not only to display all the necessary information, but rather to provide a medium for communication and team collaboration.

The literature review section is primary based on the works of Zorin [7], [18], Pishulin [6], [12], Valiullin [16], whose preceding studies grounded the system that we are currently developing. It is also based on the work of Sarikaya, Correll, Bartram, Tory, and Fisher [13], who performed a systematic literature review on designing dashboards for different domains. These authors split the dashboards into types by different criteria, and explained the common patterns and features for each of them. Finally, it is based on the work of Yigitbasioglu, and Velcu [17], who also reviewed many

sources for designing the dashboard, and suggested the mechanism for making decisions in their functionality.

Section II in greater detail the common problems which developers of the dashboard are likely to encounter. Section III summarizes fundamental goals and metrics of the GQM (Goal Question Metric) model, as required by software engineers for the evaluation of their performance. Section IV covers possible features of the future dashboard. Section V lists some common visualization patterns. Section VI shows the overall design of the working solution. Then, sections VII and VIII talk about the Complex Adaptive Systems and how they map to our use case. And finally, section IX gives the final thoughts and reflections.

II. PROBLEMS DESIGNATION

According to Valiullin [16], one of the main challenges is in selecting more appropriate metrics and displaying them in a meaningful and structured way. This choice has great influence on the understanding of those metrics. The dashboard should “provide intuitive, actionable, flexible, and programmable visualization to support effective decision making.”

In addition, effective representation of selected metrics [16] is also a great challenge that we were facing. Solving this problem would allow users to easily detect and address the issues that may occur during the development process.

To solve these problems and choose more appropriate visualization techniques for more effective representation, Brath and Peters [1] suggest answering the following three questions:

- What metrics does the user need to see?
- What context does each metric require to make it meaningful?
- What is the visual representation that best communicates the metric?

In [14], the authors suggest to focus on the important metrics, with concise visualization. Such an approach that reduces redundancy by focusing on the goals obtained during the requirement collection & analysis phase is very useful. Moreover, it helps to cope with making the wrong decisions in certain situations [17]. A similar view on data visualization is presented in [8], where the authors insist that the dashboard should be designed to be useful. This does not mean that it should contain all possible information that can be visualized, but only the necessary and sufficient data. The authors supplement this idea by suggesting the concepts of the “right”

Fig. 1. Correspondence of the dashboard types and functionality. (Taken from [3, p. 103])

data and “right” visualization technique. Therefore, it is first necessary to determine which metrics should be displayed and explain the rationale of their choice. Second, choose the way to present them to the user, which can consequently minimize the time required for understanding them.

In addition, as noticed in [13], the design of the dashboard and its functionality greatly depends on its type. Few [4] and Eckerson [3] introduced three types of dashboards. Eckerson represents them in the form of a pyramid, with the operational dashboard type at the base, the tactical in the middle layer, and the strategic at the top (as shown in figure 1). The three dashboard types differ in purpose and level of abstraction and interaction. According to this model, the strategic view is intended for monitoring the current situation. This allows minimum interaction with the user, contains an overview and consists of the most meaningful data. This is because, as mentioned in [13], the people often make screenshots and put them onto slides for showing a general picture. The purpose of the tactical view is to enable the analysis. This should contain more detailed information from the tactical view. The operational view should present a thorough form of the collected data and metrics. That would allow the user to summarize and also would help to find the reasons which led to this state, and come to possible solutions. In addition, according to the results of the surveys by Zorin in [18], the operational becomes more effective than the tactical and the strategic dashboard type.

In [17], Yigitbasioglu and Velcu suggest a path for making design decisions, that requires determining the four characteristics of a dashboard under development:

- its purpose is enabling
 - consistency,
 - monitoring,
 - planning, and,
 - communication,
- users and their
 - tasks,
 - knowledge and
 - cognitive styles,

- design features (functional or visual) and
- outcomes, that is performance progress reached by this dashboard (i.e. improved speed, consistency etc.).

Taking into account that respondents of Zorin’s surveys [18] are potential users, we can presume the variance in their education levels, age, company size and the position they are working on. Users prioritized their needs in the dashboard in the following order from the most to the least demanded:

- 1) performance monitoring;
- 2) planning;
- 3) communication;
- 4) measurement consistency.

Therefore these needs along with their priorities should be taken into account when architecting a new dashboard from scratch. The next couple of sections provide a more comprehensive description of how these can be formed in the QQM model.

III. THE MAIN GOALS AND METRICS

For selecting the necessary metrics in [8], the authors suggest using QQM+Strategies. This concept is a traditional Goal-Question-Metric approach supplemented by the links between different layers of organizational goals, i.e., high-level and measurement goals. According to this approach, the authors ensure that a good dashboard should meet business goals. Hence, the development of such a dashboard requires the participation of all stakeholders.

As part of the study [18], Zorin conducted surveys with the representatives of software engineering companies, and extracted six common goals they wanted to achieve:

- Improving effort estimation efficacy;
- Using resources in a more efficient way;
- Executing testing activities in a more efficient and systematic way;
- Improving the quality of the development process;
- Completing projects successfully;
- Completing projects phases successfully.

Later in [7], Zorin summarizes them, and extracts three main goals:

- more effective effort estimation;
- more efficient use of resources;
- better software quality and development process.

The author divides the metrics needed for evaluating the achievement of these goals into five groups, displaying the:

- Progress status of the project;
- Speed of the work performed;
- Status of testing;
- Status of software quality;
- Effectiveness of effort estimation.

In addition, Zorin distinguishes the most frequent metrics:

- 1) Iteration Burndown chart;
- 2) Team velocity;
- 3) Code coverage;

- 4) Effort estimation accuracy.

Finally, the author summarized the results in the GQM model, shown in figure 2.

In [6], [12], Pishulin validated the results of the surveys by Zorin [7], [18]. The author investigated three goals highlighted in those studies, and determined the most suitable metrics for measuring them. For assessing the effectiveness of effort estimation, Pishulin identified the following key metrics:

- 1) Iteration Burndown;
- 2) Effort Estimation Accuracy;
- 3) Team Velocity.

For the evaluation of software quality and development process, the metrics are:

- 1) Passed Tests;
- 2) Code Coverage;
- 3) Unresolved Defects;
- 4) Class / Method Length;
- 5) Iteration Burndown;
- 6) Defect Removal;
- 7) Defected Density.

For the goal of more efficient use of resources, the author did not provide any information due to the lack of obtained information.

However, in [16], Valiullin suggested to spread this set of primary metrics and allow the user to create new metrics from existing ones by combining them in different ways. For describing the available manipulations, the author splits metrics to three types:

- raw,
- composite
- expression.

The raw metrics are those extracted directly from the collected user activities and source code (primary set of metrics described above). The composite metrics are the ones constructed from two raw or composite metrics by applying:

1. Simple arithmetic operations:
 - 1.1. Addition;
 - 1.2. Subtraction;
 - 1.3. Multiplication;
 - 1.4. Division;
2. Simple mathematical functions:
 - 2.1. Average;
 - 2.2. Maximum;
 - 2.3. Minimum.

The expression metrics are obtained from one or more Raw or Composite type, and can be aggregated by some user-defined arithmetic expression.

IV. FUNCTIONALITY

The main mission of the dashboard is allowing the user to monitor key metrics in terms of completion of predefined goals. Furthermore, the dashboard should provide various toolkits for tracking the development progress and product quality, and executing comparison of the current and perfect

values [18]. Despite the amount of functionality, it is substantial to fit the dashboard to a single computer screen without compromising the content [4], [17].

In [8], the authors point out that understanding the displayed data should require the minimum effort from the user. The most relevant information should be provided with a “push” strategy. But at the same time, support interactivity with the ability to switch to “pull” mode. Moreover, the authors notice that the most relevant information should attract user’s attention. However, at the same time, it is very important to find a balance and avoid a motley design. The authors suggest displaying the same data every time in the same place in order to make users become accustomed to the design.

One of the main features that the dashboard should implement is alerting users about the deviations of measurements from normal values. In [16], the author suggested notifying the user about abnormalities by coloring metric tiles according to their trend. For example, red could be used for going out of predefined by the user predefined range, green for those in the range and any other for neutral or default. [8], [12], [13] covered an idea of using arrows for showing the current tendency. The method in [3] can be useful for color-blind people to help them with correct interpretation of results. At the same time, it is significant not to overuse colors, visual structures and other catchy elements of design, in order not to overload the view [17].

According to [13], it is significantly important to make the dashboard customizable and adaptable to different users and situations. This objective can be obtained by allowing the users to set the goals, by themselves select the metrics to be displayed and set their own admissible and critical borders. Furthermore, the author draws our attention to the fact that filters [13], [16] and comparison support [13] are able to supplement the dashboard with more flexibility.

In [16], Valiullin continues this idea, and notes that the system under study should be adaptable for any project goal and provide multivariate analysis. The author states, that this adaptability can be satisfied with the implementation of roles and settings for choosing metrics for each project and goal individually. Furthermore, since users can work on several projects at the same time, for convenience of monitoring their status, the dashboard should provide support for multiple projects, and define separate sets of metrics and goals for all of them [16].

Another frequently mentioned concept lies in presenting different views on collected data and metrics, which can be reached by two ways.

The first one is to introduce roles with specific goals. Therefore, according to the goals, different analysis and visualization techniques should be selected with different points of view about the data [2]. In addition, roles can help to solve the problem of data visibility and privacy [13].

The second one is providing different levels of abstractions and different degree of details presented to the user. For introducing the general view about the situation, there should be a way to combine many metrics to several numbers, and

Fig. 2. Summary of the GQM with the most common aspects in bold (Taken from [7])

avoiding both detailed and redundant information [2]. More complete and thorough views can be reached by drill-down navigation or an “exploration mode” [4], [11], [13], [16]. It means that for “exploring” the nature of a certain result, it should be possible to switch between a generic and a more detailed view of the same metrics down to the raw data view. Using this technique would provide users with convenience in accuracy of data analysis, performed automatically [13].

According to [6], respondents want to have the ability to watch individual team members’ metrics that can be satisfied by introducing hierarchical views to collected data, on the part of individual developers, teams and even the whole company [16]. Furthermore, interviewees would like the dashboard to display possible reasons of metrics changes and recommendations for recovering and improvement of current situation. Thus, the what-if simulation can take place [13], [18].

V. VISUALIZATION

Considering the structure of the dashboard in [16], Valiullin, relying on solutions from [8], [9] suggests the design based on Andon board with tiles, where each tile shows a numeric metric. This representation can speed up the understanding of the data and decision making as well as providing a unified way of metrics visualization in order to maintain the system scalability. However, there pointed a need in specific views for some metrics.

In [17], the authors make a investigation on data representation formats. Some researchers prefer graphical forms to tabular ones, whereas some place tables over graphs. Others do not exalt any of them believing that the choice of representation format highly depends on the task it is intended to solve. Graphs are more useful for tasks implicating comparisons and studying relationships of data, while the tabular form is more

suitable for obtaining the certain information. Thereby, the authors admit possibility of switching to the displaying format more preferable by the user. In [16] for detailed view of the metrics, it is suggested to use two representations: values the metric consists of and the chart with its behaviour.

[5] suggests using different treemaps, texture, and bump mapping; animated zooming and panning for visualization of metrics. However, in [3] the author underlines that sometimes even at first glance simple features can bring additional complexity. The ease of use becomes one of the most important characteristic of any product. As noticed in [13], some authors even suggest reducing the interactivity for simplifying the system. Therefore, finding a balance in flexibility and customization is crucial.

VI. RESULTS

This section focuses on establishing a strong relationship between the metrics and the goal. We are also going to present screenshots and various UI components we created. Results are presented in a form of a web page built with state-of-the-art front end web development library React.js.

As we noted earlier, the combination of flexibility and customization is a major factor in the User Experience (UX). That is precisely why we chose to create numerous widgets that users have control over. These widgets encapsulate graphs, charts, percentages, numerical values and time management tools alongside other key functionalities. They are designed to be easily reusable and re-sizeable. This also contributes to the fact that the solution is fully responsive and works on all screen sizes. Some of the examples you can see on figures 4, 5, and 6.

Fig. 3. HTML Web Interface Results (InnoMetrics)

Fig. 4. Doughnut Graph Representation

Fig. 6. Radial Graph Representation

Fig. 5. Polar Graph Representation

Figure 3 showcases all the parts of the dashboard. The page contains 14 metrics selected for the study. Each of these metrics is placed in one of the widgets, or so called tiles. There is a total of 3 bar graphs, 2 line charts, 4 percentage-based values, 4 numerical values and 1 date value.

This prototype assumes that the software development team is using the Scrum methodology, with 7 iterations completed. Each iteration is two weeks long. The task of this prototype is to find out if the relationship between the metrics and the goal is obvious to the engineers. In addition, it is also very important to immediately see the most representative metrics for the Metric-Goal relationship.

This is precisely why we chose to make the connection with Adaptable systems. Not only would applying the notions from Complexity Theory allow the dashboard to present the most important metrics for certain situations, but it would also prevent it from being susceptible to change.

VII. COMPLEX ADAPTIVE SYSTEMS

Realizing that the dashboard needs to adapt to rapid change in today's software development life-cycle, we suggest that modern dashboards should be a part of Complex Systems, more specifically, the Complex Adaptive Systems.

We represent each of the software development metrics, depicted as widgets on the 3, as agents within the system. Some of them are dependent on one another and some of them are not. The ones who are, are not strictly interacting in a linear, predictable way.

By embracing adaptation these agents have the potential to synchronize their internal states with the other agents in the system. Additionally, the system should be able to recognize these changes and self-adjust with the emergence of globally coherent patterns of adjustment developing.

Then, this Complex Adaptive System, the dashboard, should be able to feed back this information to micro-level agents. In order to preserve the relevance of certain metrics, the system makes the natural selection based on their fitness criteria to the environment. In our case, the fitness function is broadly defined as follows: "An agent has a higher contribution/fitness to the overall system if and only if a slight change in that specific metric would yield a significant change in the overall system fitness, the difference between the expected value and the actual value is above average, or a metric answers some custom questions that dashboard users may have". Applied and contextualized to our use case, the metrics are more relevant if the current value greatly differs from the expected one, or a slight change in a specific metric may yield a substantial increase in the well-being of the entire system.

Such a system embraces the complex dynamic between the micro-level components (metrics) and the overall system (the dashboard). The interaction between the differentiation of micro- and macro-level agents with different goals and agendas creates the core dynamic of complexity in our system.

Intrinsic to the Adaptive Systems is the notion of innovation; i.e. coming up with novel outcomes that we could not have predicted ahead of time. This is the crucial aspect of designing and developing a dashboard engineered to dynamically present the most useful metrics to the person using the dashboard. For example, a correlation between some two metrics may yield a significant improvement in the overall system functioning. However, that relationship has not been made beforehand. It is only by feeding the data to the system that it is able to come up with the correlations which may drastically change the overall system fitness.

The next two notions that need to be explored are Non-linear Dynamics [15] and Dynamic Equilibrium [10].

A. Non-linear Dynamics

We realize that common assumptions that managers and stakeholders may have when creating and maintaining the dashboards for their products are very often not close to the real-world outcome. For example, they may think that one metric may be very important, and that turns out not to be true. And over time that results in a waste of their time looking

at the metric and trying to improve it, while not focusing on the metrics that are actually significant at the moment. Put differently, the common assumptions that stakeholders may have about predicting the outcome based only on the initial input do not often work out in Complex Adaptive Systems due to their emergent complexity. Non-linear dynamics in adaptive systems is necessary to constantly change the internal states of the agents within the system, resulting in the change the entire system's state.

B. Dynamic Equilibrium

Utilizing the concept of the Dynamic Equilibrium allowed us to embrace our GQM model to come up with the state which has the following characteristics:

- the current state is never completely stable, which results in the full stagnation
- the current state is never in complete chaos, where there is nothing to bind individual actors together
- the current state is always in a so called "Dynamic Equilibrium" where all actors are loosely bound to each other with the plethora of room to innovate and improve

VIII. APPLYING THE COMPLEX SYSTEM TECHNIQUES TO OUR USE CASE

As previously noted, the agents in the Complex System are represented as widgets. Each of these widgets encapsulates one metric that is used for maintaining and monitoring the software development process. Different visual representation apply to each of these metrics, of course. Some of them are more useful and easier to think about when depicted as continuous graphs, some of them as plan numbers, as seen on figure 3.

It is then challenging to decide how these metrics can bring the variety, and the non-linear dynamics that Complex Systems impose. There is a total of 14 metrics. It is only by combining them that we will be able to achieve what we have suggested above. We needed to have a synergy between multiple metrics to see how they correlate to one another and how the increase in a specific factor or a group of factors can influence the whole system.

That is specifically why we decided to create custom widgets that represent the correlation between two or more metrics in the system. Even though such functionally complex widgets may contain a lot of data, as they are a relevant component to look after, they play a crucial role in the self-sustainability and continuous evolution of the dashboard system.

An example of such a combination is shown in the figure 7. One can immediately notice a very significant property. It is that the graph is now a complex structure consisting of two sub-graphs: a bar graph and a colored line graph. When the main 14 metrics we first presented, none of the structures had more than one metric encapsulated within the widget. Now widgets containing these complex forms with not only two but potentially more sub-graphs indicate a special correlation between them. Let us take a closer look at the figure 7 and determine why this graph was chosen to be present to the manager who is responsible for monitoring the dashboard.

Fig. 7. Correlation between Story Points and Avg. Cycle Time

To start off, it is important to note that this graph spans the time period of exactly 14 days, the duration of one Sprint. Therefore, X axis represents the corresponding stage of the Sprint. Next, the Y axis is responsible for showing the amount of Story Points that a team has managed to obtain on the specific day. It also shows the Average Cycle Time of a task in the sprint (measured in minutes).

From the chart it is also easy to notice that the cycle time is fairly low, indicating that the tasks are well segmented and there are a lot of them, as the amount of story points is not low. However, the reason this graph is particularly interesting is that the Average Cycle Time graph is trending downwards. That means that it took more time for developers and designers to complete the tasks at the beginning of the Sprint rather than on the end of the Sprint.

There are numerous possible reasons for that, but the manager can be certain that his/her team did not try to hurry up at the Sprint end to finish most of the tasks because the number of Story Points is evenly distributed throughout the week. The manager might be misled to thinking that that is the case only by looking at the Average Cycle Time. The combination of the two graphs frees him/her from that suspicion, and now he/she can focus on other, less critical reasons why that is the case.

Here are some of the possible reasons:

- the scope/size of the tasks is not uniformly distributed throughout the Sprint
- the team decided to first finish the complex tasks before moving on to easier ones

As one can see, neither of the two reasons are as critical as the team doing most of the work at the Sprint's end. Thereafter, the widget managed to effectively show that information without the manager wasting his/her precious time on tracking down the issue that did not exist in the first place.

This is an example of an effective combination of the two metrics that is far more useful when analyzed together rather than separately. It was determined to be important due to the fact that the trend was noticed in the Average Daily Cycle Time, and that such regularity did not cause any fluctuation

in the number of Daily Story Points in the Sprint. Rather than looking at the raw data and realizing that his/her team is trying to do most of the work at the end of the Sprint, manager focused his/her attention to other problems and ways the workflow can be improved. The dashboard served as a helper tool to get the job done by enhancing workers' expertise and not getting in the way of already productive existing workflow pipelines.

IX. CONCLUSION

We have shown our previous work in the field of Software Metrics and suggested a possible way of improving it, as well as to give rise to the importance of designing and developing compelling dashboards. We have previously decided on 14 key metrics for the software development process. Now, by continuous improvement and natural selection of the most relevant agents the dashboard should be able to achieve the dynamic equilibrium using non-linear dynamics. The dashboard would also be able to combine several metrics that have a higher chance of indicating a possible flaw in the workflow. That does not necessarily mean that it is going to achieve the optimal state right away, nor does it mean that just by applying the techniques presented here will we truly solve the problem of inadequate dashboards in general. But rather this all means that our dashboard will try to get there over time, by facilitating adaptation, collaboration and expertise while avoiding "chaos".

X. ACKNOWLEDGMENTS

The work presented in this paper was supported by the grant of Russian Science Foundation №19-19-00623.

REFERENCES

- [1] R. Brath and M. Peters. Dashboard design: Why design is important. *DM Direct*, 85, 2004/10.
- [2] I. D. Coman, A. Sillitti, and G. Succi. A case-study on using an automated in-process software engineering measurement and analysis system in an industrial environment. In *Proceedings of the 31st International Conference on Software Engineering, ICSE '09*, pages 89–99, Washington, DC, USA, 2009. IEEE Computer Society.
- [3] W. W. Eckerson. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. Wiley, 2010.

- [4] S. Few. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media, 2006.
- [5] D. Holten, R. Vliegen, and J. van Wijk. Visual realism for the visualization of software metrics. In *3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis*. IEEE, 2005.
- [6] V. Ivanov, V. Pischulin, A. Rogers, G. Succi, J. Yi, and V. Zorin. Design and validation of precooked developer dashboards. In *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018*. ACM, 2018.
- [7] V. Ivanov, A. Rogers, G. Succi, J. Yi, and V. Zorin. Precooked developer dashboards: What to show and how to use. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, ICSE '18*, pages 402–403, New York, NY, USA, 2018. ACM.
- [8] A. Janes, A. Sillitti, and G. Succi. Effective dashboard design. *Cutter IT Journal*, 26:17–24, 01 2013.
- [9] A. Janes and G. Succi. *Lean Software Development in Action*. Springer Berlin Heidelberg, 2014.
- [10] Y. Lajoie, N. Teasdale, C. Bard, and M. Fleury. Attentional demands for static and dynamic equilibrium. *Experimental brain research*, 97(1):139–144, 1993.
- [11] L. López, S. Martínez-Fernández, C. Gómez, M. Choraś, R. Kozik, L. Guzmán, A. M. Vollmer, X. Franch, and A. Jedlitschka. Q-rapids tool prototype: Supporting decision-makers in managing quality in rapid software development. In *Lecture Notes in Business Information Processing*, pages 200–208. Springer International Publishing, 2018.
- [12] V. Pishulin. to clarify later. 2018.
- [13] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher. What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2018.
- [14] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann. Dashboards for continuous monitoring of quality for software product under development. In *Relating System Quality and Software Architecture*, pages 209–229. Elsevier, 2014.
- [15] J. M. T. Thompson, M. Thompson, and H. B. Stewart. *Nonlinear dynamics and chaos*. John Wiley & Sons, 2002.
- [16] A. Valiullin. Designing a dashboarding system for visualization of non-invasively collected metrics. 2018.
- [17] O. M. Yigitbasioglu and O. Velcu. A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 13(1):41–59, mar 2012.
- [18] V. Zorin. to clarify later. 2017.

An Empirical Analysis on Effectiveness of Source Code Metrics for Aging Related Bug Prediction

1st Chinmay Hota

BITS Pilani Hyderabad Campus, India
f20160069@hyderabad.bits-pilani.ac.in

2nd Lov Kumar

BITS Pilani Hyderabad Campus, India
lovkumar@hyderabad.bits-pilani.ac.in

3rd Lalita Bhanu Murthy Neti

BITS Pilani Hyderabad Campus, India
bhanu@hyderabad.bits-pilani.ac.in

Abstract—Reliability of any software plays a very important role to assess the quality of the software systems. Reliability analysis and evaluation during early phases of software development life cycle (SDLC) significantly help developer and analysts in the proper allocation of limited resources during testing and maintenance process. The goal of this work is to develop one model using the measurement of the internal structure of the software system i.e., source code metrics for predicting the reason of failure in software during continuously running for a certain time i.e., software aging. Aging-Related Bugs are related with failure during the execution of the software, that leads to degradation in quality, system crashing, misuse of resources, etc.. In this paper, seven different sets of software metrics, seven model training algorithms, one data sampling technique have been empirically investigated and evaluated for predicting aging-related bugs classes. The trained aging-related bugs prediction models are validated using 5-fold cross-validation techniques. The final observation of this experiment work is assessed over seven open source application software systems. The high-value of performance parameters confirm the predicting capability of data sampling, sets of metrics, and training algorithms to predict aging-related bugs classes.

Index Terms—Aging, Software Engineering, Source Code Metrics, Machine Learning, Feature selection.

I. INTRODUCTION

Today's scenario, the majority of the software corporation has it in mind to apply the Object-Oriented (OO) tools for construction the contemporary software system due to its efficient blueprint characteristic such as reusability (extending the code use again) and reducing vulnerability, which facilitates more rapid product development [1] [2]. With the help of these features, we can achieve the OO properties like-Inheritance, data abstraction, polymorphism, etc.. Software development or solutions of business requirement using OO programming expect high-quality such reliability, maintainability, reusability, portability, etc.. Measuring quality of software and understanding their relationship with software metrics can help tester, architect, and developer control and estimate quality of software by analyzing the source code. The experimental study presented in the paper is motivated by the need to correlate and investigate aging-related bugs work and source code metrics. Aging-related bugs are related with failure during the execution of the software, that leads to degradation in quality, system crashing, misuse of resources, etc.. The effort and cost of fixing aging-related bugs increase exponentially

if these bugs are not encountered before the deployment of the software [3] [4]. The aging-related bugs prediction models can be applied in the early phase of SDLC. Further, their use can improve the quality of the software as well as reduces the maintenance and testing cost. The experimental study presented in this paper is to develop one model for automatic detection of aging-related bugs work. In particular, our motivation is to investigate the application of different feature selection techniques, classification techniques, and data imbalanced techniques for the task of predicting run time failure i.e., aging-related bugs. Thus, the following research questions are addressed in this work:

- **RQ1: What is the capability of different classification techniques to train the models for predicting aging-related bugs?**

In this research question, the 5-fold cross-validation technique has been considered to validate the models developed using seven different classification techniques. The capability and significance of all considered classification techniques are assessed based on the seven different datasets using rank-sum test and AUC values.

- **RQ2: What is the effect of different sets of metrics for training the models using classification techniques?**

In this research question, seven different sets of features extracted from source code metrics i.e., software metrics are used as an input to train the aging-related bug prediction models. The capability and significance of these sets of features are computed and compared using AUC values on the seven different datasets.

- **RQ3: What is the capability of data sampling technique to train the models for predicting aging-related bugs?**

In this research question, aging-related bug prediction models are trained using balance data and imbalance data. The capability and significance of the data sampling techniques are computed and compared using AUC values on the seven different datasets.

II. RELATED WORK

Grottke et al. [3] investigated that aging-related bugs caused software failures have been observed in numerous events and operating conditions, where it has caused major losses with respect to hazards, the human being as well as money.

Junjun Zheng et al. [4] proposed six number of rejuvenation rules, which are particularly meet with the time-

based constraints. They demonstrated a systematic study to measure those six software rejuvenation rules numerically and illustrated that the planned rejuvenation rules often termed as wait-time rules are better to the other Markovian arrival processes (MAPs). In practice eliminating entire bugs is even impractical that eventually makes a system probable of being aged after some time. Therefore, efforts must have been made to detect aging-related bugs from software systems to enable swift aging-related bugs detection and to avoid future system-failures. Both bug identification and prediction are the challenging tasks of the developer.

Padhy et al. [5] developed a framework called re-usability optimization based aging-related bugs prediction system, which will estimate the re-usability level. Their focus was on re-usability risk management, where they argued that excessive reused system enters into aging-related bugs. Rejuvenation is one of the practical approaches to avoid failures, which causes the software aging-related bugs. Generally, those aging-related bugs are hard to eliminate completely in the software development as well as testing phase. These types of bugs are often known as a memory leak. The process of handling the software aging-related bugs play the key role and proactive management is required to handle such kind issues.

III. RESEARCH BACKGROUND

In this section, the details regarding dataset, sampling technique, techniques used for selecting significant uncorrelated predictors, ELM with three kernels have been summarized.

A. Experimental Dataset

In this experiment, the empirical results are computed over seven large-scale software applications. These software are mainly used for the development of mission-critical, distributed applications, near real-time applications. This software come in the category of large-scale software used in real-world applications, including safety-critical and business contexts. The data has been collected using tera-PROMISE Repository¹. The tera-PROMISE website is a very well-known data repository consisting of various research datasets related to software engineering on defects, code analysis, effort, bugs, maintainability, refactoring, and test generation [6]. The primary objective of using these datasets to replicated our experiment finding, so that these experimental findings are used by other researchers for benchmarking and comparison. In this paper, experiments are conducting on a dataset containing information of aging-related bugs. The dataset contains various types of metrics such as program size related metrics, Halstead metrics, McCabe's Cyclomatic complexity, and aging-related metrics [7]. The characteristics of these datasets are given in Table I. The data has been collected for seven different projects (Four projects: Linux sub-systems and three projects: MySQL sub-systems). From Table I, it can be clearly seen that Linux driver net is the largest project containing 2292 classes and Linux ext3 is the smallest project

containing 29 classes among all projects. It is also observed from Table I that are considered data are highly imbalanced because % of aging-related classes varies from 0.39% to 17.24%.

B. Software Metrics

The features extracted from source code called software metrics used to measure the size and complexity of source code. These extracted features are considered as an input to train the models for predicting aging-related bugs. In this work, we have used 49 metrics related to program size, 18 metrics related to McCabe's Cyclomatic complexity, 9 metrics related to Halstead metrics, and 6 metrics related to aging-related bugs as shown in Table II for measuring size and complexity of source code. The abbreviation and details explanation of these metrics sets are explained with practical applications in [8] [9] [10]².

C. Feature Selection Techniques

The next step of this work is to apply feature selection techniques on feature extracted from source code metrics for finding the right sets of uncorrelated significant features. In this step, some important features are selected from all considered features and these selected features are considered by training algorithms to train the models for predicting aging-related bugs. In this paper, we have used univariate rank-sum test and logistic regression to remove insignificant predictor for aging-related bugs. Ramsum test is used to test null hypothesis i.e., no statistical significance between the values of a feature of the class having aging-related bugs and the values of a feature of the class having non-aging-related bugs. The considered null hypothesis has been tested at 0.05 p-value i.e., $p\text{-value} \leq 0.05$: hypothesis rejected means metrics is relevant for aging-related bugs prediction and $p\text{-value} > 0.05$: hypothesis accepted means metrics is irrelevant for aging-related bugs prediction. After removing all insignificant predictors, we are using cross-correlation analysis to remove highly correlated features and finally select significant uncorrelated features. The primary objective of the above step is to select independent significant features having a high correlation with aging-related bugs.

D. Classification Algorithms:

In this research, we are using extreme learning machine (ELM) algorithm with different kernels functions to train the aging-related bugs prediction models by considering features extracted from source code metrics as input. ELM is based on the concept of randomly generated hidden neurons with an objective to reduce the learning time of the models [11]. ELM algorithm overcomes various problems such as improper learning rate and overfitting, a problem due to local minima, backpropagation. Finally, the trained models using ELM for predicting aging-related bugs are compared with widely used classifiers using AUC value and accuracy.

¹<http://openscience.us/repo/>

²https://github.com/sealuzh/user_quality/wiki/Code-Quality-Metrics

TABLE I: Experimental Dataset Description

ID	Name	Total classes	non-aging classes	% non-aging classes	aging classes	% aging classes
P1	Linux driver net	2292	2283	99.61	9	0.39
P2	Linux ext3	29	24	82.76	5	17.24
P3	Linux driver scsi	962	958	99.58	4	0.42
P4	Linux ipv4	117	115	98.29	2	1.71
P5	MySQL optimizer	36	33	91.67	3	8.33
P6	MySQL replication	32	28	87.5	4	12.5
P7	MySQL innodb	402	370	92.04	32	7.96

TABLE II: Software Metrics

Types of Metrics Sets	Metrics List
Program size metrics	AltAvgLineCode, AltAvgLineComment, AltCountLineCode, AvgCyclomatic, AvgCyclomaticModified, AvgCyclomaticStrict, AvgEssential, AvgLineComment, CountClassCoupled, CountDeclClassMethod, CountDeclClassVariable, CountDeclFunction, CountDeclInstanceMethod, CountDeclInstanceVariablePrivate, CountDeclMethod, CountDeclMethodPrivate, CountDeclMethodProtected, CountDeclMethodPublic, CountInput, CountLine, CountLineCode, CountLineCodeDecl, CountLineCodeExe, CountLineComment, CountLineInactive, CountLinePreprocessor, CountSemicolon, CountStmt, CountStmtDecl, CountStmtEmpty, AltAvgLineBlank, AltCountLineBlank, AltCountLineComment, AvgLine, AvgLineBlank, AvgLineCode, CountClassBase, CountClassDerived, CountDeclClass, CountDeclInstanceVariable, CountDeclInstanceVariableProtected, CountDeclInstanceVariablePublic, CountDeclMethodAll, CountDeclMethodConst, CountDeclMethodFriend, CountLineBlank, CountOutput, CountPath, CountStmtExe
MCC metrics	CyclomaticStrict, Essential, Knots, MaxCyclomatic, MaxCyclomaticModified, MaxEssentialKnots, MaxInheritanceTree, MaxNesting, MinEssentialKnots, SumCyclomatic, SumCyclomaticModified, SumCyclomaticStrict, Cyclomatic, CyclomaticModified, MaxCyclomaticStrict, PercentLackOfCohesion, RatioCommentToCode, SumEssential
Halstead metrics	Number of distinct operands, Program length, Difficulty, Number of distinct operators, Effort, Total number of operands, Volume, Program vocabulary, Total number of operators
Aging-related bugs metrics	DerefUse, UniqueDerefSet, DeallocOps, AllocOps, DerefSet, UniqueDerefUse

E. Data Sampling Technique to handle Imbalanced Data:

Models trained using imbalanced data are the common problem in many domains related to data mining and machine learning such as medical diagnosis, defect prediction, fraud detection etc.. A data is called imbalanced if it has not equal distributions of classes. These types of data sets may be affected the actual performance of models. It can be seen from Table I that the percentage of aging-related bugs classes present is considered datasets varies from 0.39% to 17.24 %. Since, the percentage of aging-related bugs classes are

not equal to percentage of non-aging-related bugs classes, so models developed using these datasets suffered from class imbalance problem. So, in this work, upscale sampling technique has been considered to deal with class imbalanced class problem. The performance of the developed models using upscale data are computed using accuracy and AUC values and compared with original data.

Fig. 1: Framework of proposed work

F. Performance Parameter

In order to measure the quality of aging-related bugs prediction models, various performance metrics need to be examined which indicated the effectiveness of the trained models. In this work, we have considered two different performance metrics such as Accuracy (%) and area under curve value (AUC) for evaluating the quality of aging-related bugs prediction models. A high value of these metrics indicates that the built prediction model is good.

IV. RESEARCH FRAMEWORK AND EXPERIMENTAL RESULTS

This section presents the experimental framework used for developing aging-related bugs prediction models using different sets of source code metrics. Figure 1 shows the empirical framework consisting various steps for development of aging-related bugs prediction model. From Figure 1, it can be seen that five major steps are used in the proposed framework. The five steps are: (1) Data balancing using upscale sampling technique (2) Categorization of all considered metrics into different sets and also selection of significant and uncorrelated right sets of metrics (3) Data normalization using Min-Max normalization technique (4) Predictive model trained using different classification techniques (5) Significance and capability evaluation using different performance parameters. The above mentioned steps are applied on different packages of two long-living, complex, and large software systems i.e., MySQL and Linux kernel. In this work, eighty two different metrics such as Halstead metrics, McCabes cyclomatic complexity, size related metrics, and aging-related bugs metrics have been considered as predictors for aging-related bugs prediction models. From Table I, we can observed that the considered datasets are suffering from class imbalance problem. So, we have considered upscale sampling technique to address the issue of class imbalance problem. The second step consists of Categorization of all considered metrics into different sets i.e., Halstead metrics (HM), McCabes cyclomatic complexity (MCC), size related metrics (PSM), aging-related bugs metrics (AGM), and selection of right set of significant predictor (SGM) and uncorrelated significant predictor (CCM) using ranksum test and cross correlation analysis. Then, Min-Max Normalization has applied before training the model to normalize the all selected features in the same range of 0 to 1. Further, seven different classification techniques have been used to train the aging-related bugs prediction model. These models are validated using 5-fold cross-validation approach. Finally, accuracy, and AUC values are computed for each model to evaluate capability and significance of above considered classification techniques, data sampling technique, and different sets of metrics.

A. Feature Selection Techniques:

In this section, we have summarized the results obtained after ranksum test on original data and upscale sampling data for different projects. In this study, file having at least one aging-related bugs considered as aging-related bugs

classes else non-aging-related bugs classes. Table III provides the relevant metrics related program size metrics after ranksum test for two projects on without sampling data i.e., original data and upscale data. Similarly, the relevant features for other metrics sets are computed using ranksum test. In Table III, only 0 and 1 are used to represent relevant features and irrelevant metrics i.e., cell containing 1 denotes the relevant metrics for project using original data or upscale data and cell containing 0 denotes the irrelevant metrics for project using original data or upscale data. It can be observed from Table III that AltAvgLineBlank is relevant for all cases except scsi, ext3, and ipv4 projects. Similarly, It can also observed from Table III that CountClassBase, CountClassCoupled, CountClassDerived, CountDeclClassMethod, CountDeclClassVariable etc. are irrelevant for aging-related bugs prediction i.e., these metrics are not considered as input to train the aging-related bugs prediction models.

B. Results and analysis

We have considered seven sets of source code metrics as input to train the model using seven different classification techniques to predict aging-related bugs. Five fold cross validation method has been used to validate the trained models and performance of these trained models are express in terms of area under curve value (AUC) and accuracy. The value of Accuracy and AUC of trained model using different techniques for Linux net project on each fold shown in Table IV. Similarly, the Accuracy and AUC values for other projects are computed. It can be seen from Table IV that:

- Aging-related bugs prediction models trained using Upscale sampling data have relatively better AUC values as compare to original data.
- Aging-related bugs prediction models trained using logistic regression and decision tree have relatively better AUC values as compare to other techniques.
- Aging-related bugs prediction models trained using ELM with linear kernel function have relatively better AUC values as compare to other kernel functions.

V. COMPARISON

RQ1: RQ1: what is the overall predictive capability of model trained using different classification techniques to predict aging-related bugs? In this research question, 5-fold cross validation technique has been considered to validate the models developed using seven different classification techniques. The significance and capability of all considered classification techniques are assessed based on the seven different datasets using Boxplots, Descriptive Statistics, and Statistical tests analysis.

Boxplots and Descriptive Statistics: Classification Techniques In this paper, boxplot diagram has been considered for visual comparison showing distribution, variability, outliers, degree of dispersion of different classification techniques as depicted in Figure 2. The first, second, and third sub-figures of Figure 2 shows the boxplot diagram of AUC value for classification techniques trained using original data, upscale

TABLE III: Ranksum Test for Program Size Metrics

	Original Data							Upscale Sampling						
	Linux				MySQL			Linux				MySQL		
	net	scsi	ext3	ipv4	innodb	optimizer	replication	net	scsi	ext3	ipv4	innodb	optimizer	replication
AltAvgLineBlank	1	0	0	0	1	1	1	1	1	1	1	1	1	1
AltAvgLineCode	1	0	0	0	1	1	0	1	1	1	1	1	1	1
AltAvgLineComment	1	1	0	0	1	1	1	1	1	1	1	1	1	1
AltCountLineBlank	1	0	1	0	1	1	1	1	1	1	1	1	1	1
AltCountLineCode	1	0	1	0	1	1	1	1	1	1	1	1	1	1
AltCountLineComment	1	0	1	0	1	1	1	1	1	1	1	1	1	1
AvgCyclomatic	1	1	0	0	1	1	0	1	1	1	0	1	1	1
AvgCyclomaticModified	1	1	0	0	1	1	1	1	1	1	0	1	1	1
AvgCyclomaticStrict	1	1	0	0	1	1	0	1	1	1	0	1	1	1
AvgEssential	0	1	0	0	1	1	0	1	1	1	0	1	1	1
AvgLine	1	1	0	0	1	1	0	1	1	1	1	1	1	1
AvgLineBlank	1	0	0	0	1	1	1	1	1	1	1	1	1	1
AvgLineCode	1	0	0	0	1	1	0	1	1	1	1	1	1	1
AvgLineComment	1	1	0	0	1	1	1	1	1	1	1	1	1	1
CountClassBase	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountClassCoupled	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountClassDerived	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclClass	0	0	0	0	1	0	0	0	0	0	0	1	1	1
CountDeclClassMethod	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclClassVariable	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclFunction	1	0	1	0	1	0	1	1	1	1	1	1	0	1
CountDeclInstanceMethod	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclInstanceVariable	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclInstanceVariablePrivate	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclInstanceVariableProtected	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclInstanceVariablePublic	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethod	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodAll	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodConst	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodFriend	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodPrivate	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodProtected	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountDeclMethodPublic	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountInput	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountLine	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountLineBlank	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountLineCode	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountLineCodeDecl	1	0	1	0	1	0	1	1	1	1	1	1	1	1
CountLineCodeExe	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountLineComment	1	1	1	0	1	1	1	1	1	1	1	1	1	1
CountLineInactive	1	0	0	0	1	0	1	1	1	1	1	1	1	1
CountLinePreprocessor	1	0	0	0	1	0	1	1	1	1	1	1	1	1
CountOutput	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountPath	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CountSemicolon	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountStmt	1	0	1	0	1	1	1	1	1	1	1	1	1	1
CountStmtDecl	1	0	1	0	1	0	1	1	1	1	1	1	1	1
CountStmtEmpty	1	1	0	1	1	1	1	1	1	1	1	1	1	1
CountStmtExe	1	1	1	0	1	1	1	1	1	1	1	1	1	1

sampling data, and overall. The median value of AUC value for each classification technique has been presented using red line. The descriptive statistics i.e., **Min**, **25%**, **Mean**, **Median**, **75%**, **Max** for each box are presented in Table V.

It can be seen from Table V and Figure 2 that aging-related bugs prediction models trained using decision tree (DT) have relatively better AUC values as compare to other techniques. It can be also seen from Table V and Figure 2 that ELM with

TABLE IV: Accuracy and AUC values for Linux net and Scsi

Without Sampling (net)														
	Accuracy							AUC						
	LR	PR	LOR	DT	ELMLIN	ELMPY	ELMRBF	LR	PR	LOR	DT	ELMLIN	ELMPY	ELMRBF
Fold1	98.69	94.32	98.25	99.78	99.78	93.45	99.78	0.49	0.47	0.49	0.50	0.50	0.47	0.50
Fold2	98.47	95.64	98.47	99.56	99.35	76.47	99.56	0.49	0.73	0.49	0.50	0.50	0.63	0.50
Fold3	97.82	89.11	98.91	98.69	98.91	88.24	99.56	0.74	0.70	0.50	0.50	0.50	0.69	0.50
Fold4	98.69	93.23	98.47	99.56	99.56	89.52	99.56	0.99	0.97	0.74	0.50	0.50	0.45	0.50
Fold5	97.16	92.14	97.82	99.13	99.13	96.94	99.56	0.74	0.96	0.49	0.50	0.50	0.49	0.50
Upscale Sampling (net)														
Fold1	50.00	50.00	98.47	99.78	97.16	64.11	50.00	0.50	0.50	0.98	1.00	0.97	0.64	0.50
Fold2	49.95	49.95	98.47	99.56	95.84	54.87	49.95	0.50	0.50	0.98	1.00	0.96	0.55	0.50
Fold3	50.00	50.00	99.02	99.34	96.72	67.72	50.00	0.50	0.50	0.99	0.99	0.97	0.68	0.50
Fold4	50.00	50.00	98.57	99.23	95.18	69.30	50.00	0.50	0.50	0.99	0.99	0.95	0.69	0.50
Fold5	50.05	50.05	98.69	99.67	95.51	68.02	50.05	0.50	0.50	0.99	1.00	0.96	0.68	0.50
Without Sampling (scsi)														
	LR	PR	LOR	DT	ELMLIN	ELMPY	ELMRBF	LR	PR	LOR	DT	ELMLIN	ELMPY	ELMRBF
Fold1	95.31	76.04	97.40	100.00	99.48	82.81	100.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Fold2	92.75	76.68	97.93	98.96	98.96	37.31	99.48	0.47	0.88	0.49	0.99	0.50	0.68	0.50
Fold3	93.75	86.98	97.40	99.48	98.44	86.46	99.48	0.47	0.44	0.49	0.50	0.49	0.93	0.50
Fold4	95.85	79.79	98.96	99.48	99.48	95.85	99.48	0.48	0.40	0.50	0.50	0.50	0.48	0.50
Fold5	96.35	93.23	96.88	98.96	97.92	76.04	99.48	0.48	0.49	0.50	0.50	0.49	0.88	0.50
Upscale Sampling (scsi)														
Fold1	50.00	50.00	94.27	99.22	93.49	56.51	50.00	0.50	0.50	0.94	0.99	0.93	0.57	0.50
Fold2	50.00	50.00	92.93	99.74	95.03	63.35	50.00	0.50	0.50	0.93	1.00	0.95	0.63	0.50
Fold3	50.00	50.00	95.83	99.48	93.49	75.26	50.00	0.50	0.50	0.96	0.99	0.93	0.75	0.50
Fold4	49.87	49.87	93.99	98.69	93.47	53.26	49.87	0.50	0.50	0.94	0.99	0.93	0.53	0.50
Fold5	50.13	50.13	96.34	98.69	92.43	68.41	50.13	0.50	0.50	0.96	0.99	0.92	0.68	0.50

liner kernel on upscale sampling data and over all provided good results as compared to other polynomial kernel and RBF kernels.

Comparison of different sampling techniques using Statistical tests: In this work, we have also considered pairwise rank sum test with Bonferroni correction to compare the performance of developed aging-related bugs prediction models using different classification techniques. We have considered seven sets of metrics, and two different datasets for each seven project. Hence, each classification technique has 490 data points or AUC values [7 sets of features * (1 original data + 1 upscale sampling data) * 7 projects * 5 folds]. The null hypothesis of the rank sum test is that *aging-related*

bugs prediction models developed using different classification techniques are not significantly different. The rank sum test with Bonferroni correction adjust the threshold p-value at 0.05

. Since, we have considered seven number of different pairs different classification techniques, so the number of different pairs is $7c_2 = 7 * 6/2 = 21$. Hence, null hypothesis is accepted only if the p-value is less then $\frac{0.05}{21} = 0.0024$. The results of rank sum test with Bonferroni correction for different classification techniques are summarized in Table VI. We observe from Table VI that the p-value of 13 out of 21 pairs are less than 0.0024 i.e., hypothesis accepted means model trained using different classification techniques are

Fig. 2: Boxplots for AUC for Classification Techniques

Fig. 3: Boxplots for F-Measure

TABLE V: Statistical Measures on AUC: Classification Techniques

Original Data						
	Min	Max	Mean	Median	25%	75%
LR	0.20	0.83	0.45	0.48	0.34	0.50
PR	0.20	0.82	0.43	0.44	0.30	0.50
LOR	0.20	1.00	0.54	0.50	0.40	0.63
DT	0.20	1.00	0.55	0.50	0.40	0.65
ELMLIN	0.18	0.96	0.49	0.50	0.35	0.59
ELMPY	0.13	1.00	0.50	0.50	0.39	0.59
ELMRBF	0.20	0.78	0.41	0.40	0.30	0.50
Upscale Sampling						
	Min	Max	Mean	Median	25%	75%
LR	0.50	0.81	0.53	0.50	0.50	0.50
PR	0.50	0.50	0.50	0.50	0.50	0.50
LOR	0.56	1.00	0.88	0.92	0.80	0.99
DT	0.80	1.00	0.95	0.96	0.92	1.00
ELMLIN	0.44	1.00	0.81	0.84	0.69	0.93
ELMPY	0.45	1.00	0.73	0.69	0.58	0.92
ELMRBF	0.50	0.98	0.63	0.50	0.50	0.80
Overall						
	Min	Max	Mean	Median	25%	75%
LR	0.20	0.83	0.49	0.50	0.48	0.50
PR	0.20	0.82	0.47	0.50	0.44	0.50
LOR	0.20	1.00	0.71	0.77	0.50	0.93
DT	0.20	1.00	0.75	0.85	0.50	0.99
ELMLIN	0.18	1.00	0.65	0.67	0.46	0.88
ELMPY	0.13	1.00	0.62	0.59	0.49	0.73
ELMRBF	0.20	0.98	0.52	0.50	0.40	0.50

significantly different.

RQ2: what is the overall predictive capability of model developed using different sets of metrics as input to predict aging-related bugs?

In this paper, seven different sets of metrics are considered as input to developed a model for predicting aging-related bugs. The significance and capability of all considered sets of metrics are assessed based on the seven different datasets using Boxplots, Descriptive Statistics, and Statistical tests

TABLE VI: P-value: Classification Techniques

	LR	PR	LOR	DT	ELMLIN	ELMPY	ELMRBF
LR	1.00	0.17	0.00	0.00	0.00	0.00	0.98
PR	0.17	1.00	0.00	0.00	0.00	0.00	0.36
LOR	0.00	0.00	1.00	0.23	0.08	0.01	0.00
DT	0.00	0.00	0.23	1.00	0.00	0.00	0.00
ELMLIN	0.00	0.00	0.08	0.00	1.00	0.23	0.00
ELMPY	0.00	0.00	0.01	0.00	0.23	1.00	0.00
ELMRBF	0.98	0.36	0.00	0.00	0.00	0.00	1.00

analysis.

Comparison of Classification Techniques using Boxplots and Descriptive Statistics: In this paper, boxplot diagram has been considered for visual comparison showing distribution, variability, outliers, degree of dispersion of different sets of metrics as depicted in Figure 3. The first, second, and third sub-figures of Figure 3 shows the boxplot diagram of AUC value for classification techniques trained using original data, upscale sampling data, and overall. The median value of AUC value for each classification technique has been presented using red line. The descriptive statistics i.e., **Min, 25%, Mean, Median, 75%, Max** for each box are presented in Table VII. It can be seen from Table VII and Figure 3 that aging-related bugs prediction models developed using aging-related bugs (AGM) metrics as input have relatively better AUC values as compare to other sets of metrics. It can be also seen from Table VII and Figure 3 that significant sets of features (SGM) provided results as compared to other sets of metrics except AGM.

Comparison of different sampling techniques using Statistical tests: In this work, we have also considered pairwise rank sum test with Bonferroni correction to compare the performance of developed aging-related bugs prediction models using different sets of metrics. We have considered seven different classification techniques, and two different datasets for each seven project. Hence, each each set of metrics has 490 data points or AUC values [7 classification techniques * (1 original data + 1 upscale sampling data) * 7 projects * 5 folds]. The null hypothesis of the rank sum test is

TABLE VII: Statistical Measures on AUC: Sets of Metrics.

Original Data						
	Min	Max	Mean	Median	25%	75%
AM	0.18	0.76	0.45	0.48	0.34	0.55
PSM	0.13	0.85	0.46	0.50	0.37	0.57
MCC	0.20	0.73	0.44	0.48	0.38	0.53
HM	0.20	0.83	0.46	0.50	0.38	0.54
AGM	0.20	1.00	0.61	0.50	0.40	0.81
SGM	0.20	1.00	0.52	0.50	0.38	0.67
CCM	0.20	0.83	0.42	0.41	0.30	0.50
Upscale Sampling						
	Min	Max	Mean	Median	25%	75%
AM	0.45	1.00	0.67	0.61	0.50	0.88
PSM	0.50	1.00	0.69	0.64	0.50	0.89
MCC	0.50	0.99	0.75	0.80	0.50	0.92
HM	0.44	0.99	0.64	0.56	0.50	0.80
AGM	0.50	1.00	0.83	0.98	0.50	1.00
SGM	0.49	1.00	0.70	0.61	0.50	1.00
CCM	0.50	1.00	0.74	0.80	0.50	0.89
Overall						
	Min	Max	Mean	Median	25%	75%
AM	0.18	1.00	0.56	0.50	0.45	0.64
PSM	0.13	1.00	0.58	0.50	0.50	0.73
MCC	0.20	0.99	0.60	0.50	0.48	0.80
HM	0.20	0.99	0.55	0.50	0.47	0.68
AGM	0.20	1.00	0.72	0.80	0.50	1.00
SGM	0.20	1.00	0.61	0.50	0.49	0.80
CCM	0.20	1.00	0.58	0.50	0.41	0.81

that *aging-related bugs prediction models developed using different sets of metrics are not significantly different*. The rank sum test with Bonferroni correction adjust the threshold p-value at $\frac{0.05}{\text{number of different pairs}}$. Since, we have considered seven sets of metrics, so the number of different pairs is $7c_2 = 7 * 6/2 = 21$. Hence, null hypothesis is accepted only if the p-value is less than $\frac{0.05}{21} = 0.0024$. The results of rank sum test with Bonferroni correction for sets of metrics are summarized in Table VIII. We observe from Table VIII that the p-value of the models trained using aging-related bugs metrics are less than 0.0024 i.e., hypothesis accepted means model trained using aging-related bugs metrics and other sets of metrics are significantly different.

TABLE VIII: P-value: Sets of Metrics

	AM	PSM	MCC	HM	AGM	SGM	CCM
AM	1.00	0.67	0.41	0.87	0.00	0.17	0.73
PSM	0.67	1.00	0.65	0.52	0.00	0.33	0.96
MCC	0.41	0.65	1.00	0.26	0.00	0.51	0.59
HM	0.87	0.52	0.26	1.00	0.00	0.15	0.59
AGM	0.00	0.00	0.00	0.00	1.00	0.01	0.00
SGM	0.17	0.33	0.51	0.15	0.01	1.00	0.31
CCM	0.73	0.96	0.59	0.59	0.00	0.31	1.00

RQ3: what is the overall predictive capability of model developed using balance and imbalance data to predict aging-related bugs?

Fig. 4: Boxplots for AUC

In this work, aging-related bugs prediction models are trained using balance data and imbalance data. The significance and capability of the data sampling used for balancing data is assessed based on the seven different datasets using ranksum test and AUC values. .

Comparison of Classification Techniques using Boxplots and Descriptive Statistics: In this paper, boxplot diagram has been considered for visual comparison showing distribution, variability, outliers, degree of dispersion of original data and upscale sampling data as depicted in Figure 4. The median value of AUC value for each data set has been presented using red line. The descriptive statistics i.e., **Min**, **25%**, **Mean**, **Median**, **75%**, **Max** for each box are presented in Table IX. It can be seen from Table IX and Figure 4 that aging-related bugs prediction models developed using sampling data i.e., balance data have relatively better AUC values as compare to original data.

TABLE IX: Statistical Measures on AUC: Original data and Upscale Data

	Min	Max	Mean	Median	25%	75%
WTSMP	0.13	1.00	0.48	0.50	0.38	0.57
SMP	0.44	1.00	0.72	0.71	0.50	0.92

Comparison of different sampling techniques using Statistical tests: In this work, we have also considered pairwise rank sum test with Bonferroni correction to compare the performance of developed aging-related bugs prediction models using original data and sampled data. We have considered seven different classification techniques, seven sets of metrics, and seven project. Hence, each data set has 1715 data points or AUC values [7 classification techniques * 7 sets of features * 7 projects * 5 folds]. The null hypothesis of the rank sum test is that *aging-related bugs prediction models developed using original data and sampled data are not significantly different*. The rank sum test with Bonferroni correction adjust the threshold p-value at $\frac{0.05}{\text{number of different pairs}}$. Since, we have considered two datasets, so the number of different pairs

is $2c_2 = 1$. Hence, null hypothesis is accepted only if the p-value is less than $\frac{0.05}{1} = 0.05$. The results of rank sum test with Bonferroni correction for sets of metrics are summarized in Table X. We observe from Table X that the p-value of the models trained using original data and sampled data are significantly different

TABLE X: P-value: Original data and Upscale Data

	WTSMP	SMP
WTSMP	1.00	0.00
SMP	0.00	1.00

VI. CONCLUSION

The primary goal of this research is to find the effect of internal structure of software on aging-related bugs. In this experiment, we also empirically evaluated, analyzed, and compared the performance of developed aging-related bugs prediction models using different classification techniques, sets of metrics, selected set of metrics, and data sampling technique. The major conclusions of this study are summarized as follows.

- The aging-related bugs prediction models trained using decision tree (DT) have relatively better AUC values as compare to other techniques.
- The aging-related bugs prediction models developed using aging-related bugs (AGM) metrics as input have relatively better AUC values as compare to other sets of metrics.
- Upscale sampling technique provides relatively better performance in detecting aging-related bugs compared to the models built by original data.

REFERENCES

- [1] S. R. Chidamber and C. F. Kemerer, *Towards a metrics suite for Object-Oriented design*. ACM, vol. 26, no. 11.
- [2] R. Malhotra and A. Chug, "Application of group method of data handling model for software maintainability prediction using object oriented systems," *International Journal of System Assurance Engineering and Management*, vol. 5, no. 2, pp. 165–173, 2014.
- [3] M. Grottke, L. Li, K. Vaidyanathan, and K. S. Trivedi, "Analysis of software aging in a web server," *IEEE Transactions on reliability*, vol. 55, no. 3, pp. 411–420, 2006.
- [4] J. Zheng, H. Okamura, L. Li, and T. Dohi, "A comprehensive evaluation of software rejuvenation policies for transaction systems with markovian arrivals," *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1157–1177, 2017.
- [5] N. Padhy, R. Singh, and S. C. Satapathy, "Enhanced evolutionary computing based artificial intelligence model for web-solutions software reusability estimation," *Cluster Computing*, pp. 1–18, 2017.
- [6] "The promise repository of empirical software engineering data," 2015.
- [7] R. Natella, "Aging-related bugs and software complexity metrics Aging-related bugs and software complexity metrics," May 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.581659>
- [8] K. Aggarwal, Y. Singh, A. Kaur, and R. Malhotra, "Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study," *Software process: Improvement and practice*.
- [9] R. Malhotra and Y. Singh, "On the applicability of machine learning techniques for object oriented software fault prediction," *Software Engineering: An International Journal*.
- [10] S. R. Chidamber, D. P. Darcy, and C. F. Kemerer, "Managerial use of metrics for Object-Oriented software: An exploratory analysis," *IEEE Transactions on Software Engineering*, vol. 24, no. 8, pp. 629–639, 1998.

- [11] C. Sun, B. Wang, Y. Liu, D. Sun, and D. Chen, "Numerical simulations for near-field acoustic holographic data extrapolation based on the neural network elm method," in *Information Science and Electronic Engineering: Proceedings of the 3rd International Conference of Electronic Engineering and Information Science (ICEEIS 2016)*, January 4-5, 2016, Harbin, China, vol. 50. CRC Press, 2016, p. 87.

A new design framework of the aesthetic data visualization

Qi Li

School of Art And Design
Shanghai University of Engineering Science
Shanghai, China
qili@sues.edu.cn

Chuanhong Xu

School of Art And Design
Shanghai University of Engineering Science
Shanghai, China
chuanghongxu@sina.com

Abstract—Data visualization is adopted as a visual tool identifying patterns for analyzing and evaluating complex data. Recent research demonstrates that aesthetics have potential to enhance emotional engagement in user perception of visual images. This paper examines the possibility of applied Chinese *yijing* aesthetic to unification of user's subject experience and objective data. *Yijing* presents a unity of the spirit of subject and the reality of object in Chinese traditional paintings, which is a unification of void and reality. This paper proposes a theoretical framework integrating aesthetic into data visualization to enhance users' emotional engagement and data as creative media with the interactive technology that provides a general guidance of data visualization design.

Keywords- Data Visualization, Aesthetics, *Yijing*, Design, Creativity, Interactive technology

I. INTRODUCTION

Data visualization is used as a tool for communicating information effectively and efficiently through graphic representation (Friedman 2008). It aims to apply visual forms to display facts or patterns hidden inside abstract data. Those facts or patterns include natural phenomena or medical contexts, such as weather models or the structure of the cells. This process of data visualization emphasizes the functional and usability requirements for easy comprehension (Gaviria 2008). The forms of visualization initially used in communication purposes have 2D or 3D design, such as charts, graphs, waveform and 3D images. These visual forms share similarity that is recognition, readability and meaning. Those characteristics of form are moderated by the key criteria for effectiveness related to accuracy of communication and efficiency in terms of the speed at which meaning is extracted. Therefore, users may able to quickly and accurately recognize, read and understand the meaning of visualized data through the organized visual forms. Recent research demonstrated that aesthetic design of data visualization could enhance the human's perceive and will engage analysis more in the task (Cawthon & Moere 2007). In addition to attractive data visualization, artists have adopted data or information as a new creative medium for their artistic practice. In the recently year, artists adopt data from external sources and collecting data as part of their artworks. It emerges as a concept of "beautiful data" represented as a visual image, which involves pleasing data patterns.

This paper therefore proposes a design conceptual framework that proposes a relationship along a continuum

between communication, aesthetic experience and creative practice. The model demonstrates the relationship between the characteristics of communication, aesthetics and artistic expression, the typical forms, and how they are perceived by users. Based on this model, a new aesthetic design of data visualization that integrates the aesthetics of traditional Chinese *yijing* aesthetic and digital technology is proposed.

II. AESTHETIC REPRESENTATIVE DATA

Aesthetics in data visualization has been gaining attention as a means to promote a positive effect in viewers that enhances experience and amplifies the ability to interpret information. This has been recognized within the discipline of design, where research has demonstrated that visual attractiveness is a major factor in how people engage with content (Gaviria 2008). If data visualization has a high aesthetic quality, users may be encouraged to engage in a deep level of interpretation.

Research shows that aesthetic can promote the traditional focus of visualization on accuracy, efficiency and effectiveness (Card, Mackinlay & Shneiderman 1999). At the same time, it can be used to prompt a subjective experience in the form of an emotional response. Tateosian, Healey and Enns (2007) emphasize a role for creativity in aesthetic visualization to engage users' attention in reaction to a stimulus. They argue that artistic techniques such as abstract representation and brush strokes techniques have been used to enhance aesthetic perception of visualization through non photo-realistic forms. Therefore, an aesthetic approach to visualization is quite different from traditional visualization methods. The traditional methods emphasize the perception of communication with speed and accuracy. This paper suggests that the Chinese *yijing* aesthetic has potential used to demonstrate this, enhancing the perception of experience by fusing subject and object and promoting pleasure and engagement.

III. CHINESE *YIJING* AESTHETIC

The notion of *Yijing* is a traditional Chinese aesthetic concept that refers to 'an exceptional state of mind' or 'an artistic conception' (Li 2009). It emphasizes the expression of an artist's inner emotion through representation of their subjective experience to the natural world (Zhang 2005). It suggests that *yijing* is similar to the Western concept of "empathy" that involves "the melding of the appreciating (or creating) self with the appreciated (or created) object" (Li 2009,

p. 152). The phenomenon of “empathy” in Chinese aesthetics is the fusion of feeling and scene, which unifies self and object (Li 2009; Zhang 2005). The *yijing* aesthetic has been applied to literature, poetry, language, visual art and music (Liao 2011). In traditional Chinese painting, *yijing* constitutes criteria for the judgment of the value of the artworks (Liu & Bralowskiz 2010). For example, a ‘good’ traditional Chinese painting usually possesses the *yijing* aesthetic quality, which is linking to a perfect melding of poem and image (Zong 2005). *Yijing*, therefore, can be considered as a means by which aesthetic media such as painting can be integrated into visualization to promote a positive subjective response aligned with its inherent values.

IV. PROPOSED CONCEPTUAL FRAMEWORK

In Figure 1, a conceptual framework displays the relationship between the communication-based, aesthetic approaches and art practice to data visualization. It is argued that this model can be used to inform the design principles for data visualization. It could be a valuable design framework for design of data visualization depending on the different situation.

Figure 1. The proposed visualization conceptual framework

It is an axiomatic approach in that it represents a vertical data continuum, from data, through information, to aesthetics and to creativity. Each is a higher level of processing undertaken, initially through organization, through to design and then to expression. As the complexity of the data is reduced, the emotional experience of the user is enhanced, with the horizontal axis delineating the processes inherent in developing the three forms of representation on the left, with the user on the right. The interplay between the user and the representation is portrayed as factors that moderate or influence perceptions. While communication is primarily influenced by the need for speed, accuracy and efficiency, the user’s experience is amplified by engaging with a more aesthetic representation. Therefore, the quality of the experience varies due to the nature of representation. Information presenting a clear way promotes understanding while aesthetics promote pleasure and engagement. Inevitably some relationship still exists between the need for communication and emotional engagement, though it would be expected that the objective communication elements may become increasingly ambiguous as a subjective emotional response is elicited from the user. The important factor in this framework is the differentiation between pure organization,

design and expression as the main processes in generating the representation, with the latter process integrating aesthetic characteristics. Both information and aesthetics are defined by recognizable characteristics, though the forms may vary. In this framework, the *yijing* aesthetic achieves the specified characteristics of unity, harmony and so on through specific forms of reality, void and rhythm, unity, and the evocation of imagination of nature. Reality can be understood as tangible objects, which are represented as a realistic form with a focus on clarity. These ‘real’ objects constitute the most important parts in an image in terms of information. Void can be understood as emptiness, considered as the less important parts in an image. Void and reality have a complementary existence in visualization. This balance generates a visual rhythm that brings emotional harmony between subjectivity and objectivity, and evokes imagination.

While presented as information and emotional engagement, data can also be the media for art creativity. This model adopts gesture based technology to represent interactive data visualization. It suggests interacting with data engaging user’s body building on *yijing* aesthetic characteristics focus on embodied interactivity, pleasure, harmony and balance through the complementarity of all elements rendered unreadable and unrecognizable. In addition, gesture based technology is used as an interactive technology for immersion and experience in the design of data visualization. As the visual representation of data is not recognized, it emphasizes that user’s experience.

V. DESIGN OF AESTHETIC VISUALIZATION WITH THE *YIJING* AESTHETIC

As an instantiation of the theoretical framework proposed in Figure 1, a design has been undertaken that investigates the relationship between communication-focused waveforms and experience-focused *yijing* aesthetic. The paper applies sleep EEG data as source for visualization design. Sleep EEG data can be considered as a typical scientific data, which is seminal to human experience that has many connotations, such as divinatory or preposterous. As such, it forms an ideal medium for exploring aesthetics, particularly given the strong historical fascination with dreams or dream imagery within the Art disciplines (Shapiro, Sherman & Kryger 2013). It has been designed as two stages as following:

A. Approach to Design – Data as living objects

Sleep EEG data, which has been visually represented in this work, is considered a living object that generates visual rhythm through data movement as lines rather than as static data that conveys information for comprehension (Figure 2). Indeed, the data is neither recognizable nor readable, and it does little to identify any valuable insights by its visual representation. In contrast, according to the proposed framework, the work emphasizes users’ experience of a fusion between an individual’s subjectivity and the so-called objective world of Western epistemology. The value of pleasure is generated through the design process.

Figure 2. The screen shot of sleep EEG data visualization with Yijing aesthetic value

This design is an experimental work that specifically seeks to explore the evocation of the imagination through applying aesthetics to data visualization. As such, the focus is not primarily on efficiency of communication. Nevertheless, it is an accurate depiction of the data, which allows for recognition of the concept of sleep within a broader aesthetic interpretation. It is proposed that any deficit in accuracy and efficiency of communication is addressed through enhanced perception and sensation of experience.

B. Approach to Expression – Data as performance

This design work is based on the previous work, extending upon the engagement with embodied interactivity that was established by it. Gesture based technology was used to achieve this new aesthetic. Kinect is considered an ideal gesture-based technology with its distinct interactivity that enables one to “capture, track and decipher body movement, gestures and voice. The auditory and visual information serves as commands to interact with digital contents presented in games or software programs” (Hsu, 2011, p. 365). The work was inspired by Chinese Taoist health philosophy that emphasizes the practice of taijiquan performance with the flow of qi (void) moving through the body in order to achieve harmony and self-transcendence (Figure 3). The work is an outcome of this digital practice that attempts to represent sleep EEG data in the form of qi that users are able to interact with through an embodied data. This digital body is a kind of embodied digital body that unifies the user’s body and data associated with the digital technology.

Figure 3. The screen shot of sleep EEG data visualization with interactive creativity using Kinect

VI. CONCLUSION

The study explores how data, information and aesthetics operate as a communication continuum. It has been shown that visualization can enhance communication through organization

with a focus on speed, efficiency and accuracy while current research has highlighted the role of aesthetics in integrating perception and sensation to create an emotional relationship that engages deeply with users and also approach to artistic creation in which data as medium. The conceptual framework described here promotes a new aesthetic approach to data visualization that has the potential to enhance the end user experience. By applying the form of the traditional Chinese *yijing* aesthetic to data visualization using sleep EEG data, a creative work has been developed that integrates traditional Chinese philosophy with a focus on void and reality, unity, rhythm and the evocation of imagination in terms of natural themes. By adopting gesture based technology to create embodied data visualization, an interactive work has been presented to a new interactive experience with data and technology. The visualization work will allow the forms and characteristics of information and aesthetics to be interrogated in a manner that identifies the key organizational and design principles required not just to communicate concepts but to create a heightened experience for users when they integrated into data visualization.

ACKNOWLEDGMENT

The project was supported by Shanghai IV Summit Discipline in Design and Shanghai Pujiang Talent Program.

REFERENCES

- [1] SK. Card, JD. Mackinlay, & Sneddon, I. N *Readings in Information Visualisation: using vision to think*, Academic Press, San Francisco, CA, USA.
- [2] V. Jacobs, Data visualisation and infographics. viewed 2 September 2013, <http://www.smashingmagazine.com/2008/01/14/monday-inspirationdatavisualisation-and-infographics>, 2008.
- [3] AR. Gaviria, ‘When is information visualisation art? Determining the critical criteria’, *Leonardo*, vol. 41, no. 5, pp. 479-482.
- [4] H. J. Hsu, The potential of Kinect in education. *International Journal of Information and Education Technology*, 1(5), 365-370, 2011.
- [5] Z. Li, *The Chinese Aesthetic Tradition*, University of Hawaii Press, Honolulu. 2009.
- [6] TL. Liao, *Yijing: Mr. Wang meng-ou's Aesthetic of Language*, vol. 25, no. 1, pp. 131-172. 2011.
- [7] S. LIU, & TW. BRALEWSKIZ, Chinese Landscape Inspiration – “Yijing”, viewed 10 March 2019, <http://www.fem.uniag.sk/acta/download.php?id=1014>
- [8] CM. SHAPIRO, D. SHERMAN, & MH. KRYGER, ‘Sleep in art and literature’, in MH Kryger (eds), *Atlas of clinical sleep medicine*, Elsevier Health Sciences, London. 2013.
- [9] LG. TATEOSIAN, CG. HEALEY, & JT. ENNS, ‘Engaging viewers through nonphotorealistic visualisations’, Paper presented at the Proceedings of the 5th international symposium on Non-photorealistic animation and rendering. 2007.
- [10] W. ZHANG, *The Inheritance and Contribution of Zong Huabai's Yijing Theory*, viewed 16 April 2018, <http://www.kyoai.ac.jp/college/ronshuu/no-05/cho.pdf>
- [11] B. ZONG, *A Walk with Aesthetics*, Shanghai People's Press, Shanghai. 2005.
- [12] MA. CARSKADON, & WC. DEMENT, *Qualities of Perceived Aesthetic in Data Visualisation*, Paper presented at the Conference on Designing for User eXperiences (DUX'07), Chicago, Illinois, viewed 27 November 2018.

When E-Learning Meets Big Data, Cognitive Computing, and Collaborative Environments

Mauro Coccoli
University of Genoa
DIBRIS
Genoa, Italy
mauro.coccoli@unige.it

Paolo Maresca
Federico II University
DIETI
Naples, Italy
paolo.maresca@unina.it

Andrea Molinari
Dept. of Industrial Engineering
University of Trento, Italy
School of Industrial Engineering and
Management, Lappeenranta
University of Technology, Finland
andrea.molinari@unitn.it

Abstract— The e-learning scenario is fast-changing in all of its main aspects from both the technology side and the users' perspective. In this vein, applications and services as well as methodologies are evolving rapidly running after the more recent innovations, through the adoption of novel infrastructures and distributed architectures, to provide the most advanced solutions. In this situation, we observe that the more influential technological factors are two. The former is the availability of a new wave of big data also inherent educational tasks, which can be considered a powerful source for all the people involved in the educational process and acts as enabler for the development of new models for both management and education, based on analytics. The latter is cognitive computing, which can provide learners and teachers with innovative services enhancing the whole learning process, also introducing new capabilities to improve human machine interactions. In these circumstances, it seems that Learning Management Systems (LMS) have reached their maturity stage and they may need a redesign. In this respect, the paper outlines the evolutionary trends of Technology Enhanced Learning (TEL) and presents the results achieved within two experiences carried on in two Italian universities.

Keywords—cognitive computing; big data; collaborative systems; technology enhanced learning; e-learning

I. INTRODUCTION

Both technology and society are undergoing a continuous evolution process, which is pushing innovation and driving the development of novel solutions in almost any field of application. The wide adoption of such solutions in a plethora of services that are commonly available on the Web for daily operations heavily impacts on users' behaviors and expectations. Specifically, with reference to the Technology Enhanced Learning (TEL) scenario, we observe significant steps ahead in techniques and methodology; consequently, the relevant technological solutions are subjected to continuous upgrades to cope with these, to the aim of improving the quality of services, the usability, the overall performances, the effectiveness of education, and to provide learners with a more pervasive experience. Accordingly, e-learning environments and the related tools have been growing in complexity, i.e., traditional Learning Management Systems (LMS), based on a centralized software architecture, are moving towards clusters of Massive

Open Online Courses (MOOC) platforms in cloud-based distributed architectures [1] and Learning Objects (LO) include videos and multimedia interactive artifacts. Concurrently, LMSs have been exploring new spaces of possibilities; among these, we mention the adoption of users' profiling techniques and analytics to the aim of tailoring personalized learning paths and activities as well as to predict students' careers, to achieve a further empowerment of the individual learning model. Anshary et al. provide us with a clear synthesis of the more significant evolutionary steps of learning solutions [2] by depicting a timeline that shows the changes occurred in tools, services, learning strategies, methodologies, and delivery techniques. The authors highlight that each progress achieved is not a replacement of the previously adopted solution, yet it implements one or more of the following: (i) extend capabilities; (ii) improve performances; (iii) promote different educational approaches and methodologies; (iv) change the way contents are delivered; (v) change users interactions model. Despite the methodology is sound, their analysis concerns the period from 2001 to 2015 and, since then, many things have evolved mainly owing to the rise of big data and cloud computing.

In this respect, we highlight that the adoption of cognitive computing solutions can be the technological enabler for a number of new functionalities [3]. Moreover, the introduction of the cognitive computing paradigm also impacts on the learning process, as schools and universities must face with new jobs and new training demands, where big data contexts are fundamental, since new knowledge and skills should be promptly delivered [4]. As a result, it seems that LMSs have reached their maturity stage in the innovation adoption curve. In fact, they are pervasively adopted by learning providers at any level, from the primary school to higher education institutions, to guarantee the effective implementation of TEL solutions, yet they hardly can cope with the high degree of interoperability and complexity that novel paradigms require, hence should expand their boundaries offering new services and also involving industries and the labor market in the learning design and process. In this respect, the paper discusses the evolutionary trends of e-learning and presents the results achieved within two experiences carried on in two Italian universities: (i) an innovative didactic approach to software engineering, which adopts advanced technologies and mixed solutions in a cloud infrastructure; (ii) a novel

collaborative learning environment that surpasses the typical functionalities of the more prominent LMSs. We highlight that such experiences are driving TEL platforms towards new standards where both learners and teachers, rather than technology, will be at the center of the process.

The remainder of the paper is structured as follows: in Section 2 we present some related works investigating these topics; Section 3 outlines the main drivers for the development of the next generation TEL platforms; in Section 4 we report the experience of a cognitive-computing-based laboratory, while in Section 5 we depict a newly developed e-learning collaborative environment. Section 6 concludes the paper also giving hints on future development directions.

II. RELATED WORKS

As highlighted in the previous section, we are going to focus mainly on two phenomena: (a) big data, and (b) cognitive computing. In fact, the possibility of gathering information from a wide number of heterogeneous sources, combined with the unprecedented opportunities of processing such data by means of cognitive computing techniques, is reshaping the technological scenario of e-learning applications, allowing the improvement of existing methodologies as well as imagining new ones. Moreover, the possibility of recording, storing and aggregating information, can significantly improve learning performances of both students and teachers [5]. For these topics, we report a number of related works that illustrate different solutions showing how variedly these two aspects can empower e-learning methodologies and how platforms are evolving with new functionalities. From a quick analysis of the relevant literature, i.e., searching indexing services such as Science Direct, Scopus and Web of Science, we found several publications on these challenging topics, enlightening that, more and more, LMSs offer such services among basic tools or as add-on functionalities.

A. The Impact of Big Data on E-Learning

Due to the presence of big data, various changes have occurred in educations and in e-learning systems. Their impact on the academic environment, which covers a large scope and has specific peculiarities, is investigated by Banica and Radulescu. Their expectation is that a change will suddenly occur in the way e-learning is approached by students and teachers. In this respect, based on the currently available software solutions, they propose a system architecture fostering universities to constitute consortia to analyze, organize and access huge data sets in common, in a cloud-based environment [6]. A similar approach is followed by Mahmud and Ali, which propose a strategy to improve the education process output through the collaboration with other international universities, to the aim of supporting the teaching and learning process by means of shared e-learning services. The integrated framework they propose links the individual impact with the organizational impact, promoting a collaborative culture model [7]. As hinted, another key factor for the empowerment of e-learning services, is their integration with information from other platforms and services given that a large amount of data is accessible in, e.g., online communities, blogs, discussion forum, messaging services, and social network sites. In this respect, Dietz-Uhler

and Hurn showcase some effective learning analytics techniques to derive knowledge from large blobs of information. Specifically, they focus on tracking students' data, to help them succeed. To this aim, they investigate which different learning analytics tools are adopted in different universities and institutions and discuss how faculty can exploit such data to monitor and predict students' performances, to finally enhance them [8]. Moreover, Kolekar et al. are observing the same large amount of information freely available over the Web, exploring the opportunities of using such data to get enhancements not only in assessments but in all the stages of the learning process [9]. Following the same current, Yu and Jo also show an example of how to fruitfully exploit big data for the prediction of students' performances, to the aim of optimizing their careers [10]. We point out that, currently, social media can play a vital role with respect to e-learning systems and the effectiveness of information is strongly tied to the way we process these data. In this respect, also Sheshasaayee and Malathi believe that the application of big data with e-learning is a hot topic, which has the potential for creating a huge impact on the whole education system [11].

We also highlight that processing big data in an effective way is possible only thanks to cognitive computing solutions implementing suited machine learning techniques, able to cope with data characterized by large volume, different types, high speed, uncertainty and incompleteness, and low value density. Going deep in technical details, algorithms and methods (e.g., representation learning, deep learning, distributed and parallel learning, transfer learning, active learning, and kernel-based learning) is out of the scope of this paper, however the interested readers can find a survey of how machine learning is used for big data processing in a paper by Qiu et al. [12]. Other references can be found in [13], where Gupta et al. present a large understanding of past, present and future directions in this domain, made through a mapping of the characteristics of cognitive computing, i.e., observation, interpretation, evaluation and decision, versus the so-called V's of big data, i.e., Volume, Variety, Veracity, Velocity and Value.

B. The Impact of Cognitive Computing on E-Learning

From the side of cognitive computing, only recent works can be found that deal with more advanced services for e-learning applications. For example, they can serve the instructional design process, helping to find personalized learning assets and improving the definition of individual learning strategies or classifying resources. Specifically, Leitão et al. focus on recommendation systems implemented on the basis of cognitive services. They envisage the possibility of using the same approach to meet the needs of students and teachers, especially to enable personalized learning strategies and recommend educational resources, based on information derived from the interactions between students. The paper proposes the prototype of a platform and surveys the approaches to develop advanced TEL solutions, analyzing the state of the art of using cognitive systems in e-learning, identifying paradigms and pedagogical methodologies, techniques, tools and learning objects with respect to the recommendation of pedagogical activities using cognitive computing [14]. In [15], the authors use cognitive systems for the automated classification of learning videos, with special reference to MOOCs. Dessi et al. exploit the capability

of Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR) services to extract text from audio and visual frames so to be able to perform a classification based on a taxonomy. Further development of their work led the authors to find a solution to analyze the content of large video collections, overcoming traditional term-based methods by means of cognitive services such as a Speech-to-Text tool to get video transcripts and the use of Natural Language Processing (NLP) methods to extract semantic concepts and keywords from the above video transcripts [16].

C. Empowering E-Learning Methodologies

Besides, from the methodology perspective, Cen et al. introduce the idea of “big education” applying the paradigm of big data to the whole education process to predict students’ performances, based on individuals’ learning attitudes and their after-school activities [17]. This seems a promising vein, since also Gudivad theorizes about cognitive analytics driven personalized learning, which can be achieved owing to the advances in cognitive computing for analyzing unstructured data, e.g. blogs, discussions, e-mail, and course messages, to gain insights into student learning at an individual level [18]. New functionalities are strictly connected to new technologies such as, for example, the mix of learning and semantic technologies through the use of ontologies for the description of domain(s) and, in this specific case, the availability of sophisticated cloud infrastructures is required to handle properly such a huge quantity of information, as well as the design and development of new learning environments, supporting suited machine learning technologies [19].

D. Privacy and Security Issues

Finally, we observe that when considering big-data-capable learning applications another paramount item emerges, that is the students’ data protection. In fact, personal information in the e-learning frameworks can be very detailed, thus very precise profiling can be obtained and used maliciously for different scopes, such as, e.g., remarketing. As well as privacy, security cannot be neglected and should be considered one of the most important factors in the design of TEL platforms [20]. For example, this topic was faced by Habegger et al., which clearly present possible threats of considering big data within e-learning platforms [21].

III. EMERGING TRENDS IN TECHNOLOGY ENHANCED LEARNING

According to the above considerations, we propose a new point of view on TEL platforms, enlightening the main characteristics they should offer. Their design must consider the new wave of cognitive services for their use in advanced solutions and for their ability to cope with the huge amount of data circulating within the learning frameworks and the connected software environments outside the LMS (e.g., discussion forums and blogs, social network sites, indexing services, digital libraries, etc.). In more detail, personal data about learners, their learning tasks, scores in the assignments, etc. should be stored and historicized to the aim of improving the whole learning and teaching process, also including administrative information to monitor how learning processes are conducted and make an assessment to the aim of predicting

performances through suited learning analytics. More precisely, we state that the next-generation TEL platforms will offer pervasive cognitive services.

From the side of the learning process, we have to consider two different aspects: (i) learners will acquire new knowledge thanks to the educational strategy and methodology adopted, and (ii) at the same time, machines, systems and platforms will acquire information about learners and their individual learning processes, owing to cognitive services. However, thinking of cognitive services merely as the result of an algorithm is not a good starting point: there is something more because learning does not derive from a software algorithm but also from the complex hardware architecture it relies on. In fact, cognitive services can be effective only within a parallel architecture, whose capabilities must be suited to the learning needs of the algorithm itself. In other words, since the algorithm must be trained, we often need substantial memory and computational resources to be implied in educational processes based on a cognitive approach. The personalization of cognitive services may require additional hardware and software resources and thus, for this reason, when defining a cognitive TEL, we have to go beyond a basic hardware configuration to face to possible personalization issues, which may require a system flexible enough to adapt over time its characteristics to the use we will make of it, in the perspective of a lifelong learning support to our knowledge growth. Such features can be achieved through the adoption of cloud architectures, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) solutions. Furthermore, feedback is needed to improve the effectiveness of this approach, and we should be able to observe the system and get feedback to perform real-time tuning operations.

In the light of the above considerations, it is required to build innovative architectures for the platforms that aim to provide innovative didactics, able to change configuration quickly in order to adapt its functionalities to the different needs of both learners and teachers. In particular, we believe that a wise adoption of the best of some services from multiple vendors, in the same platform, could improve the satisfaction of end users as well as solve some structural problems of the laboratories. In fact, often university laboratories need to be scalable for different number of students attend classes or they may need to serve different educational activities (e.g., quizzes, lessons, exercises, application laboratories, exams, etc.). Sometimes they must even be scaled up to more applicative situations, i.e., performing tests for research projects or experiments. Specifically, Section 4 provides a detailed description of the innovative learning environment that the Federico II University in Naples, Italy, setup, based on the above concepts.

Moreover, we observe that e-learning platforms and applications continuously change together with learning itself and learners’ attitude and needs. In many cases, educational paths are designed in collaboration with institutions from other countries, i.e., to promote mobility of both workers and researchers, in other cases a strict interaction with industries and the labor market is required. Consequently, performing educational tasks in a “classroom”, whether real or virtual, may be limiting the perspective of what today a smart education should be. Hence, in order to stay competitive, formal education

models should expand their boundaries also involving the external world and this can only be achieved through the adoption of suited technological solutions for online collaboration, interoperability, data exchange, and the seamless integration with legacy systems. The implementation of such solutions requires suited machinery and infrastructures as well as a paradigm shift able to drive the transformation of classrooms into communities to enhance humans' faculties and empowering the transformation of their skills so that, in such new "places", we can speak of men *and* machines, rather than men *or* machines. Even in this case a specific example is presented and Section 5 depicts the characteristics of the innovative e-learning platform developed at the University of Trento, Italy.

IV. AN INTEGRATED ENVIRONMENT FOR EXPONENTIAL LEARNING

This section provides a detailed description of the innovative learning environment setup at the Federico II University in Naples, Italy, based on cognitive computing to implement exponential learning. We recall that cognitive computing was born based on human reasoning models and this special capability can be exploited transversally on, potentially, any domain of application, so that it can be regarded as a real revolution, exponentially accelerating processes. In the past decades, we have been accustomed to a swirling growth, according to the Moore's law (the number of transistors in a dense integrated circuit doubles about every two years) or the analogous Metcalfe's law for the number of nodes of a network that clearly explains the enormous dimensions reached by social networking sites such as, e.g., Facebook. However, we were not prepared to cope with the needed exponential growth of learning caused by the growth of knowledge, which is in turn derived by the growth of data processed with novel Artificial Intelligence (AI) techniques. Consequently, the learning environments should be promptly adapted to this reality.

A. The Hardware and Software Setup

According to the above considerations, the Federico II University in Naples, Italy, setup a laboratory with suited hardware and software configurations, providing students of the "Software engineering" class with the possibility to use the most advanced methodologies and tools to develop quite complex projects within their assignments and to share their results with mates. Previously, the authors already carried on didactic activities focusing on computer programming in collaborative environments (see, e.g., [22], [23], and references therein) but cognitive computing has accelerated this need, thus enabling more ambitious projects to be implemented both from the point of view of the hardware resources that can be used and the complexity of the software that can be used. The implemented solution is an integrated environment offering high-level tools for cognitive computing design and programming. Specifically, after a preliminary analysis and a series of stress tests, Microsoft Azure and IBM Cloud have been identified as the best-matching solutions for developing cognitive-computing-based projects. Owing to their cloud-based architecture, both platforms can coexist in the same installation, where they can be used simultaneously if needed, allowing to realize the needed PaaS and SaaS infrastructures.

More precisely, students can use the cognitive e-learning platform from both IBM Cloud and Microsoft Azure with relevant laboratory and customized virtual machines. Taking advantage of such a mixed configuration, enables students to disregard hardware and software issues, focusing on their individual learning tasks. Besides, they can use complex machines simply through a browser, which is a strong point because they may have to use laptops or even personal computers with poor performances. To better clarify, we highlight that the Azure environment offers different profiles for sizing the hardware machine. One can choose one of the following: (i) courses, (ii) laboratories, and (iii) exams. Then, the decision to instantiate a laboratory session or to take part in an exam session is up to the user. The only requirements are the availability of a PC and a network connection. In more detail, the teacher can, in a laboratory consisting solely of the PCs that students bring with them, instantiate an exam session on the fly. Then, the students will have to use their institutional credentials to log into the system, where they can access a personalized dashboard and choose which activity to carry on such as, e.g., laboratory, exams etc. Even if a student is taking an exam, in the same classroom, at the same time, other students can perform different activities. For example, writing and compiling a Java program within the Eclipse IDE as well as entering the IBM Cloud platform, and also use, e.g., the IBM Watson cognitive services.

B. Results

At the end of the activities, the platform allows students to assess the effectiveness and ease-of-use of the environments they used. In particular, they were asked to provide their overall feedback on satisfaction and usability. In order to evaluate the global satisfaction of the platform we considered an ordinal scale with values between 5 and 8. The students were asked to express their opinion after working on the platform for several hours. More precisely, Figure 1 shows that 87% of users gives a medium-high evaluation (scores 7-8), while only 13% believes that the platform has low-sufficient usability (scores 5-6).

Figure 1: Results from the questionnaires on users' satisfaction.

Since these integrated environments of multiple technologies may be too complex for young students, we have also asked a feedback on usability whose results are shown in Figure 2. In

order to measure the usability of the platform we used an ordinal scale of values between 5 and 9. Students, in a slightly lower number (36 vs. 38), expressed their opinion about the usability of the platform after working on it for the didactic projects they were engaged in. More precisely, Figure 2 shows that the 61% of students gives a medium-high evaluation (scores 7-8-9), while only 39% (scores 5-6) believes that the platform has low-sufficient usability. Besides, students were also asked to express suggestions and possible improvements. From this analysis it emerges that students experienced some difficulties, due to the presence of two different environments but also for the functionalities, rather different from the classical university-laboratory setup they are used to, which offer only basic tools.

Figure 2: Results from the questionnaires on usability.

In conclusion, the integrated environment in use was judged satisfactory with respect to the functions it had to perform. Given its apparent complexity, we consider such results quite positive and encouraging for the prosecution of this experiment also in the next academic years, in new classes and with a growing number of students.

V. FROM CLASSROOMS TO COMMUNITIES

According to the above considerations, in this section we depict the characteristics of an innovative e-learning platform developed at the University of Trento, Italy. As highlighted in previous sections, classrooms should shift to communities and in such new places both students and teachers will be able to enhance humans' faculties and empowering their skills transformation, also owing to modern technologies and new paradigms for the possible relationships between men and machines. Specifically, we will discuss the Online Community (OLC) project.

A. The Online Community (OLC) Project

The project started back in 1998 with the idea of implementing a different approach to educational content management, in contrast to proprietary platforms like WebCT™ and Blackboard™, which were dominant at that time. In this scenario, OLC was created from scratch, after having considered three possible alternative solutions: (i) adopting a commercial platform; (ii) adapting free-open source software to the needs of educational environments, as Moodle or similar where not existing yet (iii) building a brand-new platform. Finally, the third

option, despite having the traditional pros and cons of every "make" solution, was chosen for other motivations that now, in the idea of innovating e-learning systems with new cognitive services, revealed to be the right decision. Compared to the adoption of commercial software, the Total Cost of Ownership (TCO) of this kind of solution was probably too high especially in the hypothesis of an extension, not counting the rest of the associated costs for maintenance, management and training, when compared to budget limits. We observed in the following years that this is a very common reason for the adoption of Free and Open Source Software (FOSS) platforms such as, e.g., Moodle by many small-medium educational institutions like high schools and universities. Put simple, being free of charge is the main reason for adopting this solution, without taking care of the side effects on the educational processes that it implies. In fact, a software platform implements processes in a rigid way once they have been coded, so the adoption of this component forces the institution to customize, adapt or even change the way things can be done because "the system does not allow me to do differently". As a secondary effect, the connection of the LMS with the rest of the organization's information system is mostly impossible [24]. As a consequence, many system administrators, are adapting their needs to the software system that, somehow, is able to solve most of their problems, and they mostly are resistant if not reluctant to develop an internal solution. Money, availability of qualified resources, short time to implement the solution, these all are comprehensible reasons for choosing the easy way of acquiring a pre-cooked solution.

We consider much more important to have the possibility of investing in a platform that can be easily extended with new services according to the needs of those trainers willing to experience the use of computer technologies in their educational processes. We therefore decided to develop a completely new platform, and very soon the platform was transformed from a mere LMS to a more structured platform devoted to support collaboration among members of a virtual community. The idea of "classroom" that is lying behind most of the LMS available today is, in our opinion, very restrictive with reference to the more complex processes that normally happen inside educational tasks, and can be extended to any other collaboration environment where collaboration among participants to the community are mediated by technologies. In other words, the idea of a customized LMS that could constitute a competitive advantage for one university versus another one is established [25]. This personalized software is able to supply better and personalized services that ease procedures and processes for the different users such as students, professors, or administrative personnel. Finally, a second decisive item was the need of deep integration with the rest of the legacy information system: authentication with single sign-on, integration with exam records and administrative procedures, possibility of bi-directionally exchange news and messages among people living the university day-by-day routine. Substantially, creating the platform from scratch was related with the rejection of "one-size-fits-all" approach to software components of an information system. Many administrators of the information system, especially in the educational sector, are adapting their needs to the software system that, somehow, is able to solve most of their problems, and they mostly are resistant if not reluctant to develop an internal solution. Money, availability of qualified

resources, short time to implement the solution, these all are comprehensible reasons for choosing the easy way of acquiring a pre-cooked solution.

This adaptation of educational tasks to software platforms is a typical situation where many institutions are lying today. Because teams not always have the knowledge and resources to modify existing (open source) software educational platforms (i.e., LMSs like Moodle), they normally “adapt” themselves to what the platform supplies out-of-the-box, thus limiting the innovation potential of their ideas, and forcing users to adapt their learning processes to the improper technological tool. The typical example is the use of social media (the one that is more appreciated by learners in that moment) to support educational tasks. Social media such as, e.g., Facebook, Twitter, WhatsApp and Instagram are great tools when applied to the context they have been originally created, mostly exchanging multimedia information among peers. Yet, it is not so easy to integrate them in the educational processes: it is clearly a technical problem of available software Application Programming Interfaces (API), but it is also an instructional design problem introducing issues on how to cope the style of the lecture, how is changing the role of the teacher and what are the expectations of learners about the use of social media. In fact, an educational process is something wider than posting a photo or retweeting others’ comments, even if it can benefit of this situation. Sometimes, educational processes need the support of other tools and services, that Facebook (for example) can provide through a distorted usage of its services. This normally forces users (mainly educators) to adapt their learning processes to what the platform provides, while it should be exactly the opposite, i.e., the platform should be able to adapt its services to the users’ needs. The same happens with LMSs: most of the educational organizations have no possibilities of intervention, nor adaptation or modification on software platforms that have the size and complexity of Moodle, and so they adapt their educational processes to what the chosen platform provides as standard services.

B. Latest Innovations

The innovative aspect that we introduced with OLC, and that now constitutes an extra advantage, is therefore reverting this master-slave approach that sees software platform as masters and end-users as slaves. The approach followed by OLC is to construct from scratch those services identified as relevant by educational experts, based on the precise educational needs of the different users: teachers, students or any other role involved. OLC has some architectural aspects there are very important for our argumentation about a next generation of TEL platforms: (i) we own and have created every single line of the source code, so the whole knowledge of the platform architecture and its potential are not scattered among different contributors (like in many open source projects); (ii) the platform is equipped with a micro service architecture, thus allowing an easy extension of the platform itself with new parts but taking advantage of the many services that any LMS should provide both to users and to developers that want to extend it; (iii) some services have been already developed in the past towards the direction of providing “intelligent” services. Owning and knowing the whole source code of the platform means to have a great advantage if you want to extend it with innovative elements, so the idea of using OLC as a basis for cognitive computing has been straightforward.

Nevertheless, some crucial evolutionary changes had to be applied to the platform, and these changes are the key success factors for this shift:

a) stimulate interaction: the platform should encourage users to interact, not just to download files with teacher’s slides. Today, the vast majority of LMSs are used solely to download files, while interactive and more participative services are left to the availability in the platform (if any) and/or at the goodwill of the teacher;

b) pervasive and enriched logging: the platform should log actions of users in order to activate cognitive processes: logging is essential for cognitive computing in order to classify users and infer the best service at the best moment. This logging should not be just the web application server logging, but specialized, application-level logging are needed to capture specific actions inside the single page of the LMS;

c) extensible, service-based architecture: the platform must be extensible through a service-by-design approach, in order to add new services whenever new possibilities can be explored. A micro service architecture is highly recommendable;

d) inference-oriented persistence layer: in order to facilitate inference, reasoning and cognitive computing algorithm, the persistence layer of web platform should be updated to more efficient and flexible data structures [19].

On this basis, a set of profound re-engineering operations has been implemented inside OLC. For example, we extended the platform towards a semantic representation of the knowledge inside the contents of the platform. We also integrated some soft computing, fuzzy-logic-based decision support systems [1], to support decision makers with intelligent tools about educational processes. Moreover, we experimented new storage layers, in order to collect data not only from traditional sources inside the educational environment (i.e., files, forums, galleries, posts, photos, etc.), but also collecting a lot of analytical information about the use of the platform and its services by the users. This immediately opens the problem of the size and appropriateness of traditional relational databases. We performed some experiments in substituting some parts normally stored in relational tables into triples available for a semantic knowledge representation. This meant using triple stores in the beginning as a persistence layer, thus facilitating operations like inference, reasoning, machine learning, etc. The triple (or quadruple) format for persisting (part of) data relevant for decision making and cognitive computing is another step that is not currently available in mainstream LMSs.

To better clarify the above concepts, let us consider as an example the diagram sketched in Figure 3, which depicts the preparation process of proper data sources for big data integration and analysis. The first step is the selection of the data from the persistence of OLC. This persistence is a typical big data source, with structured and unstructured information (file, learning objects, blog posts, forum topics, wikis, etc.) created inside the platform itself. The idea is to separate such data from the rest of the platform, to create the background to be able to apply the cognitive algorithms. So, in a sense, this resembles an Extract Transform and Load (ETL) process, typical in any data warehouse as well as OLAP and data mining solution.

Figure 3: Overview of the big data preparation process for the analytics tools.

The most part of information is coming from the first data source we used in our experimentation, due to its affinity with big data sources, that we call “Actions”. This service collects all data coming from users’ interactions with other OLC objects or services. In practice, it acts like a sensor introduced inside the source code of the platform in any place the software needs to capture an “action” from the user interface. This is a relevant enrichment of the logs recorded by the web application server, and has been used for many different purposes. Due to volume issues, the system at the moment is blocked on collecting only some types of events, to a certain granularity defined by the system administrator. This choice has not been a design choice, but a performance-related one. In fact, it was clear from the early experimentations that the amount of data could have been compromising the capacity of the DBMS to stand with data acquisition pace and volumes: that is a typical “Velocity” and “Volume” big data problem.

In more detail, there are several elements of data gathering and manipulation pushing our virtual learning environment towards big data, thus increasing the need of a structural change of LMSs architecture, which should adopt new approaches and technologies. Specifically, we mention:

- traditional weblogs, being the application a Web-based software;
- internal logs of usage of the platform, the so-called digital breadcrumbs, that track the learner’s journey throughout the entire learning experience;
- mobile logs, where data about mobile learning actions are collected;
- service logs, users’ actions on the different elements of the platform like documents, forums, blogs, FAQs etc.;
- logs from the SCORM player, normally an external entity respect to the core services of the platform, with the records of the objects’ execution;
- X-API calls, in case the platform is connected or acting as a Learning Record Store (LRS);

- MOOCs, by definition a generator of high volumes of data;
- lifelong learning, an old buzzword of e-learning that is still valid and interesting and, most of all, is another generator of big data, specifically along time;
- serious games that will use materials inside the platform, thus generating a relevant dataset related with users’ performances.

The next step in the pipeline is the execution of cognitive tasks which can materialize implicit knowledge to support learning processes. At the moment, this reasoning is limited to basic inference regarding actions performed on certain parts and services of the platform, but the mechanism is ready for larger application scenarios. When the inference process is complete and new knowledge is inferred, a set of administrative routines is executed to load and transform part of the knowledge base to feed applications persistence.

VI. CONCLUSIONS AND FUTURE WORK

As a conclusion, we remark that nowadays we are on a turning point in which cognitive-computing-based approaches are significantly transforming many aspects of our lives. In fact, beside the changes we can notice in professional applications and high-end software and legacy systems, we are already experimenting a set of cognitive computing services in everyday activities, for example, through natural user interfaces and voice assistants, whose presence is becoming pervasive. In practice, more often than we think, we are interacting with machineries that apply sophisticated decision-making processes with very low time constraints and a high level of accuracy. E-learning is one of the fields of application that can mostly benefit from this situation, due to its complexity and to the variety of disciplines that must be adopted concurrently to achieve good learning outcomes. Especially, the use of big data strongly empowers the process of personalization and individualization of the learning processes. Moreover, e-learning is also called to provide suited solutions to the problem of learning to use and exploit such new technologies, which cannot be achieved in environments designed for generic purposes. This raises the problem of developing a new generation of TEL platforms.

The paper introduces the vision of the authors, where a self-made, highly customizable virtual community platform will be integrated with scalable, top-notch cloud platforms and congruent cognitive algorithms applied to the different parts of learning processes, from material selection to educational path suggestions, from peer evaluation to big data discovery for decision makers. The process is still in its infancy, mostly because these three worlds (TEL platforms, cloud services and cognitive computing) are still separated and mostly focus on their own scope. What we are trying to do is merge the three disciplines into one single research area, with precise objectives and deliverables, thus allowing e-learning to maximize the advantages of the fusion of the three.

The two experiences presented in this paper can be regarded as an embryo for the development of future, unpredictable e-learning solutions. To this aim, the results achieved while using the integrated environment for exponential learning deployed at

the Federico II University convinced us to scale the process on a wider and even younger audience, in order to test the real simplicity of such a TEL environment. Another lesson learned is that the surrounding world has become so complex and the changes so rapid that they will never be as slow as in the past. We should not waste our precious time any more in installing software, configuring applications, customizing solutions, tuning databases, maintaining laboratories, etc. Students and teachers definitely have to devote their intellectual power to more creative and less repetitive activities. Moreover, the applications we started using will increasingly have to analyze large amounts of data and therefore be cognitive. Future developments of the OLC also will follow in the direction of providing more advanced cognitive services and making the best from the available information mixing data from different online sources.

REFERENCES

- [1] P. Bouquet, and A. Molinari "A new approach to the use of semantic technologies in e-Learning platforms," *International Journal of Advanced Corporate Learning*, Vol. 9, No. 2, pp. 5–12, 2016.
- [2] M. Anshari, Y. Alas, and L. Sei Guan, "Developing online learning resources: Big data, social networks, and cloud computing to support pervasive knowledge," *Education and Information Technologies*, Vol. 21, pp. 1663–1677, 2016.
- [3] M. Coccoli, P. Maresca, and L. Stanganelli, "Cognitive computing in education," *Journal of e-Learning and Knowledge Society*, Vol. 12, No. 2, pp. 55–69, 2016.
- [4] M. Coccoli, P. Maresca, and L. Stanganelli, "The role of big data and cognitive computing in the learning process," *Journal of Visual Languages & Computing*, Vol. 38, 2017, pp. 97–103.
- [5] M. Fedrizzi and A. Molinari, "A multi-expert fuzzy TOPSIS-based model for the evaluation of e-Learning paths." In *Proceedings of the 8th Conference of the European Society for Fuzzy Logic and Technology*, pp. 554–558, 2013.
- [6] L. Banica, and M. Radulescu, "Using big data in the academic environment," *Procedia Economics and Finance*, Vol. 33, pp. 277–286, 2015.
- [7] M. A. Mahmod and A. M. Ali, "Promotion the e-learning success in universities in Baghdad through enhancing their organizational innovative collaboration environment: A qualitative study," *International Journal on Perceptive and Cognitive Computing*, Vol. 4, No. 1, pp. 12–18, 2018.
- [8] B. Dietz-Uhler and J.E. Hurn, "Using learning analytics to predict (and improve) student success: A faculty perspective," *Journal of Interactive Online Learning*, Vol. 12, No. 1, pp. 17–26, 2013.
- [9] S. V. Kolekar, R. M. Pai, and M. M. M. Pai, "Prediction of learner's profile based on learning styles in adaptive e-learning system," *International Journal of Emerging Technologies in Learning*, Vol. 12, No. 6, pp. 31–51, 2017.
- [10] T. Yu and I. H. Jo, "Educational technology approach toward learning analytics: relationship between student online behaviour and learning performance in higher education," in *Proceedings of the 4th International Conference on Learning Analytics and Knowledge*, 2014, pp. 269–270.
- [11] A. Sheshasaayee and S. Malathi, "Impact and consequences of big data in e-learning," in *Proceedings of the International Conference on Innovative Mechanisms for Industry Applications*, 2017, pp. 726–729.
- [12] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *URASIP Journal on Advances in Signal Processing*, Vol. 67, 2016.
- [13] S. Gupta, A. Kumar Kar, A. Baabdullah, and W. A. A. Al-Khowaiter, "Big data with cognitive computing: A review for the future," *International Journal of Information Management*, Vol. 42, pp. 78–89, 2018.
- [14] G.d. S. Leitão, E. B. Valentin, E. H. T.d. Oliveira, and R.d. S. Barreto, "Survey on pedagogical resources recommendation using cognitive computing systems," in *Proceedings of the IEEE Frontiers in Education Conference*, 2018, pp. 1–7.
- [15] D. Dessi, G. Fenu, M. Marras, and D. Reforgiato Recupero, "Leveraging cognitive computing for multi-class classification of e-learning videos," in Blomqvist E., Hose K., Paulheim H., Lawrynowicz A., Ciravegna F., Hartig O. (eds), *The Semantic Web: ESWC 2017, Lecture Notes in Computer Science*, vol 10577, Springer, Cham, 2017.
- [16] D. Dessi, G. Fenu, M. Marras, and D. Reforgiato Recupero, "Bridging learning analytics and cognitive computing for big data classification in micro-learning video collections," *Computers in Human Behavior*, Vol. 92, pp. 468–477, 2019.
- [17] L. Cen, D. Ruta, and J. Ng, "Big education: opportunities for big data analytics," in *Proceedings of the IEEE International Conference on Digital Signal Processing*, Singapore, 2015, pp. 502–506.
- [18] V. N. Gudivad, "Cognitive analytics driven personalized learning," *Educational Technology*, Vol. 57, No. 1, pp. 23–31, 2017.
- [19] P. Bouquet and A. Molinari, "Semantic technologies and e-learning: Towards an entity-centric approach for learning management systems," *Journal of E-Learning and Knowledge Society*, Vol. 8, No. 2, pp. 65–84, 2012.
- [20] L. Caviglione and M. Coccoli, "Smart e-learning systems with big data," *International Journal of Electronics and Telecommunications*, Vol. 64, No. 4, pp. 445–450, 2018.
- [21] B. Habegger, O. Hasan, L. Brunie, N. Bennani, H. Kosch, and E. Damiani, "Personalization vs. privacy in big data analysis," *International Journal of Big Data*, pp. 25–35, 2014.
- [22] M. Coccoli, P. Maresca, L. Stanganelli, and A. Guercio, "An experience of collaboration using a PaaS for the smarter university model," *Journal of Visual Languages & Computing*, Vol. 31, pp. 275–282, 2015.
- [23] P. Maresca, A. Guercio, and L. Stanganelli, "Building wider team cooperation projects from lessons learned in open communities of practice," in *Proceedings of the 18th International Conference on Distributed Multimedia Systems*, 2012, pp. 144–149.
- [24] L. Colazzo, A. Molinari, and N. Villa, "Social networks, virtual communities and learning management systems: Towards an integrated environment," in *Proceedings of the 8th IASTED International Conference on Web-Based Education*, 2009, pp. 209–215.
- [25] L. Kimball, "Managing distance learning: New challenges for faculty," in *The Digital University - Building a Learning Community*, Springer London, 2002, pp. 27–40.

Event Detection in Complex Environments: An Effective and Efficient Machine-Learning-Based Framework

Alfredo Cuzzocrea

DIA Dept., University of Trieste, Italy
alfredo.cuzzocrea@dia.units.it

Enzo Mumolo

DIA Dept., University of Trieste, Italy
mumolo@units.it

Abstract

In this paper we describe a falls detection and classification algorithm for discriminating falls from daily life activities using a MEMS accelerometer. The algorithm is based on a shallow Neural Network with three hidden layers, used as fall/non fall classifier, trained with daily life activities features and fall features. The novelty of this algorithm is that synthetic falls are generated as multivariate random Gaussian features, so only real daily life features must be collected during some day of normal living. Moreover, the features related to synthetic fall events are generated as complement of normal features. First of all, the features acquired during daily life are clustered by Principal Component Analysis and no Fall activities shall be recorded. The complement set of the normal features is found and used as a mask for Monte Carlo generation of synthetic fall. The two feature sets, namely the features recorded from daily life activities and those artificially generated are used to train the Neural Network. This approach is suitable for a practical utilization of a Neural Network based fall detection characterized by high Recall-Precision rate.

1 Introduction

The detection of falls of the elderly and people with diseases like epilepsy or Parkinson or simple people with motor difficulties, is today a problem of great public interest. This generated a wide range of research and led to the development of various falls detection and tele-monitoring systems to allow prompt intervention when a fall occurs. Studies in the past years have shown that 1/3 of Senior citizens over age 65 ([25]) are often victims of undetected falls and most of their injuries are due to a lack of intervention. Generally speaking the classification between daily life activities and falls is a difficult task because many daily life activities look like fall (for example running, sitting in a car or lying in bed) and many falls may look like daily life activities. All the types of errors of these systems are of great

importance. In case of False Positive error, users are not motivated to use the falls detection system because in many normal life activities are wrongly detected as falls. In those cases, the operator soon gets tired of the false alarms. On the other hand, in case of False Negatives errors it happens that the system leads to lack of interventions in case of fall. Of course this situation is followed by problems of serious injury and also of mortality.

Generally speaking, while it is quite simple to gather accelerometer data during Normal Daily Living (Daily Life Activities or DLA), it is very difficult if not impossible to collect data during Falls, so the question is: how can the false positive rate be reduced if enough fall data is not available? The answer to this question is that we produce synthetic fall data starting from data collected during normal daily living.

Our approach reduces the amount of False Positives compared to threshold based systems by performing extensive training of a neural network if large quantities of features are gathered during normal life activities.

In the last decade, there has been a great deal of research that has examined the use of inertial sensors such as accelerometers and/or gyroscopes to realize systems for automatic detection of falls. The goal of these systems is to detect the falls of patients who have any difficulty in walking to quickly alert the operators who provide a suitable assistance. The characteristics these systems must have are on one side low cost, consumption and size, and on the other side high performance. While the first characteristics are always better met, the performance is still unsatisfactory. Typically, performance is measured in false positives and false negatives. False positives, if excessive, could demotivate the use of these systems because the recipient of a fall warning could get tired of it. Clearly, false negatives are the most dangerous for the patient's health because they correspond to missed falls. Normally these errors are measured by Sensitivity (ability to detect actual falls) and Specificity (ability to avoid false positives) or equivalently Recall and Precision. From the literature, the existing technology for falls detection systems can be roughly classified into three

categories.

- With wearable sensors: These systems use a triaxial accelerometer or gyroscope or a combination of both to estimate the posture of the subject's body. The sensors are placed in different places, such as the waist, the thigh, the wrist, the shoes. Many systems use the smart-phones inertial sensors.
- With environmental sensors: The environmental sensors are nothing more than sensors positioned around the subject. Floor sensors such as pressure sensors on the mat, microphones, infrared sensors, microwave motion detectors are used to detect the fall. The classifiers of neural networks are used to classify daily activities.
- With image processing: In such methods, a camera is used to monitor body postures. Falls are detected using various image processing techniques such as pattern matching, posture recognition, skeleton extraction, background subtraction, optical flow processing, etc.

A fall can be described as the rapid change from standing or sitting towards an elongated position in earth or almost elongated [24]. This definition has been used in many studies. The paper is organized as follows. In Section 2 we describe the state of the art. In Section 3 we provide the main features of the accelerometer. In Section 4 we introduce preliminary definitions of our work. In Section 5 we describe our main approach. In Section 6 we provide the proposed metrics for performance evaluation. In Section 7 we provide the experimental evaluation and assessment of our proposed framework. Finally, in Section 8 we discuss concluding remarks and future work of our research. A preliminary version of this paper appears in the short paper [5].

2 Previous Work

The algorithms used in systems with wearable sensors can be divided into approaches that use thresholds-based heuristics and approaches that use machine learning tools [23]. The latter may be k-Nearest Neighbor (kNN), neural networks, hidden Markov models, or the two classes classification schemes based on the Support Vector Machines (SVM) classifiers. In all cases, however, it is important to have both falls and daily life activity features. In the case that the approaches are based on thresholds, the availability of both types of data is important to find optimal thresholds, while in the other case the data are important for correctly training the machine learning algorithms.

The thresholds-based heuristics approaches are methods that use thresholds and appropriate functions derived from

inertial parameters. The simplest approach to detecting a fall could be to detect the ground position of the person, by means of a horizontal inclination detection sensor. This method is suitable for monitoring "isolated subjects" but less suitable for the detection of falls of an elderly person in his home environment as the hours of sleep are not regular. Therefore this method provides many "false positives". A complementary solution is to detect the person lying on the floor, using floor tiles equipped with sensors. But when the falls do not end on earth, or if the floor does not have these sensors, obviously they are not detectable. When it falls, the person often hits the ground or an obstacle. The "shock impact" causes an intense inversion of the polarity of the acceleration vector in the direction of the trajectory, which can be detected with an accelerometer or a shock detector, which is actually a threshold accelerometer. Although most of the falls occur in the "front" plane (forward or backward), the direction of the trajectory of fall and is obviously variable from one fall to another. Also the position of the sensor on the body relative to the point of impact modifies the signal recorded at the moment of shock. The lack of movement can be used to detect the fall as, after the "serious" fall, in which the person can be seriously injured, they often remain immobilized in one position. A motion / vibration sensor, positioned on the body (e.g., wrist or ankle), can be used or, again more simply, infrared sensors of presence disseminated in the home. The choice of latency time (the delay before the decision) that should be long enough to reduce "false positives", which translates into a longer delay before intervention, represents a critical problem for these approaches. As discussed earlier, during a fall there is a temporary "fall free" period, during which the vertical speed increases linearly over time due to gravitational acceleration. If you measure the vertical speed of the normal movements of the person (getting up, lowering, sitting down), you can discriminate these speeds from what you do during the fall, which would exceed an appropriate threshold. The intrinsic of analytical methods lies in the choice of this threshold, which if too low causes "false positives" and if too high causes "false negatives". Also this threshold differs from subject to subject. Image processing of video signals can also be used to detect one fall identifying the lying posture using analysis of the visual scene or detecting brusque movements using the revelation of movements with respect to the background. The latter method usually consists in subtracting successive images to keep only the variations, which are then sorted according to their direction and / or their width. While these techniques are well established in controlled environments (e.g., laboratory), they must be modified in environments uncontrolled where parameters such as lighting or framing can be arbitrary. Furthermore, if the subject moves in a 3-dimensional space, it may need more complex techniques, namely the

use of 2 cameras ("VisionStereo"). These image techniques are absolutely feasible at present, both technically and economically, thanks to the presence on the market of low cost cameras (web cam), with the possibility to transmit images in wireless mode over short distances and availability of appropriate processing algorithms. However the acceptance of these technologies for images poses big problems of privacy, as it requires the positioning of video cameras in the living space of the person, and in particular in the bedroom and the bathroom.

An alternative to heuristics approaches, machine learning methods can be used to detect falls. These methods are based on the data acquired on the real working system, used in a preliminary training phase. "Supervised" or "unsupervised" classification algorithms can be used. In case of classification algorithms with "supervised" learning, the person wearing the device performs a series of voluntary actions in order to identify the parameters in the normal cases. In the case of "unsupervised" learning, it is possible to record the movements of the person, within a few hours or several days, and then perform a statistical analysis of the measured speed. These approaches to developing fall detection algorithms are based on observing of the data (the training period) and then on the classification. The choice of classification algorithms is very broad. If you use a supervised method, a simpler choice is to train a neural network, which will then be used to automatically classify the signal. Only the situations encountered during training can be recognized, all others can be shuffled into a class called "others" if the algorithm "is" "unsupervised", falls can be isolated if the training period is much longer than the fall event. Furthermore, it is likely that the first event fall is not detected since its class is still unknown before its premiere appearance.

Regarding the thresholds-based heuristics approaches, Bourke *et al.* [2] uses signals from triaxial accelerometers mounted on the trunk and the thigh to distinguish falls from the *Activities of Daily Living* (ADLs). They propose a higher fall threshold (UFT) and a lower fall threshold (LFT) in an attempt to optimize the balance of false positives and false negatives. Likewise, Kangas *et al.* [13] attached a triaxial accelerometer to the waist, wrist and head of volunteers who performed simulated drops and ADLs in laboratory. Their algorithms considered the phases of pre-impact, impact and post-impact of the fall, separately and in combination, and achieved up to 100% of specificity and sensitivity of 95%, using a single sensor mounted at the waist. However, this algorithm has not been tested in real environments. The only study that examined its accuracy in the real world was conducted by Bagala *et al.* [6], which evaluated fall detection methods (including the Bourke and Kangas algorithms described above) using data from real falls, achieving much better results. In laboratory settings,

the development of improved algorithms for automatic fall detection in the elderly requires an understanding of real-life fallout scenarios in older adults and the integration of such information into the design of laboratory experiments. The common fall scenarios are often absent in the majority, if not in all, of the previous laboratory experiments of fall, and the consequent discrepancy in the sensor data is, perhaps, the main cause of the lack of accuracy of the fall detection algorithms, when tested on real scenarios. In [9] a system is described that uses a tri-axial accelerometer and gyroscope. The detection algorithm uses three thresholds: one to recognize pre-fall situations, one to detect the maximum of the acceleration vector module and one to detect the maximum angular velocity. The algorithm described in [10] also uses three thresholds, one for the local minima of the acceleration module and two for the local maxima of the acceleration module and the angular velocity module. Systems based on machine learning algorithms use classifiers that are trained with both ADL and falls data. However, in a realistic context, due to the lack of sufficient availability of falls data and the lack of knowledge and understanding of what could be the falls, approaches based on the detection of anomalies and classification of a single class can be used. These techniques can not identify falls directly because fall data is not available for classifier training. However, they can identify falls indirectly by classifying them as abnormal activities. In these approaches, therefore, abnormal activities are classified as deviations from normal behavior. Naturally, the concept of normal activities must be clearly defined to identify abnormal activities. Moreover, even if the data of normal activities are not sufficient, then these techniques can produce excessive false positives. Recent research projects [4], [22], [16] show that falls can be identified without actually acquiring them. As evidenced by Klenk *et al.* [11], simulated falls differ significantly from real-world falls. Thus, having simulated falls in the training data set could lead to achieve classifiers that show different behaviors with real-world falls. However, other authors such as Zhou and others [21] have presented a method to detect falls using transitions between activities to model falls. Zhou and others trained supervised classifiers using the normal activities collected by a mobile device, then used transitions between these activities to train a One-class Support Vector Machine (OSVM) and showed that it performs better than an OSVM trained only with activities normal. Micucci *et al.* [22] evaluates methods of detecting falls that do not require dropping data during training on different data sets collected using the smart-phone accelerometer. Their results show that in most cases, the One Class k-Nearest approach Neighbor classifier OCNB behaves better or equivalently to supervised SVM and KNN classifiers that require both types of data, i.e. data for normal and abnormal activities. In other words, Micucci *et al.*

use the one-class k-Nearest Neighbor (kNN) classifier and the one-class SVM classifier. These classifiers have been trained only with ADL and FALL instances. If the anomaly score is higher than a given threshold, the new instance is classified as an anomaly / fall, otherwise is classified as an ADL. Micucci et al compares the anomaly detectors with a two classes kNN and a two-classes radial basis SVM. These classifiers have been trained and tested with both instances of ADL and FALL. This is the case that we are looking for in a real scenario. The main contribution of [22] is the discovery that to design an effective method of detection of falls, it is not necessary to acquire data of falls but it is sufficient to classify the test data as anomalous. As for the HMM models, the traditional way to detect unseen abnormal activities appears as a model of normal activity using an HMM, appears the likelihood of a test sequence with the trained models and if it is below a pre-defined threshold then identify it as an anomalous activity [20]. Another common method to detect anomalous activities is to model the normal activities by a common HMM instead of modeling them separately. In [17] two HMM algorithms are presented that are normal HMM, in which the system noise covariance of the normal dynamics is used to determine the region with highest likelihood which are far from normality based on which events can be classified as 'not normal'. Their results show high detection rates for falls on two activity recognition data sets, albeit with an increase in the number of false alarms. In [16], Khan *et al.* experimentally show that this approach can give better results than supervised classification with limited fall data. When the number of fall data increases, the performance of supervised classifiers improves, but falling data collection can take a long time.

3 Accelerometer Features

The output of the MEMs accelerometer are the three component of the acceleration vector according to the three axis x, y, z , namely a_x, a_y, a_z , each of them related to the current time instant. From this signal, many features are extracted, see for example [14, 27]. We first compute the modulus of the acceleration vector, namely $Acc = \sqrt{a_x^2 + a_y^2 + a_z^2}$. Let us give a look to Figure 1 which is the time evolution of Acc for a typical fall. In this case, it is a Fall backward while trying to sit on a chair, taken from Mobifall v.2. It is worth noting that the overall time frame is the typical fall time. Here we choose the following measures: the maximum value of the modulus, labeled as **Peak** in Figure 1, the length between the two arrows, labeled as **Base**, and the modulus of the slopes of the three components within the signal frame. The slope is computed as follows. Calling t_1^i, t_{1+N}^i the first and last time instant of

Figure 1. Features Extracted from the Graph of the Module

the N samples i -th frame, let us consider the values:

$$\begin{aligned} \max_{a_x} &= \max\{a_x(t_1^i), \dots, a_x(t_{1+N}^i)\}, \\ \min_{a_x} &= \min\{a_x(t_1^i), \dots, a_x(t_{1+N}^i)\}, \\ \max_{a_y} &= \max\{a_y(t_1^i), \dots, a_y(t_{1+N}^i)\}, \\ \min_{a_y} &= \min\{a_y(t_1^i), \dots, a_y(t_{1+N}^i)\}, \\ \max_{a_z} &= \max\{a_z(t_1^i), \dots, a_z(t_{1+N}^i)\}, \quad \min_{a_z} = \min\{a_z(t_1^i), \dots, a_z(t_{1+N}^i)\}. \end{aligned}$$

Then, in the interval t_1^i, t_{1+N}^i , the slopes of the three components are: $slope_x = \max_{a_x} - \min_{a_x}$, $slope_y = \max_{a_y} - \min_{a_y}$, $slope_z = \max_{a_z} - \min_{a_z}$. The modulus of the slope component is

$$Slope = \sqrt{slope_x^2 + slope_y^2 + slope_z^2} \quad (1)$$

Another feature we use is the Ratio of the Peak over Base, as described in (2)

$$Ratio = \frac{Peak}{Base} \quad (2)$$

These features have been chosen because they require very low computation, and so they can be used also on embedded processors with very little computational power.

4 Preliminary Definitions

We now make some preliminary definition useful in Section 5. Assume we use N features describing Falls and ADL. Then, consider the following definitions.

Definition 1 The Features Space (FsS) is an hyper-cuboid with 2^N vertices and $2 \cdot N$ sides where all the original features points lie.

Calling $\max(feature_i)$, and $\min(feature_i)$ respectively the maximum and minimum values the i -th feature can reach in the current case, the length of the first side of the hypercuboid is $\max(feature_1) - \min(feature_1)$, of the second side is $\max(feature_2) - \min(feature_2)$ and so forth. The two vertices V_1 and V_2 with respectively the minimum and maximum Euclidean distance from the origin have coordinates $V_1 = (\min(feature_1), \min(feature_2), \dots, \min(feature_N))$ and $V_2 = (\max(feature_1), \max(feature_2), \dots, \max(feature_N))$ respectively. Of course, if we have only two features, FsS is a rectangle and if we have three features, FsS is a 3D cuboid.

Definition 2 $DLAS \in FsF$ is a set whose elements are vectors of features collected in one or more days of Daily Living Activities (DLA).

Definition 3 $FAS \in FsF$ is a set whose elements are vectors of features collected during Fall Events.

In Figure (2) we report an example of FsF , $DLAS$, FAS and vertices V_1 , V_2 for two and three features respectively.

Definition 4 $SynFS \in FsF$ is a set whose elements are features representing Synthetic Falls, and C_{SFS} is its Cardinality, i.e. the number of its elements. The contours of such sets will be used as Masks in Monte Carlo synthetic generation of falls.

As stated before, our assumption is that $SynFS$ can be viewed approximately as a complement to $ADLS$, provided that certain conditions are verified. $SynFS$ is represented by an N -dimensional sphere with center in V_2 , which is the vertices of FsS most distant from the origin. The radius of SFS , initially equal to zero, is found with an iterative approach which increase its value until the desired number of elements of $DALS$ are included in it. In other words, $SynFS$ is defined as follows:

$$SynFS = \{z | z \in DLAS \cup \{\text{some elements of } DLAS\} \wedge C_{SFS} \leq \gamma\} \quad (3)$$

Let us look at the Figure 2 and Figure 3. Here we assume that only two features are used, so the sets can be drawn as a 2D plane. The points represented with squares are ADL elements. The circles' boundaries are related to some SFS with different Cardinalities and define two Masks which will be noted as $Mask_{Cardinality}$.

We recall now the definition of a useful operator from Binary Morphology [15, 19].

Figure 2. Example of 2D and 3D Features Obtained in Daily Living Activities and Fall Events

Definition 5 The Delation of a given matrix A by a structuring element B , represented with the $\oplus$ symbol is defined as follows: $A \oplus B = \{x | B \cap A \neq \emptyset\}$.

5 Description of the Approach

Our goal is to generate synthetic falls by using a Monte Carlo algorithm. In other words N -dimensional random vectors from Gaussian Distribution are generated and filtered by the Masks described above so that only the vectors falling within the Mask are retained. In order to reduce the computational complexity of the Monte Carlo algorithm, binary operations are performed.

First of all, Principal Component Analysis [1, 12] of DLAS data is performed to cluster DLA data. To clarify the

Figure 3. Example of Two Synthetic Fall Sets with Different Cardinality

algorithm description let us assume that only two features are used. Using the data of Figure 2 by PCA we approximate the shape of the data with the ellipse depicted in Figure 4. It is worth observing that the major and minor axis of the

Figure 4. Shape of the Data Captured by PCA

ellipse are the first and second components of the data.

The binary operation starts by a binary version of the PCA ellipse, which is simply a projection of the PCA ellipse on a 100×100 binary matrix. It is worth noting that here we make use of binary data for complexity reduction. The ellipse is filled with ones. This is shown in Figure 5, where the binary matrix corresponding to Figure 4 is depicted. The set whose elements are the pixels inside the binary ellipse is called BE for binary ellipse. Then, a mask is generated

Figure 5. PCA Shape as a Binary Matrix

with Morphological Delation using n -dimensional polytope structuring elements. It is worth noting that we use delation instead of erosion because the shapes are easier to binarize. In the 2D example we use an octagon. Starting from an octagon in V_2 , a rough approximation of the spherical boundaries shown in Figure 2 are obtained using the following iteration: $mask = mask \oplus octagon$. Let M be the set of pixels set to one in the mask. Then we estimate the number of pixels of the intersection between the mask and the binary ellipse by performing the set operation $Len(M \cap BE)$. Now we introduce the threshold γ such that the number of pixels in the intersection between mask and binary ellipse be $\leq \gamma$. A loop is performed such as:

$$\text{while } Len(M \cap BE) \leq \gamma \text{ then } M = M \oplus B \quad (4)$$

where B is an octagon structuring element. In the upper panel of Figure 6, a sequence of Masks (the black surface) for different values of γ is reported. The panel at the bottom

Figure 6. Portion of the Sequence of Masks for Increasing Values of γ

of Figure 6 shows the corresponding result of the set operation $M \cap E$. The number of pixels of $M \cap E$ increases until the area is greater than the threshold γ .

Synthetic falls, finally, are generated as random vectors $X = [X_1 \dots X_n]^T$, where $X_1 = \text{feature}_1, X_2 = \text{feature}_2, \dots, X_N = \text{feature}_N$, according to the Gaussian multidimensional probability distribution $N - Gauss$ reported in (5).

$$p(x; \mu, \Sigma^2) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) \right) \quad (5)$$

In 5 the mean is estimated as

$$\mu = \left[\frac{\max(\text{feature}_1) + \min(\text{feature}_1)}{2}, \dots, \frac{\max(\text{feature}_N) + \min(\text{feature}_N)}{2} \right] \quad (6)$$

so $\mu \in \mathbf{R}^n$. Moreover, in (5) the covariance matrix is an $N \times N$ symmetric matrix. The diagonal elements are the variances of each feature. It is estimated as follows:

$$\Sigma(i, i) = \frac{(\max(\text{feature}_i) - \min(\text{feature}_i))^2}{4}. \quad (7)$$

All the other elements are equal to zero. In Figure 7 we report an example of random generation of features according to the bi-variate Gaussian distribution.

Figure 7. Example of Random Bi-Variate Gaussian Generation

Finally we evaluate the synthetic fall features as intersection of the random bi-variate features with the feature Space and the binary Mask. The synthetic falls features are reported in the example shown in Figure 8 where also the real fall features are reported for a first comparison.

In Figure 9 we recall that the neural network has input data derived from the sensors. Then we have three hidden layers and one exit layer. The dimension of this network allows to perform the network training using usual back-propagation.

The key point of our approach is the following. The neural network is trained with three features, namely Ratio,

Figure 8. Example of Synthetic Fall Features

Figure 9. Neural Network Structure Trained with Real DLA and Synthetic Falls

Base and Peak1 extracted from ADL and falls. The DAL output of the network is set to zero during daily living features and the FALL output is set to one for synthetic falls features. When a real fall happens, the network should be able to detect it by looking which output is greater than the other.

It is worth recalling that the problem with training such

neural network classifier is, in practice, we have only real ADL features and it is not easy to have real falls. To solve this difficulty, we generate features which are classified as Falls. In other words, we generate features which are points in the set that is complement to the set of the DAL set.

6 Metrics for Performance Evaluation

The starting point for measuring the quality of a classifier is to obtain the rate of false positive (fp), false negative (fn), true positive (tp) and true negative (tn) from the classifier. In our case of a falls detector, let us suppose that there is a fall. If the detector detects it, a **tp** is measured. If it does not detect it, we have a **fn**. For example, on 100 actual fall events, the detector could have 80 **tp** and 20 **fn**. If there has not been a fall, the detector could say that there was a fall (**fp**) or that there was no fall (**tn**). So on 100 non-fall events, we could have 80 **tn** and 20 **fp**.

In other words:

- TP (true positive): This is a situation in which a fall occurs and the system correctly detects it.
- FN (false negative): In this situation we have the fall happens, but the device does not announce it.
- TN (true negative): This is the situation in which a fall does not occur and the system correctly detects that there has not been a fall.
- FP (false positive): In this situation the fall does not happen but the device incorrectly announces that it has detected a fall.

These measures are also called:

tp → **hit** **fn** → **correct rejection**
fp → **false alarm**
tn → **miss**

It is sometimes convenient to measure errors in a more concise way. The most used measures are:

$$Precision = \frac{tp}{tp + fp}$$

This parameter measures the following quantity: the proportion of positive responses that have really fallen.

$$Recall = \frac{tp}{tp + fn}$$

This parameter represents the system's ability to detect a fall every time it occurs. The algorithm is good if the recall approaches 1 because in this case there are no false negatives. In other words, this parameter measures the proportion of falls that have been correctly identified.

7 Experimental Results

First we give a look to the data sets used for experiments. The first is the MobiAct v2.0 data set [7]. MobiAct contains data of four different types of falls and nine different daily living activities from a total of 57 subjects with more than 2500 trials. As well as being used to obtain experimental results, in this paper MobiAct is used as representative data set in all the Figures.

The second is UMAFall, which contains data from 17 subjects performing 8 different types of ADL and 3 different types of falls. Sensing point of MobiAct and UMAFall is the right trouser pocket.

The third data set used for experimental result is the Sis-Fall [8, 26], selecting data related to the waist sensor point. It was generated with 38 participants performing repetitions of 19 ADL and 15 fall types.

We first obtain False Positive, and Negative as well as True positive and True Negative. All the results reported in the following are averaged over these data sets. Of course as usual we try an input signal from the test section of the data sets and we look if the output is correct or not. These results are reported in Figure 10 and in Figure 11.

Figure 10. False Positive and False Negative Obtained with Real ADL and Falls Data

Then we obtain the values of Recall and Precision, reported in Figure 12.

These performances have been compared with the Bourke algorithm [2], which is three threshold based. To find the optimum values of the thresholds, we noticed that the most important threshold is the third one, so we found its performances at various values of the third threshold. The results are reported in Figure 13 which shows that the val-

Figure 11. True Positive and True Negative Obtained with Real ADL and Falls Data

Figure 12. Recall and Precision Obtained with Real ADL and falls Data

ues of Recall/Precision are quite lower than our algorithm.

8 Final Remarks and Future work

Many fall detection systems are based on thresholds applied on features derived from inertial sensors. In this paper we report a novel algorithm which is able to achieve high performance, namely an equal Recall and Precision value more of 90% and a false error rate in this point less than

Figure 13. Recall and Precision Performances of the Bourke Algorithm for Various Values of the Third Threshold

8%. The main feature of this algorithm are that it only requires collecting features in periods of normal daily living and then, from these features, it estimates features of artificial falls. The availability of many features which describe normal and fall events allow to use Machine Learning approaches which are very powerful classifiers provided that sufficient amount of data is given. An important aspects of the described approach is that we use only one type of inertial sensor, namely the accelerometer. Future work will be focused on the fusion of the described results obtained with only an accelerometer with other types of inertial sensors, for example a gyroscope or a magnetometer. In this way many other data could be given to the neural network. The computation complexity is very low because it needs only to compute a trained neural network, so it can be performed in real time. Moreover, we used only three features. It would be interesting to see how much the performance increase if other features are used, hence leading to a feature hyperspace as indicated in Section 3. Another possible direction of research consists in making our algorithm compliant with emerging features of novel big data systems (e.g., [18, 28, 3, 29]).

References

- [1] H. Abdi and L. J. Williams. Principal component analysis. *WIREs Comput. Stat.*, 2(4):433–459, July 2010.
- [2] A. Bourke and G. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med Eng Phys.*, 30(1):84–90, 2008.

- [3] P. Braun, J. J. Cameron, A. Cuzzocrea, F. Jiang, and C. K. Leung. Effectively and efficiently mining frequent patterns from dense graph streams on disk. In *18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2014, Gdynia, Poland, 15-17 September 2014*, pages 338–347, 2014.
- [4] M. C, I. R, P. I, and C. M. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE*, 2014.
- [5] A. Cuzzocrea, E. Mumolo, and M. Tassarotto. Towards an effective and efficient machine-learning-based framework for supporting event detection in complex environments. In *Proceedings of the 43rd IEEE Computer Society Signature Conference on Computers, Software and Applications, COMPSAC 2019, Milwaukee, WI, USA, July 15-19, 2019*, 2019.
- [6] B. F, B. C, C. A, C. L, A. K, H. JM, Z. W, and K. J. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS One*, 7(5):1–9, 2012.
- [7] V. G., C. C., M. T., P. M., and M. Tsiknakis. The mobiaet dataset: Recognition of activities of daily living using smartphones. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, pages 143–151, 2016.
- [8] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi. A depth-based fall detection system using a kinect[®] sensor. *Sensors*, 14(2):2756–2775, 2014.
- [9] H. W. Guo, Y. T. Hsieh, Y. S. Huang, J. C. Chien, K. Haraikawa, and J. S. Shieh. A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope. In *Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2015 International Conference on*, pages 54–57. IEEE, 2015.
- [10] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian, and B. Q. Tran. Optimization of an accelerometer and gyroscope-based fall detection algorithm. *J. Sensors*, 2015.
- [11] K. J, B. C, L. F, N. S, M. W, A. W, Z. W, H. JM, van Lummel RC, C. L, and L. U. Comparison of acceleration signals of simulated and real-world backward falls. *Medical Engineering & Physics*, pages 368–373, 2011.
- [12] T. T. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences*, pages 1–16, 2016.
- [13] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jms. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture*, 28(2):285–291, 2008.
- [14] A. O. KANSIZ, M. A. GUVENSAN, and H. I. TURKMEN. Selection of time-domain features for fall detection based on supervised learning. In *Proceedings of the World Congress on Engineering and Computer Science 2013 Vol II*, pages 1–6, 2013.
- [15] L. R. Kennell, R. W. Ives, and R. M. Gaunt. Binary morphology and local statistics applied to iris segmentation for recognition. In *ICIP*, pages 293–296. IEEE, 2006.
- [16] S. S. Khan, M. E. Karg, D. Kulic, and J. Hoey. Towards the detection of unusual temporal events during activities using hmms. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 1075–1084, 2012.
- [17] S. S. Khan, M. E. Karg, D. Kulic, and J. Hoey. X-factor hmms for detecting falls in the absence of fall-specific training data. In *Ambient Assisted Living and Daily Activities - 6th International Work-Conference, IWAAL 2014, Belfast, UK, December 2-5, 2014. Proceedings*, pages 1–9, 2014.
- [18] K. Li, H. Jiang, L. T. Yang, and A. Cuzzocrea, editors. *Big Data - Algorithms, Analytics, and Applications*. Chapman and Hall/CRC, 2015.
- [19] S. Z. Li and A. K. Jain, editors. *Encyclopedia of Biometrics*. Springer US, 2009.
- [20] S. Lühr, S. Venkatesh, G. A. W. West, and H. H. Bui. Explicit state duration HMM for abnormality detection in sequences of human activity. In *PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004. Proceedings*, pages 983–984, 2004.
- [21] Z. M., W. S., C. Y., C. Z., and Z. Z. An activity transition based fall detection model on mobile devices. In: *Park J., Jin Q., Sang-soo Yeo M., Hu B. (eds) Human Centric Technology and Service in Smart Space. Lecture Notes in Electrical Engineering*, vol 182. Springer, Dordrecht, 2012.
- [22] D. Micucci, M. Mobilio, P. Napolitano, and F. Tisato. Falls as anomalies? an experimental evaluation using smartphone accelerometer data. *Journal of Ambient Intelligence and Humanized Computing*, pages 87–99, 2017.
- [23] V. Mirchevska, M. Lustrek, and M. Gams. Combining domain knowledge and machine learning for robust fall detection. *Expert Systems*, 31(2):163–175, 2014.
- [24] N. Noury, P. Rumeau, A. Bourke, G. Laighin, and J. Lundy. A proposal for the classification and evaluation of fall detectors. *IRBM*, 29(6):340 – 349, 2008.
- [25] T. Shi, X. Sun, Z. Xia, L. Chen, and J. Liu. Fall detection algorithm based on triaxial accelerometer and magnetometer, vol. 24, no.2, pp157-163, 201. *Engineering Letters*, 24(2):157–163, 2016.
- [26] A. Sucerquia and J. D. Lpez. Sisfall: A fall and movement dataset. *Sensors*, pages 1–14, 2016.
- [27] S. Wu and Z. Wang. Applying online feature selection for fall detection. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–5, 2016.
- [28] Z. Wu, W. Yin, J. Cao, G. Xu, and A. Cuzzocrea. Community detection in multi-relational social networks. In *Web Information Systems Engineering - WISE 2013 - 14th International Conference, Nanjing, China, October 13-15, 2013. Proceedings, Part II*, pages 43–56, 2013.
- [29] C. Yang, J. Liu, C. Hsu, and W. Chou. On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *The Journal of Supercomputing*, 69(3):1103–1122, 2014.

Exploiting Coordinated Views for Scholarly Reading and Analysis

Francesco Poggi

**Dept. of Computer Science and Engineering*
University of Bologna, Italy
francesco.poggi5@unibo.it

Paolo Ciancarini

**Dept. of Computer Science and Eng.*
Universities of Bologna and Innopolis
paolo.ciancarini@unibo.it

Angelo Di Iorio

**Dept. of Computer Science and Eng.*
University of Bologna, Italy
angelo.diiorio@unibo.it

Silvio Peroni

†Dept. of Classical Philology and Italian Studies
University of Bologna, Italy
silvio.peroni@unibo.it

Fabio Vitali

**Dept. of Computer Science and Eng.*
University of Bologna, Italy
fabio.vitali@unibo.it

Abstract— There are several ways of reading scientific papers, and in general, documents. The sequential reading of the text flow can be complemented with alternative visualizations such as tree-based views, maps and charts. Infoview techniques are very promising in this context but not yet fully exploited. This paper presents a system called DocuDipity that is able to smoothly integrate infoview techniques, in particular a SunBurst visualization and a flow text reader, in a coordinated environment that helps researchers to read and analyze scientific articles. We discuss some end-user tests that showed that our tool is highly effective to find patterns and disclose some habits in writing articles. For instance, we used DocuDipity to study how the authors organise their content and use citations. The paper presents both the system and the details of the evaluation, together with some indications on future developments.

Index Terms—Document visualization, information interfaces and presentation, scholarly articles, structural patterns, sunburst

I. INTRODUCTION

In the last three decades, the advent of the World Wide Web and the consequent development of hypermedia content has produced a profound impact in the way people access, use and consume information. This fundamental shift has also involved the way researchers access scholarly information. For instance, the results of a study based on surveys performed in time span of more than 30 years show that the average number of readings per year per science faculty member continues to increase, while the average time spent per reading is decreasing[1][2]. The fact that the amount of time available for reading scientific articles is likely reaching a maximum capacity poses challenging issues.

Things are even more complex as scholars read documents in many different ways, depending on the goal they want to achieve and the time they can spend, our assumption is that they need different tools.

In some cases scholars need to scan documents thoroughly, nitpicking every single word and character; in others, a bird's eye view is enough: they browse pages, searching relevant

parts, or they jump to specific sections, starting from the table of content. Consider for instance the citations. Incoming citations are checked for evaluation purposes: we just count them to measure the impact of a paper, and, as a consequence, of a given researcher. When we review a paper, instead, we also look at outgoing citations so as to evaluate if it is up-to-date and fits a given research area. We might also scan self-citations in order to get indications about the relevance and impact of that work. It is also common to identify the citation context (the sentence in which a work is cited) or the citation function (the reason why a work is cited) to better characterise and weight citations. A full analysis of the citations requires researchers to *scan* documents. To do so, the *readers* jump from the text of the paper to the bibliography, alternating *focused readings* to *overview readings*, and moving from one part to another.

The heterogeneity of these *reading patterns* is the main focus of this work. We limit the concept to the definition of Hornbæk et al. in [3]: "how readers navigate and manipulate documents as they try to accomplish their aims with reading."

Our results are directly connected to those of [3]. The authors investigated the support provided by a representative set of visualization tools to scholars in their reading and analysis activities. Three interfaces were compared in the study: a linear interface (where a document is shown as a linear sequence of text and pictures), a fisheye interface (where the document parts considered of minor importance are initially distorted below readable size, but can be expanded and made readable if the user clicks on them with the mouse) and an overview+detail interface (where a linear interface called detail pane is complemented on a side by an overview of the document, that can be used to navigate the document - by moving the viewport of the detail pane).

The users of the experiment were asked to perform two types of tasks (i.e. writing an essay and answering questions) on the same set of documents using the three aforementioned interfaces. Experimental results show that the linear interface is in many ways inferior with respect to other ones: it has the

lowest satisfaction and usability scores, and leads to lower essay grades. The fisheye interface is the fastest interface, but subjects using it obtained lower incidental learning scores, resulting in outcomes of lower quality. The conclusion is that the fisheye interface is more appropriate for tasks where a detailed understanding of documents is not the main aim (e.g. relevance judgements in information retrieval systems). The overview+detail interface led to the highest quality outcomes, and subjects strongly preferred this interface. However it was slow for question-answering tasks. The authors explain this fact with the overall higher affordance of this interface, which engages users and encourages them to perform deeper analyses. In fact, the study of the experiment logs showed that users spent more time to further explore the document after the first contact, and also that - on average - these explorations took more time than in the other interfaces. In other words, the frequency and duration of further explorations are higher than those of the counterparts, increasing the quality of the results.

The basic idea of our work is to improve the overview+detail approach by combining *coordinated views* so as to help researchers to discover traits of the scientific articles, **without necessarily reading their full content**. The system we present here, called DocuDipity², combines the hypertext flow layout of the article, in which readers can scroll the sequential content of the paper and read it, with a novel way of displaying tree-based documents, in particular XML ones, based on SunBurst visualization [5]. SunBurst is a form of radial visualization designed for depicting and summarizing in a very compact form even complex information hierarchies. Initially proposed to surf filesystems, SunBurst tools have been applied to various contexts, e.g. for summarizing navigation paths [6] or showing firewall configurations [7]. However, to the best of our knowledge, it has not been used in document-centric systems as the one we propose in this paper.

The main contribution of this work is, in fact, to demonstrate how the SunBurst visualization can be successfully combined with the flow layout in order to help scholars in their tasks. Our goal is not only to confirm the exploitability of the SunBurst but also the strengths of *coordinating* these two views in a single integrated environment that allows readers to easily access different aspects at different levels of the same paper. DocuDipity is also highly customizable, since it relies on editable rules that can be shared and refined by the users. This paper presents a set of rules, but the system is totally independent from them, and can be extended to carry out other analyses.

This paper describes the system and some experiments in either using the pre-loaded rules or writing new ones. We tested DocuDipity with 25 users studying the articles' internal

structures and hierarchical organization, the self-citations, and some writing styles and authoring habits. The results of the test proves that DocuDipity is effective and usable. Some limitations still exist when editing rules but the overall users' perception was very good, as detailed in the experiments section.

The rest of the paper is organized as follows. Section II describes some related works, focusing on similar visualization techniques. Section III presents our solution in detail. Section IV describes how DocuDipity, and in particular its rule-based engine, can be used to support document readings and analysis, together with some examples of applications and findings. The experimental evaluation is presented in Section V, while some discussions and conclusions are in Section VI.

II. RELATED WORKS

One of the most popular principle in visual analytics tools is to provide an overview of the entire information space, so as to give users the context to support their browsing and search tasks[8]. This principle is at the base of many visual analytics systems such as Jigsaw [9], a popular tool that provides coordinate views for investigating concepts (e.g. places, persons, dates, etc.) extracted from text reports. A good overview about these technique is provided by [10]. Only a few works explored the application of this principle for supporting non-linear reading paths in large document collections. Most of them focused on supporting the navigation between documents in large document collections, or searches within a set of predefined keyword/features [11] [12] [13].

Another relevant issue concerns how to visualize the logical structure of a document. This problem may be reduced to the well-known problem of visualizing hierarchies. In times where typical data sizes grow much faster than the available screen sizes, a first element to consider is the efficient use of space. For this reason, implicit hierarchical visualizations (i.e. those that resort to an implicit representation of parent-child relations by positional encodings of the hierarchy items) must be preferred to explicit techniques (i.e. those that explicitly show parent-child relations as straight arcs or lines) [14].

One of the first and more popular alternatives is the Treemap [15], a space-filling slice-and-dice technique based on a rectangular layout. In a treemap, each element of a tree is depicted by a rectangle, which is then tiled with nested rectangles representing sub-branching. The color and area of each item correspond to attributes of the item as well: for example, these visual variables may be used to encode the type of the element and the number of contained characters, respectively. A plethora of variations and improvements of the initial treemap techniques have been proposed over the years in the literature (e.g. 3D Treemap [16], Triangular Aggregated Treemap [17], Quantum Treemap [18], Cascaded Treemap [19], etc.). For representing document information, one of the most suitable alternatives is specific kind of Treemap named Ordered Treemap [20], since it has the property of preserving the order within the document hierarchy.

²The name DocuDipity stresses on the *serendipity* of the system and its ability to make unexpected behaviors emerge, following the Kingrey's definition of *information seeking*[4]: "information seeking involves the search, retrieval, recognition, and application of meaningful content. This search may be explicit or implicit, the retrieval may be the result of specific strategies or serendipity, the resulting information may be embraced or rejected, ... and there may be a million other potential results."

The SunBurst technique [5], which we adopted in our tool, is another relevant alternative based on a space-filling visualization approach that uses a radial layout to represent tree-like structures. In SunBurst, items in the hierarchy are laid out radially, with the root element at the center and deeper levels further away from the center. Comparing Treemap and SunBurst, the former has a longer learning curve and a less explicit portrayal of the hierarchy structure, as described in [6]. Apart from these two techniques, which are the most relevant, a wealth of implicit tree visualizations have been proposed in the last 30 years. A summary of these methods and approaches are described in [21].

Widening the scope of the analysis without limiting to implicit techniques, a quite complete catalog³ of tree visualization techniques is presented in [22]. Two of the most notable examples that apply such a tree-based visualization to documents are the Cone Tree [23] and Hyperbolic Tree [24]. Cone Tree is one of the first focus+context technique that embeds the tree in a three dimensional space. An expensive 3D animation support is required, since the tree has joints that can be rotated to bring different parts of the tree into focus. Another limitation of Cone Tree is its scalability: in fact, trees with more than 1000 nodes are difficult to manipulate. The Hyperbolic tree is the two dimensional counterpart of Cone Tree. The basic idea is to use a radial layout, and put the nodes in focus in the center, while out of focus ones are compressed near the boundaries. A better use of space, and the relatively modest computational needs, makes Hyperbolic tree potentially useful on a broad variety of platforms.

More recently, document visualization research has focused on the development of tools and techniques to support specific task relevant to document analysis. Usually, these tools require a pre-analysis of text to extract and compute a set of relevant features, that are given as input to an engine that produces a visual representation of the document based on them. For example, in [25] Kleim et al. used a pixel-based technique, which they call “literature fingerprinting”, to understand and visualize document signatures, such as vocabulary richness and sentence length. Three coordinated views of document are provided to explore the feature values of text corpora at different levels of detail. Another notable example is DocuBurst [26], which infers word relationships within documents by traversing the WordNet hypernym graph, and applies a radial layout to show the word hierarchy.

III. DOCUDIPITY

In this section we describe DocuDipity⁴, the interactive web-based tool we developed to support the exploration and analysis of heterogeneous document collections.

The DocuDipity interface is composed of four main parts, as shown in Figure 1. The navigation bar on the top lets users select the document to investigate, whose content is presented in the two different and coordinated forms in the central area.

³The interested reader can find an interactive catalog of the tree visualization techniques at the address <http://treevis.net>.

⁴DocuDipity is available online at <http://eelst.cs.unibo.it/docudipity2>

On the left, a view based on the SunBurst technique provides a summarization of the document structure, while on the right side the content of the document is displayed as an hypertext.

Finally, users can select coloration rules to highlight relevant features and patterns in the document under investigation. Each rule changes the color of some selected elements in the document. users can modify or create their custom rules to support personal investigations, and interactively evaluating them on-the-fly. Rules are written in Javascript and CSS and will be briefly described below.

It is worth noting that DocuDipity is a general tool that can work on *any* XML document, without requiring any background information about the format documents are written in. In fact, the DocuDipity engine is based on the structural pattern theory[27]. By leveraging the pattern identification algorithm described in [28], DocuDipity is able to extract relevant information about the structure of the set of documents provided as input, and use it (a) to visualise the documents and (b) to provide an analysis framework to inspect their content easily.

A. FullText View

The FullText view provides a classical reading environment where the content of the document is shown as an hypertext. As all the other DocuDipity components and features, also this view is generated through an automatic analysis and segmentation of the document. The model of the document under investigation used by DocuDipity engine is based on the theory of structural patterns [27] and any textual document encoded in XML can be loaded into DocuDipity.

To generate the hypertextual representation of documents, DocuDipity converts XML documents into HTML composed of generic containers, blocks and inlines associated with some JavaScript code and CSS rules. For instance, the Fulltext view highlights the logical structures of the document (e.g. headings, sections, subsections, paragraphs, text fragments, structured data, etc.). References to figures, tables and bibliographic items are converted into hyperlinks, and other elements (e.g. blockquotes, footnotes, etc.) are collapsed and can be expanded on request. DocuDipity also leverages the pattern-based analysis to automatically generate the table of contents shown at the beginning of the document.

B. SunBurst View

DocuDipity uses the SunBurst technique [5] for providing an explicit but at the same time very compact view of the overall structure of documents. In the SunBurst, items in the document hierarchy are laid out radially, with the root element at the center and deeper levels farther away from the center. In addition to containment, the SunBurst view also preserves information about the element order, by starting to draw elements from midnight, and proceeding clockwise.

Other graphical attributes are used to highlight other information about the document under focus. For example, the angle swept out by each element corresponds to the number of characters contained, even recursively, by it.

Fig. 1: An overview of the DocuDipity interface. The user's analysis starts with the selection of the document to investigate (top area), whose content is displayed with the SunBurst technique (center left) and as an hypertext (top right). In the area in the bottom, users can select predefined coloration rules to highlight relevant document features, or edit new ones and interactively evaluate them on-the-fly.

The SunBurst view has been chosen among the others representing tree structures (e.g. direct graphs, treemaps, etc.) for two main reasons: (i) because it is a space-filling technique that provides a very compact summarization of the document structure, and (ii) because it is more effective and easy to learn (i.e. it has a shorter learning curve) than the other main alternatives [6].

The default coloration rule highlights the most relevant components of the document (i.e. sections, bibliography, metadata, paragraphs, citations, etc.). The document in Figure 1 for example, after an initial metadata section (in orange) contains nine sections (in blue: three quite short at the beginning, followed by two longer in the middle of the document, and four very short at the end), and a quite long bibliography (in pink). No citations are contained in the second part of the

document (they are spread equally in the first four sections, plus two at the beginning of the fifth one). Users can switch between rules to analyze different aspects of the document, looking at the SunBurst view.

C. Coordinated Behaviours

The SunBurst view and the FullText view are coordinated during the user's investigation: when he/she clicks on an element in the SunBurst view, the FullText view scrolls to the corresponding element. Moreover, when the user hovers the mouse cursor over a text fragment in the hypertext view on the right, both (i) the text fragment in the hypertext view and (ii) the corresponding element and all its ancestors in the document hierarchy on the left are highlighted; similarly, when a user focuses on an element in the SunBurst view, the cor-

responding element in the hypertext view is highlighted. The coordination between these two views is crucial in DocuDipity, since it facilitates customized and personal investigations of document collections. The SunBurst view provides a compact overview of the document, and users can easily jump into the details of the document (e.g. by reading a text section in the FullText view) without losing the overall context. This is fundamental for supporting readings that are different from sequential ones, providing at the same time a general view and a more detailed one, and allowing users to jump among them.

This behavior is based on the so-called information-seeking mantra "overview first, zoom and filter, then details-on-demand" [29]. The idea is to develop interfaces that give a general context for understanding datasets by summarizing their most salient features (overview), reduce the complexity of the representation by removing extraneous information from the view and allowing for further data organization (zoom and filter), and finally reveal additional information on a point-by-point basis while the user interacts with the visualization (details-on-demand).

An important issue faced during DocuDipity development is that, as the document hierarchy grows in size, many items in the SunBurst view become small and peripheral slices are difficult to distinguish. Also this problem has been solved using interaction by allowing users to zoom in and out: when users double-click on an element, the SunBurst is re-drawn to show only the subtree rooted in that element (zoom in); to move up one step in the hierarchy, users can double-click on the central circle (zoom out).

D. Editing Coloration Rules

In the bottom area DocuDipity provides an editing area where users can define coloration rules that can be applied to the SunBurst view to highlight relevant features and patterns in the document under investigation. To this end, users need a language to translate their hypotheses into (even complex) conditions, and a method to verify their validity. Instead of creating new languages and tools from scratch, we decided to use well-known web technologies like JavaScript and CSS.

By switching on the "Edit" tab, two textual areas are shown in the bottom of the DocuDipity interface, as shown in Figure 2. In the former one, the user can use JavaScript (in particular JQuery 3) to select elements and to assign CSS classes to them. In the latter one, the user can define CSS rules to specify a style for the classes assigned by the JavaScript code. To facilitate the development of new rules, users can also inspect the XML source of the current document by clicking on the "inspect XML" button.

For example, the following excerpt is the JavaScript code for coloring paragraphs according to their length. This code assigns the class `para-short` to short-length paragraphs (i.e. containing less than 400 characters), `para-medium` to medium-length paragraphs (i.e. containing between 400 and 799 characters), and `para-long` to long-length paragraphs (i.e. containing more than 800 characters).

```
1 var limitShort = 400;
2 var limitLong = 800;
3 // Selects all the paragraphs
4 var toFilter = $('para');
5 // evaluates the provided function over
  each paragraph
6 toFilter.each(function() {
7   // Counts the characters contained in
8   // the current paragraph
9   var textLength = $(this).text().length;
10  // Tests if the current paragraph is
11  // short, medium or long, and assigns
12  // the appropriate CSS class
13  if (textLength < limitShort) {
14    $(this).addClass('para-short');
15  } else if (textLength >= limitShort &&
16    textLength < limitLong) {
17    $(this).addClass('para-medium');
18  } else if (textLength >= limitLong) {
19    $(this).addClass('para-long');
20  }
21 }
```

The following CSS rules associate colors to the elements belonging to one of the previously defined classes (i.e. green to short-length paragraphs, yellow to medium-length paragraphs, and red to long-length paragraphs), and paint lightgray all the remaining elements.

```
1 path { fill: lightgray !important; }
2 path.para-short { fill: green !important; }
3 path.para-medium { fill: yellow !important; }
4 path.para-long { fill: red !important; }
```

By clicking on the "view" button, DocuDipity evaluates both the edited JavaScript and CSS fragments over the document under focus, and updates the SunBurst view.

IV. READING AND EXPLORING SCHOLARLY ARTICLES WITH DOCUDIPITY

In order to showcase DocuDipity we deployed a test installation on a collection of 211 papers from a niche computer science conference, called Balisage Markup Conference ⁵, active for more than 20 years, whose collection of papers is publicly available in XML format since 2008. The same installation has been also used for the use-case study described in Section V.

We loaded an initial set of rules covering some common tasks that researchers already perform today by combining different reading patterns. These rules are not meant to be exhaustive but to give readers indications about the potentialities of the tool.

The following coloration rules are currently available:

- *Document overview*: this view highlights the main document components (e.g. sections, metadata section, paragraphs, bibliography, citations, etc.);
- *Citations*: this view help users to study citations networks by highlighting the position of the citations (i.e. reference in the text of the paper pointing to a bibliographic entry);

⁵The complete proceedings of the conference are freely available online at the address <http://balisage.net/Proceedings/index.html>

Fig. 2: The editor panel to define custom coloration rules.

- *Self-citations*: this extends the previous one by distinguishing between generic citations and self-citations (i.e. references in the text the paper pointing to a bibliographic entry authored by one of the authors of the current paper);
- *Paragraphs length*: this view helps users to study different writing styles by using different colors to highlight short, medium and long paragraphs;
- *British vs American English*: this view help users to distinguish the sentences containing American English spellings and peculiarities (e.g. airplane), from the ones containing British English forms (e.g. aeroplane), or a mix of them.

We discuss two coloration rules here but similar considerations can be generalized to all others and to new ones.

A. Citation analysis

Let us consider, first of all, the use of citations. While reading a new paper, citations are primary tools to find related works and to quickly get an overview of the research space around the paper. Though citations are often aggregated in related works section, this is not always the case. In some disciplines authors tend to aggregate citations in the introduction, in some others to scatter them throughout the paper. The SubBurst view is very effective to locate citations and to go the relevant part of the paper. Consider now the review process of an article or a report: reviewers often check, for instance, the number of self-citations or the publication year of the cited paper or even the publication venue. Instead of reading the flat bibliography, a reviewer could exploit the SunBurst view to get this information and then continue reading, even jumping to the citation she/he thinks is more interesting and worth checking.

Given our research interests in citation networks for scholarly communications [30][31], we experimented a few DocuDipity rulesets that illustrate the positioning of citations (`xref` elements in the Balisage XML) within documents, and we found interesting and different patterns in how citations scatter in the sections.

A glimpse to the DocuDipity SunBurst chart can tell us immediately about how the paper is structured. For instance,

we can immediately observe structures where:

- authors use citations to provide the scaffolding and context of their own contribution, by collecting most of them in an initial section before introducing the core contribution, as in Fig. 3a; or
- authors use citations to provide a final comparison of their contribution against similar other ones in the relevant literature by collecting most of them in a section at the end, as in Fig. 3b; or
- authors scatter citations throughout the paper, so as to signify a continuous and meticulous dialogue between the contribution and the literature, as in Fig. 3c; or, even,
- authors (possibly practitioners, rather than scholars, not versed in the habits of scholarly communication?) that provide a rich set of citations in the final references section... totally not cited in the body of the paper, as in Fig. 3d.

B. Content and style analysis

The stylistic and linguistic analysis of an article is another peculiar task of scholars when writing their own articles, or when giving feedback to other colleagues, or when reviewing articles for a conferences or a journal. Such an analysis goes with the actual content reading, which is and has to be a personal activity of the scholar. On the other hand, DocuDipity can be exploited to automatically identify some features of the article under evaluation.

For instance, we could consider the length of individual paragraphs mentioned in the previous Section and color gray the paragraphs that are about average, red the paragraphs that are much longer than average, and green the paragraphs that are way shorter. Is is easy to identify authors whose style uses long, winding, multi-conceptual paragraphs (e.g., in Fig. 4) from authors preferring paragraphs with short individual sentences straight to the point (as the one in Fig. 5).

So what happens if you point the rules to a paper with multiple authors? Maybe you could distinguish the individual contribution of one author having a writing style different from the others, as in Fig. 6.

(a) The DocuDipity visualization of a paper with citations (i.e. purple slices) grouped in the initial part of the document. This writing style is common in the humanities.

(b) The visualisation of a paper with a "Related Works" section at the end (top-left corner of the SubBurst).

(c) A paper with citations scattered throughout the body of the paper. This writing style is common in scientific articles.

(d) The visualization of a paper that contains plenty of bibliographic items in the bibliography section (in pink), none of which is mentioned in the document body. The SunBurst clearly shows an incomplete usage of XML for marking citations.

Fig. 3: Plots of papers where citations highlights different styles and structures

Fig. 4: A paper made of mostly long paragraphs (in red). The paper was written by a non-native English speaker.

Fig. 6: A paper by multiple authors with a striking difference in the length of the paragraphs, possibly pointing out the individual style of one of the authors. The introduction (shown in the top-right corner) was in fact written by an author, who only worked on that section and whose style is more verbose than that of the others'.

Fig. 5: A paper made of mostly short paragraphs (in green). The paper was written by an English native speaker, and is fluid and easy to read.

Fig. 7: A paper with a mix of American (in red) and British (in blue) words within – the inner circle are the first level section of the paper. The colour green is used when a section and/or paragraph has been written according to both UK and US English.

Or maybe, you could find yourself examining a multi-author paper by non-native English speakers, one of whose authors takes pride in his English, learnt during his post-doc station at the Oxford University, while the others have gotten theirs from the usual mish-mash of styles gained from magazines, blogs, novels, and Hollywood movies, plus the peculiar international English of scientific literature. Point DocuDipity over that paper, add rules that distinguish American English versus British English spelling and peculiarities, such as the *-ise/-ize* endings⁶, and presto: you can immediately find out which sections were written by one individual and which by the others, as in Fig. 7.

V. EVALUATION

We performed some end-user tests in order to evaluate DocuDipity and to identify its main strengths and weaknesses. Following the ISO 9241 standard⁷, we measured the system in terms of *efficacy*, *efficiency* and *satisfaction*. The tests involved 25 users in total, with different backgrounds and expertises, and was split in two parts. In the first test we measured how DocuDipity can support researchers to explore scientific papers by using the rules already available in the system, and involved 20 users. In the second one we asked the other five (more expert) testers to add new rules to DocuDipity and to employ them for further analysis. These two parts are discussed separately in the following subsections.

A. Using DocuDipity

The first test was further split in two parts. We divided the 20 participants in two groups, each formed by 10 testers equally distributed in terms of professional level and skills. Each group was asked to analyze a set of scientific articles and complete two assignments. The source articles were 184 in total, taken from 7 different editions of the Balisage Markup Conference Series⁸.

The first group was asked to use DocuDipity to complete the tasks, while the second could only use the official Balisage web site (where the articles are freely available). The site contains one web page for each article with a sidebar showing a clickable table of content. It is worth remarking that the two groups were given the same input data and tasks, so as to compare their behaviors.

The real test was preceded by a warm-up session, in which the testers were asked to complete a simple task so as to become confident with the platform they were about to test. In the case of DocuDipity, the users were also shown a 5-minutes demo of the system. After completing the two assignments, the participants were asked to fill some questionnaires about their experience, in the form of closed and open questions.

The assignments are summarized below. Each of them was repeated on three different articles:

⁶Remembering also that some words, such as *devise*, do not have spellings in the two dialects...

⁷<https://www.iso.org/standard/16883.html>

⁸The proceedings of the Balisage Series Conferences are freely available at <http://www.balisage.net/>

TABLE I: Execution time for each subtask.

	T1.1	T1.2	T1.3	T2.1	T2.2	T2.3
DocuDipity	2:38	0:38	1:00	1:15	1:16	1:32
Balisage	4:51	4:22	2:06	5:09	7:02	3:13

TABLE II: The percentage of testers who answered correctly for each subtask.

	T1.1	T1.2	T1.3	T2.1	T2.2	T2.3
DocuDipity	80%	80%	90%	100%	40%	60%
Balisage	40%	10%	30%	20%	0%	10%

T1. Study self-citations: given the title of a paper, the testers were asked to indicate how many self-citations it contains. We expected users to search a paper by title and to inspect both the bibliography section and the citations in the text. Note that the complexity of the task depends on the number of authors of the paper. For instance, if we just look for authors' surnames in the bibliography, in a paper written by dozens of authors (which is not unusual in some disciplines like physics) the task would require to manually repeat the search for each author. Moreover, it would be important to count self-references only once in case of multi-authored papers.

T2. Study length of paragraphs : the testers were asked to find the paragraphs whose length is under/over/in-between a given threshold. They were expected to check the length of the paragraphs by counting characters, manually or with an external tool.

Details of each evaluation are presented below:

1) *Efficiency*: We first measured how much time was needed by the testers to complete the tasks, comparing the users' performance with or without the help of DocuDipity. Table I shows the average execution time of the two tasks. For each task, we report the execution time on three different papers. As expected, a specialized tool like DocuDipity outperformed the plain sequential reading of the Balisage web site content. The difference however is significant and worth remarking.

2) *Efficacy*: We also studied if the testers could complete the tasks correctly, by manually comparing the answers given by each tester to the expected answers. Though some participants failed to complete some tasks, the overall efficacy was very high for DocuDipity. On the other hand, the efficacy of searching the same information on the plain web site was very low. Table II summarizes our results. The difference is evident, even more than we expected.

The efficacy of DocuDipity on assignment T1 (self-citations) was very high on all three papers taken into account. The interpretation of one specific user lowered the three scores of 10%. In fact she/he failed all three questions but in a consistent way: each self-citation was counted twice since the SunBurst visualization, in presence of a self-citation, colored

TABLE III: Comparison of the SUS scores for the tests performed with DocuDipity and Balisage

TestSet	n	Mean	SD	t	df	p	95% CI
DocuDipity	10	81	10.29	-	-	-	-
Balisage	10	64.25	7.17	-	-	-	-
Total	20	68.13	9.04	3.92	16	.0012	7.83 - 26.27

in red both the segment corresponding to the XML element and the segment corresponding to its textual content. For Task 1.1, for instance, the given answer was 4 while the correct one was 2. This suggests us that a clearer coloring scheme is needed for segments corresponding to text fragments. We do not have further elements to speculate on the other wrong answers, given by two other different users.

The results on DocuDipity were less positive for Task 2 (length of the paragraphs). Apart from T2.1, whose percentage of success was 100%, the efficacy goes much lower for T2.2 and T2.3. The reason is that most testers had difficulties in manually counting the number of paragraphs highlighted in the SunBurst. They confirmed us that the visualization was clear but they got confused while counting the SunBurst slices. In fact, as discussed later, most of the testers suggested us to add counters and summaries of the results in the DocuDipity interface. Though not shown in the table, note also that the deviation between the wrong and correct answers was very low. Consider for instance T2.2: the correct answer was 19, while 30% of the users answered 18 and 10% answered 16. This confirms the difficulties that the testers experienced while counting the items. As expected, such difficulties are accentuated without the help of a specialized tool like DocuDipity.

3) *Usability*: After the completion of all assignments, the testers were asked to fill two questionnaires. The first one was a *System Usability Scale (SUS)* [32], a well-known questionnaire used for the perception of the usability of a system. It has the advantage of being technology independent and it is reliable even with a very small sample size [33]. The mean SUS score for DocuDipity was 81 (in a 0 to 100 range). The value shows how the users' satisfaction was very high, confirming that the coordinates views and the SunBurst visualization are well accepted by the testers.

In addition to the main SUS scale, we also examined the sub-scales of pure *Usability* and pure *Learnability* [34]. They gave us a more precise characterization of the users' feedback. The mean values for Usability was 81 and Learnability was 82. This means that the system was perceived both easy to learn and effective in supporting users to perform their tasks.

We also repeated the SUS test for the plain Balisage web site, just to compare this score and the DocuDipity one. In fact, the difference was very high: 64 against 81, which is statistically significant according to an unpaired student t-test ($t = 3.92$, $p < 0.01$) as shown in Table III.

In order to go deeper, we also included four open questions, only to the testers that used DocuDipity:

- What were the most useful features of DocuDipity to help you realise your tasks?
- What were the main weaknesses that DocuDipity exhibited in supporting your tasks?
- Currently DocuDipity includes a limited set of analysis rules. Can you think of any additional one(s) that would be useful/interesting for you?

We subjected the text answers to a *grounded theory* analysis. Grounded theory [35] is a method often used in social science to extract relevant concepts from unstructured corpora of natural language resources. In opposition to traditional methods aiming at fitting (and sometimes forcing) the content of the resources into a predefined model, grounded theory aims at having the underlying model emerge bottom-up. We proceeded first with *open coding*, with the purpose of extracting actual relevant sentences – called *codes* – from the texts, and subsequently performed the so-called *axial coding*, which is the rephrasing and aggregation of original codes in concepts.

Figure 8 shows the concepts we extracted. The size of each bar depends on the number of codes which contributed to the corresponding concept. Note that the codes mentioned by less than two users were not taken into account.

Fig. 8: The codes mentioned by at least two users, ordered by their frequency and grouped in positive and negative codes

The Figure is actually split in three parts. The top area contains the aspects rated as positive by the testers, followed by the negative aspects and, at the bottom, some mixed notes about the use of the colours in DocuDipity. The overall users' perception was very good, considering that the total number of positive codes is significantly more than the negative ones.

In particular, the testers appreciated the presence of the SunBurst view labelling it as a very effective tool for the proposed tasks. The coexistence of multiple views was also considered a further strength of DocuDipity, together with the possibility of zooming into the document' components and navigating the tree XML structure in an alternative way. This confirms our initial assumption that mixing plain text reading and infoview techniques is a valuable tool for scholars.

An element that we expected to be more appreciated by the testers is the synchronization between the two DocuDipity views. However, only two testers stressed its importance,

while all others used both views but never together: when looking a SunBurst segment, for instance, they did not read the corresponding element in the text flow. The possibility of customizing views and modifying rules was instead enjoyed by some more testers. The result is not optimal, we speculate, because the editing of the DocuDipity rules is not simple (as we will discuss later) and hinders the full exploitation DocuDipity.

The plot also shows that the main weakness of DocuDipity is the interface to surf conferences and papers. In fact, DocuDipity requires users to scan the list of conferences, load papers and then scan the list of the papers. This caused some confusion and impacted negatively on the feedback given by the testers. Most of them suggested us to include a search by title option. This was already in our plan but we had preferred to focus on the coordinated views and the SunBurst at this stage. With hindsight, this was not a good choice. The time needed to find papers and the frustration in missing some of them, impacted negatively on the overall judgment of some testers, more than we expected.

The second element which basically all testers agreed on is the fact that some statistics, for instance some counters about the number of paragraphs, sections and inline elements, would have made DocuDipity much more effective and usable. We liked this suggestion and decided to implement it. The last bar in the plot is about colours: some testers considered suboptimal the choice of colors, in particular, in the SunBurst view: the difference between the segments was not very evident and they had difficulties in selecting and extracting some information. A more balanced palette will be included in the next release of the tool.

The answers given to the last question (about new rules for DocuDipity) are also worth mentioning. Apart from one user, all others suggested to get information that could be easily spotted by DocuDipity. The suggestions can be divided in three groups. Most users suggested to identify document components, such as figures, tables, charts, etc.. Other proposed to process the textual content of the articles, for instance by characterising elements according to their language and the presence of grammatical/syntactical mistakes or imperfections. Finally, some testers suggested to go deeper and highlight topics and claims within the articles. Apart from the details of each suggestion, we consider very positive their variety, confirming again the potentialities that users see in DocuDipity.

B. Editing rules in DocuDipity

In order to complete our evaluation we also asked five participants, over 10 who used DocuDipity, to create new visualization rules. Half of the testers were actually excluded since they declared to have a very limited knowledge of Javascript and would have added great difficulties in completing the tasks. Indeed, the main direction of our research is to make it simple to edit rules for such non expert users as well.

The test consisted of two assignments preceded by a warm-up and some training on the Javascript and CSS syntax of DocuDipity:

T3.1 Studying lists: testers were asked to add a rule for identifying lists whose number of items was higher than a given threshold. They were expected to find lists, count the items in each list and filter. The rule was then quite similar to one that they have already seen in the first part of the test.

T3.1 Searching titled sections: testers were asked to add a rule for identifying sections whose title contained the string 'XML' or 'XSLT'. The expected rule was slightly different from the previous ones and based on regular expressions.

After they completed the assignments, we asked testers to answer some open questions about strengths and weaknesses of the editing process. The overall feedback was positive but all testers confirmed that the editing process is still quite complex. There was no particular difference between the two assignments. As expected, the second one was a bit easier after having learnt how the DocuDipity rules work.

All the testers appreciated the possibility of editing rules directly in the interface and to verify results on the fly; the fact that the two DocuDipity views are synchronized was also considered very positive: the users needed to validate the results found by their queries and they were helped a lot by the side-by-side view.

On the other hand, they all agreed that editing rules is not simple. The testers complained that basic Javascript skills are not enough but some knowledge of JQuery, and in particular of selectors, is needed. Most testers suggested us to add at least some documentation about JQuery selectors and API, so as to mitigate this issue. Two testers also stressed on the difficulties in debugging their code, as they basically had to use directly the Javascript console. They also spotted some constraints in the CSS classes that were not well documented (we fixed it now). A more integrated editor and debugger would definitely be helpful.

One tester also complained about the font size of the interface, though all others gave positive feedback about the overall organization of the editing and visualization areas.

In conclusion, DocuDipity rules were well accepted but still a bit difficult to write and some simplification is needed. We are envisioning an intermediate layer that takes as input visualization rules written in a simplified language and convert them in JavaScript in a transparent way. For instance, a very simple XPath expression would be enough to handle the previous case. Anyway, DocuDipity is totally independent from a particular framework, and any other tools can be used (e.g., EXT-JS) even instead of those included in the current implementation.

Connected to this aspect, there is also the need for a library of visualization rules within DocuDipity. The goal is to create a shared *reading* environment in which users can experiment new rules, write them rules incrementally and reuse them. This opens very interesting perspectives: the initial experiments of one user could be followed by other ones, and even partial intuitions could give input and urge new ideas. Existing rules might be also used to take inspiration by inexperienced users.

VI. CONCLUSIONS

This paper presented a system that exploits coordinated views to analyse scientific articles. In particular, DocuDipity integrates a plain text sequential reader and a SunBurst visualization. These views have been successfully applied to explore scholarly writing habits and to study, for instance, the distribution of citations and the internal organization of the articles' content. DocuDipity proved to be very effective in our tests. On the other hand, the editing process was still considered quite difficult and we are studying how to make it simpler and faster.

Another direction we are investigating is the integration of further views in the overall interface, based on infoview techniques. In fact, these techniques have been successfully deployed in data-centric applications but not yet fully investigated on documents.

Our plain is also to apply the DocuDipity analysis to other documents and other domains. In particular, we are keen to investigate the usage of well-know XML languages, like TEI or DocBook, in structuring real document. To this end, we would also like to make DocuDipity easy to use to other scholars with no expertise nor skills in Computer Science, e.g., scholars in Humanities, to discover things far beyond our knowledge.

REFERENCES

- [1] C. Tenopir, D. W. King, S. Edwards, and L. Wu, "Electronic journals and changes in scholarly article seeking and reading patterns," in *Aslib proceedings*, vol. 61, no. 1. Emerald Group Publishing Limited, 2009, pp. 5–32.
- [2] C. Tenopir, R. Mays, and L. Wu, "Journal article growth and reading patterns," *New Review of Information Networking*, vol. 16, no. 1, pp. 4–22, 2011.
- [3] K. Hornbæk and E. Frøkjær, "Reading patterns and usability in visualizations of electronic documents," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 10, no. 2, pp. 119–149, 2003.
- [4] K. P. Kingrey, "Concepts of information seeking and their presence in the practical library literature," *Library Philosophy and Practice*, vol. 4, no. 2, pp. 1–14, 2002.
- [5] J. Stasko and E. Zhang, "Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE, 2000, pp. 57–65.
- [6] J. Stasko, R. Cattrambone, M. Guzdial, and K. McDonald, "An evaluation of space-filling information visualizations for depicting hierarchical structures," *International journal of human-computer studies*, vol. 53, no. 5, pp. 663–694, 2000.
- [7] F. Mansmann, T. Göbel, and W. Cheswick, "Visual analysis of complex firewall configurations," in *Proceedings of the ninth international symposium on visualization for cyber security*. ACM, 2012, pp. 1–8.
- [8] D. F. Jerding and J. T. Stasko, "The information mural: A technique for displaying and navigating large information spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 3, pp. 257–271, 1998.
- [9] J. Stasko, C. Görg, and Z. Liu, "Jigsaw: supporting investigative analysis through interactive visualization," *Information visualization*, vol. 7, no. 2, pp. 118–132, 2008.
- [10] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*. IEEE, 2007, pp. 61–71.
- [11] D. M. Eler, F. V. Paulovich, M. C. F. de Oliveira, and R. Minghim, "Coordinated and multiple views for visualizing text collections," in *Information Visualisation, 2008. IV'08. 12th International Conference*. IEEE, 2008, pp. 246–251.
- [12] M. Jern, S. Palmberg, M. Ranlof, and A. Nilsson, "Coordinated views in dynamic interactive documents," in *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*. IEEE, 2003, pp. 95–101.
- [13] R. Gove, C. Dunne, B. Shneiderman, J. Klavans, and B. Dorr, "Evaluating visual and statistical exploration of scientific literature networks," in *Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on*. IEEE, 2011, pp. 217–224.
- [14] M. J. McGuffin and J.-M. Robert, "Quantifying the space-efficiency of 2d graphical representations of trees," *Information Visualization*, vol. 9, no. 2, pp. 115–140, 2010.
- [15] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on graphics (TOG)*, vol. 11, no. 1, pp. 92–99, 1992.
- [16] B. S. Johnson, "Treemaps: visualizing hierarchical and categorical data," 1993.
- [17] M. C. Chuah, "Dynamic aggregation with circular visual designs," in *Information Visualization, 1998. Proceedings. IEEE Symposium on*. IEEE, 1998, pp. 35–43.
- [18] B. B. Bederson, "Photomesa: a zoomable image browser using quantum treemaps and bubblemaps," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 2001, pp. 71–80.
- [19] H. Lü and J. Fogarty, "Cascaded treemaps: examining the visibility and stability of structure in treemaps," in *Proceedings of graphics interface 2008*. Canadian Information Processing Society, 2008, pp. 259–266.
- [20] B. Shneiderman and M. Wattenberg, "Ordered treemap layouts," in *Proceedings of the IEEE Symposium on Information Visualization 2001*, vol. 73078, 2001.
- [21] H.-J. Schulz, S. Hadlak, and H. Schumann, "The design space of implicit hierarchy visualization: A survey," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 4, pp. 393–411, 2011.
- [22] H.-J. Schulz, "Treevis. net: A tree visualization reference," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 11–15, 2011.
- [23] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: animated 3d visualizations of hierarchical information," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1991, pp. 189–194.
- [24] J. Lamping, R. Rao, and P. Pirolli, "A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 401–408.
- [25] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim, "Visual readability analysis: How to make your writings easier to read," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 662–674, 2012.
- [26] C. Collins, S. Carpendale, and G. Penn, "Docuburst: Visualizing document content using language structure," in *Computer graphics forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 1039–1046.
- [27] A. Di Iorio, S. Peroni, F. Poggi, and F. Vitali, "Dealing with structural patterns of xml documents," *Journal of the Association for Information Science and Technology*, vol. 65, no. 9, pp. 1884–1900, 2014.
- [28] —, "A first approach to the automatic recognition of structural patterns in xml documents," in *Proceedings of the 2012 ACM symposium on Document Engineering*. ACM, 2012, pp. 85–94.
- [29] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.
- [30] A. Di Iorio, A. G. Nuzzolese, and S. Peroni, "Identifying functions of citations with citalo," in *Extended Semantic Web Conference*. Springer, 2013, pp. 231–235.
- [31] A. Di Iorio, R. Giannella, F. Poggi, S. Peroni, and F. Vitali, "Exploring scholarly papers through citations," in *Proceedings of the 2015 ACM Symposium on Document Engineering*, ser. DocEng '15. New York, NY, USA: ACM, 2015, pp. 107–116. [Online]. Available: <http://doi.acm.org/10.1145/2682571.2797065>
- [32] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.
- [33] J. Sauro, *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.
- [34] J. Lewis and J. Sauro, "The factor structure of the system usability scale," *Human centered design*, pp. 94–103, 2009.
- [35] J. Corbin, A. Strauss *et al.*, "Basics of qualitative research: Techniques and procedures for developing grounded theory," 2008.

Graphical Animations of the Suzuki-Kasami Distributed Mutual Exclusion Protocol

Dang Duy Bui, Kazuhiro Ogata
School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
Email: {bddang,ogata}@jaist.ac.jp

Abstract—A state machine that formalizes a distributed mutual exclusion protocol called the Suzuki-Kasami protocol is graphically animated. The network with which messages are exchanged among nodes in the protocol is displayed as follows: some limited number of messages in the network are displayed and the others are depressed when there are many messages in the network. We prepare one place dedicated to the message that has been just put into the network by a node (or just sent by a node) and one place dedicated to the message that has been just received by a node (just deleted from the network by a node). The main purpose of graphically animating state machines is to make it possible for humans to perceive characteristics or properties of the state machines by observing their graphical animations. We can guess some properties of the state machine formalizing the Suzuki-Kasami protocol by observing graphical animations of the state machine and confirm them by model checking, which demonstrates that state machine graphical animations could make humans perceive state machine properties.

Keywords—distributed protocols; graphical animations; Maude; model checking; SMGA; state machines;

I. INTRODUCTION

Many kinds of information and communication technology (ICT) systems can be formalized as state machines. Some ICT systems, such as the Internet, have become important societal infrastructures, they must be highly reliable. It is crucial to comprehend ICT systems better so as to make them highly reliable. Because ICT systems can be formalized as state machines, one possible way to comprehend ICT systems is to understand state machines that formalize the ICT systems. There may be multiple possible ways to understand state machines, but one promising way to do so would be to rely on human visual perception. Therefore, Nguyen and Ogata [1] have developed a tool called SMGA that generates graphical animations of state machines. Some shared-memory mutual exclusion protocols and some communication protocols have been tackled with SMGA so far. But, ICT systems that are societal infrastructures are often in the form of distributed systems. In this paper, thus, a distributed mutual exclusion protocol called the Suzuki-Kasami protocol [2] is tackled with SMGA.

This work was partially supported by JSPS KAKENHI Grant Number JP26240008 & JP19H04082.

DOI reference number: 10.18293/DMSVIVA2019-012

As all other distributed systems, the Suzuki-Kasami protocol uses a network with which messages are exchanged among nodes. The network is expressed as an associative-commutative collection (called a soup) of messages in a state machine formalizing the Suzuki-Kasami protocol. Because there may be a lot of messages in the network, it may be impossible to display all of the messages on a limited space. Our idea is to display a limited number of messages and suppress the others. The message that has been just put into the network (just sent by a node) or just deleted from the network (just received by a node) is crucial information and then we prepare one place dedicated to the one that has been just put into the network and one place dedicated to the one that has been just deleted from the network.

Understanding a state machine is to know properties the state machine enjoys. The more state machine properties we know, the better we understand the state machine. The main purpose of SMGA is to make it possible for humans to perceive characteristics or properties of a state machine by observing graphical animations of the state machine. Observing graphical animations of state machines, we could guess some properties of the state machines [3]. We guess some properties of the state machine formalizing the Suzuki-Kasami protocol by observing its graphical animations and confirm the properties by model checking. We use Maude [4], a rewriting logic-based computer language, as a specification language for state machines and a model checker.

The rest of the paper is organized as follows: § II Preliminaries, § III State Machine Graphical Animation (SMGA), § IV Suzuki-Kasami Protocol, § V Specification of Suzuki-Kasami Protocol, § VI Graphical Animations of Suzuki-Kasami Protocol, § VII Confirmation of Guessed Properties with Model Checking, § VIII Related Work and § IX Conclusion.

II. PRELIMINARIES

A state machine $M \triangleq \langle S, I, T \rangle$ consists of a set S of states, the set $I \subseteq S$ of initial states and a binary relation $T \subseteq S \times S$ over states. $(s, s') \in T$ is called a state transition and may be written as $s \rightarrow_M s'$. The set $R_M \subseteq S$ of reachable states w.r.t. M is inductively defined as follows: (1) for each $s \in I$, $s \in R$ and (2) if $s \in R$ and $(s, s') \in T$,

then $s' \in R$. A state predicate p is called invariant w.r.t. M if and only if $p(s)$ holds for all $s \in R_M$. A finite sequence $s_0, \dots, s_i, s_{i+1}, \dots, s_n$ of states is called a finite computation of M if $s_0 \in I$ and $(s_i, s_{i+1}) \in T$ for each $i = 0, \dots, n-1$.

There are multiple possible ways to express states. We express a state as a braced associative-commutative (AC) collection of name-value pairs. AC collections are called soups, and name-value pairs are called observable components. That is, a state is expressed as a braced soup of observable components. The juxtaposition operator is used as the constructor of soups. Let oc_1, oc_2, oc_3 be observable components, and then $oc_1 \ oc_2 \ oc_3$ is the soup of those three observable components. A state is expressed as $\{oc_1 \ oc_2 \ oc_3\}$. There are multiple possible ways to specify state transitions. We specify them as rewrite rules. Concretely, we use Maude [4], a programming/specification language based on rewriting logic. Maude makes it possible to specify complex systems flexibly and is also equipped with model checking facilities (a reachability analyzer and an LTL model checker).

Let us consider as an example a test&set mutual exclusion (or spin-lock) protocol whose pseudo-code is as follows:

Loop:

“Remainder Section”

rm : **repeat while** test&set(*locked*);

“Critical Section”

cs : *locked* := false;

locked is a Boolean variable shared with all processes participating in the protocol. test&set(*locked*) atomically performs the following: it sets *locked* to true and returns the old value stored in *locked*. Each process is located at either rm (Remainder Section) or cs (Critical Section). Initially, each process is located at rm and *locked* is false. When a process wants to enter Critical Section, it repeatedly conducts test&set(*locked*) until false is returned and then goes to cs. When it leaves the critical section, it sets *locked* to false and goes back to rs.

The protocol is formalized as a state machine M_{TS} . When there are three processes participating in the protocol, a state in S_{TS} is expressed as follows:

```
{(pc[p1]: l1) (pc[p2]: l2) (pc[p3]: l3)
 (locked: b)}
```

where l_i (for $i = 1, 2, 3$) is either rs or cs and b is either true or false. I_{TS} consists of one state that is expressed as follows:

```
{(pc[p1]: rs) (pc[p2]: rs) (pc[p3]: rs)
 (locked: false)}
```

which will be referred as *ic*. T_{TS} is specified by the following two rewrite rules:

```
rl [enter] : {(locked: false) (pc[I]: rs) OCs}
```

Figure 1. A picture of states in S_{TS}

```
=> {(locked: true) (pc[I]: cs) OCs} .
```

```
rl [exit] : {(locked: B) (pc[I]: cs) OCs}
=> {(locked: false) (pc[I]: rs) OCs} .
```

Rewrite rules are defined with `rl`, while conditional ones are defined with `crl` and their conditions are written after `if`. `enter` and `exit` are labels (or names) given to the rules, respectively. `I` is a Maude variable of process identifications, `B` is a Maude variable of Boolean values and `OCs` is a variable of observable component soups.

III. STATE MACHINE GRAPHICAL ANIMATION (SMGA)

SMGA [1] basically takes a finite computation of a state machine and generates a graphical animation of the state machine. For each state, a picture is designed and then we get a series of pictures from a finite computation. Such a series of pictures is regarded as a movie film. This is how SMGA generates a graphical animation of a state machine.

We could design a picture of states in S_{TS} as shown in Fig. 1 when there are three processes. Fig. 2 shows a sequence of pictures for M_{TS} generated by SMGA. Observing such a sequence of pictures or a graphical animation of a state machine, we could guess some properties of the state machine [3]. Observing the sequence of pictures shown in Fig. 2, for example, we could guess the property that *locked* is false if and only if there is no process in the critical section, or equivalently *locked* is true if and only if there exists a process in the critical section. The guessed property is invariant with respect to M_{TS} . Such a guessed property could be confirmed by model checking with Maude. We use the Maude reachability analyzer (or the search command). The following command can be used to confirm the guessed property of M_{TS} :

```
search [1] in TS : ic
=>* {(locked: false) (pc[I]: cs) OCs} .
```

Maude exhaustively searches the reachable states $R_{M_{TS}}$ from the initial state *ic* for a state that can match $\{(locked: false) (pc[I]: cs) OCs\}$ but does not find any such states. This means that the guessed property is invariant with respect to M_{TS} when there are three processes.

IV. SUZUKI-KASAMI PROTOCOL

The Suzuki-Kasami protocol is a distributed mutual exclusion protocol [2]. The basic idea is that a node that has a privilege is only allowed to enter its critical section,

Figure 2. A sequence of pictures for M_{TS}

and there exists one and only one privilege in the system. The privilege is owned by a node, or is in the network being transferred by a node to another. We suppose that N nodes participate in the protocol and the natural numbers $1, \dots, N$ are used as their identifications. Let Node be $\{1, \dots, N\}$. The N nodes have no memory in common and can communicate only by exchanging messages. The communication delay is totally unpredictable, namely that although messages eventually arrive at their destinations, they are not guaranteed to be delivered in the same order in which they are sent. There are two kinds of messages

used in the Suzuki-Kasami protocol: request and privilege messages. A request message is in the form $\text{request}(j, n)$, where j is the identification of the node that has sent the message and n is a request number. A privilege message is in the form $\text{privilege}(q, a)$, where q is a queue of node identifications and a is a natural number array of size N .

The Suzuki-Kasami protocol consists of two procedures P1 and P2 for each node $i \in \text{Node}$. The procedures for node i are shown in Fig. 3.

request and *have_privilege* are Boolean variables. *request* indicates whether or not node i wants to enter its critical section, and *have_privilege* indicates whether or not node i owns the privilege. *queue* is a queue of Node . It contains the identifications of nodes that wait to enter their critical sections. *ln* and *rn* are natural number arrays of size N . *ln*[j] for each node $j \in \text{Node}$ is the sequence number of node j 's request granted most recently. *rn* records the largest request number ever received from each of the other nodes. Node i uses *rn*[i] to generate the sequence numbers of its own requests. For each node $i \in \text{Node}$, its *rn* is always meaningful, while its *queue* and *ln* are meaningful only when node i owns the privilege. When the privilege is in the network, *queue* and *ln* contained in the privilege message are meaningful. For each node $i \in \text{Node}$, initially, *request* is false, *have_privilege* is true if $i = 1$ and false otherwise, *queue* is empty, and *ln*[j] and *rn*[j] for each $j \in \text{Node}$ are 0.

If node i wants to enter its critical section, it first calls its own procedure P1, which sets *request* to true. If it happens to own the privilege, it immediately enters the critical section. Otherwise, it generates the next sequence number, namely, incrementing *rn*[i], and sends the request message $\text{request}(i, \text{rn}[i])$ to all other nodes. When it receives a privilege message $\text{privilege}(\text{queue}, \text{ln})$, it enters the critical section. When it leaves the critical section, it sets *ln*[i] to its current sequence number *rn*[i], meaning that the current request has been granted, and updates *queue* such that if there are nodes that want to enter their critical sections and whose identifications are not yet in the queue, their identifications are added to the queue. After that, if *queue* is not empty, node i sets *have_privilege* to false and sends the privilege message $\text{privilege}(\text{deq}(\text{queue}), \text{ln})$ to the node found in the front of the queue. Otherwise, node i keeps the privilege. Finally, node i sets *request* to false and leaves procedure P1.

Whenever $\text{request}(j, n)$ is delivered to node i , node i executes its own procedure P2. However, procedure P2 has to be atomically executed. When node i executes procedure P2, it sets *rn*[j] to n if n is greater than *rn*[j]. Then, if node i owns the privilege, does not want to enter its critical section, and the n th request of node j has not been granted, that is, $\text{rn}[j] = \text{ln}[j] + 1$, then it sets *have_privilege* to false and sends the privilege message $\text{privilege}(\text{queue}, \text{ln})$ to node j .

try(<i>i</i>)	↔	rem	procedure P1
setReq(<i>i</i>)	↔	l1	<i>requesting</i> := true;
chkPrv(<i>i</i>)	↔	l2	if $\neg$ have_privilege then
incRN(<i>i</i>)	↔	l3	<i>rn</i> [<i>i</i>] := <i>rn</i> [<i>i</i>] + 1;
sndReq(<i>i</i>)	↔	l4	for all $j \in \{1, \dots, N\} - \{i\}$ do send request(<i>i</i> , <i>rn</i> [<i>i</i>]) to node <i>j</i> ; endfor
wtPrv(<i>i</i>)	↔	l5	wait until privilege(<i>queue</i> , <i>ln</i>) is received; have_privilege := true; endif
exit(<i>i</i>)	↔	cs	Critical Section;
cmpReq(<i>i</i>)	↔	l6	<i>ln</i> [<i>i</i>] := <i>rn</i> [<i>i</i>];
updQ(<i>i</i>)	↔	l7	for all $j \in \{1, \dots, N\} - \{i\}$ do if ($j \notin \text{queue} \wedge (rn[j] = ln[j] + 1)$) then <i>queue</i> := enq(<i>queue</i> , <i>j</i>); endif endfor
chkQ(<i>i</i>)	↔	l8	if <i>queue</i> ≠ empty then
trsPrv(<i>i</i>)	↔	l9	have_privilege := false; send privilege(deq(<i>queue</i>), <i>ln</i>) to node top(<i>queue</i>); endif
rstReq(<i>i</i>)	↔	l10	<i>request</i> := false; endproc
			// request(<i>j</i> , <i>n</i>) is received; P2 is indivisible.
			procedure P2
			<i>rn</i> [<i>j</i>] := max(<i>rn</i> [<i>j</i>], <i>n</i>);
			if have_privilege $\wedge \neg$ request $\wedge (rn[j] = ln[j] + 1)$ then
			have_privilege := false;
			send privilege(<i>queue</i> , <i>ln</i>) to node <i>j</i> ;
			endif
			endproc
recReq(<i>i</i>)	↔		

Figure 3. Suzuki-Kasami distributed mutual exclusion protocol in an Algol-like language

V. SPECIFICATION OF SUZUKI-KASAMI PROTOCOL

Let Nat, Bool, Loc, NodeQueue, and NatNArray be the set of all natural numbers, the set of the Boolean values (true and false), the set of all locations, such as rem and l1, the set of all queues of Node, and the set of all natural number arrays whose sizes are N , respectively. A request message addressed to node $i \in \text{Node}$ by node $j \in \text{Node}$ is expressed as $\text{msg}(i, \text{req}(j, k))$, where $k \in \text{Nat}$, msg is used as the constructor of messages and req is used as the constructor of requests. A privilege message addressed to node $i \in \text{Node}$ is expressed as $\text{msg}(i, \text{priv}(q, a))$, where $q \in \text{NodeQueue}$, $a \in \text{NatNArray}$ and priv is used as the constructor of privileges. Let Req and Priv be $\{\text{req}(i, n) \mid i \in \text{Node}, n \in \text{Nat}\}$ and $\{\text{priv}(q, a) \mid q \in \text{NodeQueue}, a \in \text{NatNArray}\}$, respectively. The network is formalized as a soup of messages that are request and privilege messages. The set MsgSoup of all soups of messages is inductively defined as follows: $\text{void} \in \text{MsgSoup}$, for each $i \in \text{Node}$, $r \in \text{Req}$, and $p \in$

Priv, $\text{msg}(i, r) \in \text{MsgSoup}$ and $\text{msg}(i, p) \in \text{MsgSoup}$, and for each $ms_1, ms_2 \in \text{MsgSoup}$, $ms_1 ; ms_2 \in \text{MsgSoup}$. A semicolon $;$ is used as the constructor of soups of messages. void denotes the empty soup of messages and is the identity of $;$, namely that $ms ; \text{void} = \text{void} ; ms = ms$ for each $ms \in \text{MsgSoup}$. Each message is also treated as the singleton soup that only consists of the message.

The Suzuki-Kasami protocol is formalized as a state machine M_{SK} , which is specified in Maude. P1 is divided into 12 regions shown in Fig. 3. The 12 regions are referred as the 12 locations, such as rem and l1. We suppose that each node is at one of those 12 locations. Procedure P2 is regarded as one region and then there are totally 13 regions in the Suzuki-Kasami protocol. The 13 regions are given names, such as try(*i*) and setReq(*i*), shown at the left-most column in Fig. 3. For each node *i*, there are 13 kinds of transitions that corresponds to the 13 regions. The 13 region names are used to refer to the 13 kinds of transitions. Let

```

r1 [try] : {(#req[I]: X) (pc[I]: rem) (tran: T) OCs}
=> {(#req[I]: (if X < M then X + 1 else X fi)) (pc[I]: (if X < M then l1 else rem fi))
(tran: try(I)) OCs} .

r1 [setReq] : {(pc[I]: l1) (request[I]: F) (tran: T) OCs}
=> {(pc[I]: l2) (request[I]: true) (tran: setReq(I)) OCs} .

r1 [chkPrv] : {(pc[I]: l2) (havePriv[I]: F) (tran: T) OCs}
=> {(pc[I]: (if F then cs else l3 fi)) (havePriv[I]: F) (tran: chkPrv(I)) OCs} .

r1 [incRN] : {(pc[I]: l3) (rn[I]: RN) (idx[I]: K) (tran: T) OCs}
=> {(pc[I]: l4) (rn[I]: RN[I] := (RN[I]) + 1) (idx[I]: l) (tran: incRN(I)) OCs} .

r1 [sndReq] : {(pc[I]: l4) (idx[I]: K) (rn[I]: RN) (nw: NW) (tran: T) OCs}
=> {(pc[I]: if K == N then l5 else l4 fi) (idx[I]: if K == N then l else K + 1 fi) (rn[I]: RN)
(nw: (if K == I then NW else msg(K, req(I, RN[I])) ; NW fi)) (tran: sndReq(I)) OCs} .

r1 [wtPrv] : {(pc[I]: l5) (havePriv[I]: F) (ln[I]: LN') (queue[I]: Q')
(nw: (msg(I, priv(Q, LN)) ; NW)) (tran: T) OCs}
=> {(pc[I]: cs) (havePriv[I]: true) (ln[I]: LN) (queue[I]: Q) (nw: NW)
(tran: wtPrv(I)) OCs} .

r1 [exit] : {(pc[I]: cs) (tran: T) OCs} => {(pc[I]: l6) (tran: exit(I)) OCs} .

r1 [cmpReq] : {(pc[I]: l6) (rn[I]: RN) (ln[I]: LN) (idx[I]: K) (tran: T) OCs}
=> {(pc[I]: l7) (rn[I]: RN) (ln[I]: LN[I] := RN[I]) (idx[I]: l) (tran: cmpReq(I)) OCs} .

r1 [updQ] : {(pc[I]: l7) (idx[I]: K) (rn[I]: RN) (ln[I]: LN) (queue[I]: Q) (tran: T) OCs}
=> {(pc[I]: if K == N then l8 else l7 fi) (idx[I]: if K == N then l else K + 1 fi)
(rn[I]: RN) (ln[I]: LN)
(queue[I]: if K /= I and not(K \in Q) and (RN[K] == (LN[K]) + 1)
then put(Q, K) else Q fi) (tran: updQ(I))} .

r1 [chkQ] : {(pc[I]: l8) (queue[I]: Q) (tran: T) OCs}
=> {(pc[I]: if empty?(Q) then l10 else l9 fi) (queue[I]: Q) (tran: chkQ(I)) OCs} .

r1 [trsPrv] : {(pc[I]: l9) (havePriv[I]: F) (ln[I]: LN) (queue[I]: Q) (nw: NW)
(tran: T) OCs}
=> {(pc[I]: l10) (havePriv[I]: false) (ln[I]: LN) (queue[I]: Q)
(nw: (msg(top(Q), priv(get(Q), LN)) ; NW)) (tran: trsPrv(I)) OCs} .

r1 [rstReq] : {(pc[I]: l10) (request[I]: F) (tran: T) OCs}
=> {(pc[I]: rem) (request[I]: false) (tran: rstReq(I)) OCs} .

crl [recReq] : {(pc[I]: L) (request[I]: F) (havePriv[I]: F') (rn[I]: RN) (ln[I]: LN)
(queue[I]: Q) (nw: (msg(I, req(J, X)) ; NW)) (tran: T) OCs}
=> {(pc[I]: L) (request[I]: F) (havePriv[I]: if C then false else F' fi)
(rn[I]: RN[J] := Max) (ln[I]: LN) (queue[I]: Q)
(nw: if C then (msg(J, priv(Q, LN)) ; NW) else NW fi) (tran: recReq(I)) OCs}
if I /= J /\ L /= l10 /\ L /= l8 /\ L /= l7 /\
Max := if (RN[J]) < X then X else RN[J] fi /\
C := F' and not(F) and Max == (LN[J]) + 1 .

```

Figure 4. Rewrite rules specifying T_{SK}

us note that $\text{sndReq}(i)$ and $\text{updQ}(i)$ correspond to each iteration of the loops at labels l4 and l7, respectively.

When there are three nodes, a state in M_{SK} is expressed as follows:

$\{n(1) \ n(2) \ n(3) \ (nw: ms) \ (tran: t)\}$

where $n(i)$ is as follows:

$(\#req[i]: n) \ (pc[i]: l) \ (request[i]: b1) \ (havePriv[i]: b2) \ (rn[i]: a1) \ (ln[i]: a2) \ (queue[i]: q) \ (idx[i]: j)$

and where ms is a soup of messages in the network, t is the transition that has been just taken, n is the number of requests made by node i , l is the location where node i is, $b1$ is the value of the node i 's *request*, $b2$ is the value of the node i 's *have_privilege*, q is the value of the node i 's *queue* and j is the value of the node i 's j , a loop variable. The state expression defines S_{SK} .

When there are three nodes, I_{SK} consists of one state that is expressed as follows:

$\{n(1) \ n(2) \ n(3) \ (nw: void) \ (tran: notran)\}$

Figure 5. A picture of states in S_{SK}

where *notran* means that no transition has been taken and $n(I)$ is as follows:

```
(#req[I]: 0) (pc[I]: rem) (request[I]: false)
(havePriv[I]: (I == 1)) (rn[I]: ia)
(ln[I]: ia) (queue[I]: empty) (idx[I]: 1)
```

where *ia* denotes the natural number array such that each slot is 0, $I == 1$ is true if *I* is 1 and false otherwise, and *empty* denotes the empty queue.

T_{SK} is specified in terms of (conditional) rewrite rules that are shown in Fig.4. The words starting with a capital letter, such as *X*, *I*, *T* and *OCs*, are Maude variables. Their types (or sorts) could be understood from the context. For example, *X*, *I*, *T* and *OCs* are variables of *Nat*, *Loc*, *transition names* and *observable component soups*, respectively. $RN[I] := (RN[I]) + 1$ is the array assignment at index *I*. $K \setminus in Q$ is the membership predicate $K \in Q$ over queues. *put*(*Q*,*K*) denotes the queue obtained by putting *K* into *Q* at bottom. *top*(*Q*) is the top element of *Q*. *get*(*Q*) denotes the queue obtained by deleting the top from *Q*. What is called a matching equation¹ $V := T$, where *V* is a fresh variable and *T* is a term, can be used in rule conditions and is like **let** expressions in functional programming languages.

This paper describes how to interpret four rewrite rules *sndReq*, *wtPrv*, *trsPrv* and *recReq*. The other rules can be interpreted likewise. Rewrite rule *sndReq* says that when node *I* is located at *l4*, the message $msg(K, req(I, RN[I]))$ is put into the network, where the message is addressed to node *K*, unless *K* equals *I* because *I* does not need to send itself the message. Moreover, if *K* equals *N*, node *I* moves to *l5* and otherwise it stays *l4*. Rewrite rule *wtPrv* says that when node *I* is located at *l5*

and there is a privilege message $msg(I, priv(Q, LN))$ addressed to *I*, node *I* receives it, enters the critical section and sets its *have_privilege* to true. Rewrite rule *trsPrv* says that when node *I* is located at *l9*, it puts a privilege message $msg(top(Q), priv(get(Q), LN))$ in the network, where the message is addressed to the node *top(Q)*, and sets its *have_privilege* to false. Rewrite rule *recReq* says that when there is a request message $msg(I, req(J, X))$ addressed to node *I* in the network and the designated condition is fulfilled, then node *I* puts a privilege message $msg(J, priv(Q, LN))$ in the network and sets its *have_privilege* to false. Among the designated condition is that node *I* is not located at *l7*, *l8* or *l10*. This is because otherwise a livelock may occur [5], [6], [7].

VI. GRAPHICAL ANIMATIONS OF SUZUKI-KASAMI PROTOCOL

Fig.5 shows a picture of states in S_{SK} when there are three nodes participating in the Suzuki-Kasami protocol. There is a pane (called the *nw* pane) located in the left upper corner where the messages in the network are displayed. Under the *nw* pane, there is a pane (called the *nw (received)* pane) where the message that has been just received by a node (or just deleted from the network) is displayed. Under the *nw (received)* pane, there is a pane (called *nw (sending)* pane) where the message that has been just sent by a node (or just put into the network) is displayed. For each node $i = 1, 2, 3$, there are places to display the node *i*'s *request*, *j*, *have_privilege* and *rn*. There is always exactly one *queue* that is meaningful and then there is one place to display the meaningful *queue*. There is always exactly one *ln* that is meaningful and then there is one place to display the meaningful *ln*. If there is a node whose *have_privilege* is true, its *queue* and *ln* are displayed there. If there is

¹In general, a matching equation is in the form $T1 := T2$, where *T1* and *T2* are terms.

Figure 7. A sequence of pictures for M_{SK} (2)

Fig. 7 shows another sequence of two pictures for M_{SK} . The two pictures correspond to two consecutive states State 149 and State 150 in another finite computation of M_{SK} . State 149 goes to State 150 by rewrite rule `recReq` (or `receiveReq(1)`). If it is possible to display all messages in the network on the nw pane, SMGA does so. Otherwise, a limited number of messages are displayed on the nw pane and the others are depressed. In the first picture (of State 149), all messages in the network are displayed on the nw pane. `msg(1, req(2, 5))` is received by node 1 and `msg(2, priv(empty, (1 : 4), 2 : 4))` is put into the network by node 1. Then, it is impossible to display all messages in the network on the nw pane. Therefore, 10 messages out of 12 ones are displayed on the nw pane and the two messages `msg(3, req(2, 5))` and `msg(2, priv(empty, (1 : 4), 2 : 4))` are depressed in the second picture (of State 150). Instead of displaying the two messages, “...” is displayed on the nw pane in addition to the 10 messages.

VII. CONFIRMATION OF GUESSED PROPERTIES WITH MODEL CHECKING

Observing some graphical animations of S_{SK} , we guess that there is always at most one node at cs, l6, l7, l8 or l9 at any given moment. This is true in the six pictures shown in Fig. 6 and Fig. 7. The guessed property can be confirmed by model checking with Maude as follows:

```
search [1] in SK : ic
=>* {(pc[I]: cs) (pc[J]: cs) OCs} .
search [1] in SK : ic
=>* {(pc[I]: cs) (pc[J]: l6) OCs} .
search [1] in SK : ic
=>* {(pc[I]: cs) (pc[J]: l7) OCs} .
search [1] in SK : ic
=>* {(pc[I]: cs) (pc[J]: l8) OCs} .
search [1] in SK : ic
=>* {(pc[I]: cs) (pc[J]: l9) OCs} .
```

```
search [1] in SK : ic
=>* {(pc[I]: l6) (pc[J]: l6) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l6) (pc[J]: l7) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l6) (pc[J]: l8) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l6) (pc[J]: l9) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l7) (pc[J]: l7) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l7) (pc[J]: l8) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l7) (pc[J]: l9) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l8) (pc[J]: l8) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l8) (pc[J]: l9) OCs} .
search [1] in SK : ic
=>* {(pc[I]: l9) (pc[J]: l9) OCs} .
```

For example, the first search command tries to find a reachable state in which there are two processes at cs simultaneously. Maude does not find any reachable states from `ic` that match any of those patterns. Therefore, the guessed property is invariant with respect to M_{SK} when there are three nodes and each node enters its critical section once.

Observing some graphical animations of S_{SK} , we also guess that there is a privilege message in the network if and only if there is no node located at cs, l6, l7, l8 or l9 at any given moment. This is true in the six pictures shown in Fig. 6 and Fig. 7. The guessed property can be confirmed by model checking with Maude as follows:

```
search [1] in SK : ic
=>* {(nw: (msg(I, priv(Q, A)) ; NW))
      (pc[J]: cs) OCs} .
search [1] in SK : ic
=>* {(nw: (msg(I, priv(Q, A)) ; NW))
```

```

      (pc[J]: 16) OCs} .
search [1] in SK : ic
=>* {(nw: (msg(I,priv(Q,A)) ; NW))
      (pc[J]: 17) OCs} .
search [1] in SK : ic
=>* {(nw: (msg(I,priv(Q,A)) ; NW))
      (pc[J]: 18) OCs} .
search [1] in SK : ic
=>* {(nw: (msg(I,priv(Q,A)) ; NW))
      (pc[J]: 19) OCs} .

```

For example, the first search command tries to find a reachable state in which there is a privilege message in the network and a process is located at *cs* simultaneously. Maude does not find any reachable states from *ic* that match any of those patterns. Therefore, the second guessed property is also invariant with respect to M_{SK} when there are three nodes and each node enters its critical section once.

Observing some graphical animations of S_{SK} , we also guess that there exists a node whose *have_privilege* is true if and only if there does not exist any other node at *cs*, *l6*, *l7*, *l8* or *l9* at any given moment. This is true in the six pictures shown in Fig.6 and Fig.7. The guessed property can be confirmed by model checking with Maude as follows:

```

search [1] in SK : ic
=>* {(pc[I]: cs) (havePriv[J]: true) OCs}
such that I /= J .
search [1] in SK : ic
=>* {(pc[I]: l6) (havePriv[J]: true) OCs}
such that I /= J .
search [1] in SK : ic
=>* {(pc[I]: l7) (havePriv[J]: true) OCs}
such that I /= J .
search [1] in SK : ic
=>* {(pc[I]: l8) (havePriv[J]: true) OCs}
such that I /= J .
search [1] in SK : ic
=>* {(pc[I]: l9) (havePriv[J]: true) OCs}
such that I /= J .

```

For example, the first search command tries to find a reachable state in which a process is located at *cs* and some other process *have_privilege* is true simultaneously. Maude does not find any reachable states from *ic* that match any of those patterns. Therefore, the third guessed property is also invariant with respect to M_{SK} when there are three nodes and each node enters its critical section once.

VIII. RELATED WORK

A graphical user interface for Maude-NPA has been developed [8]. Maude-NPA is a high-level security protocol analysis language and system implemented on the top of Maude. The graphical user interface is dedicated to Maude-NPA and then cannot be used for our purpose, namely that graphical animations of state machines can be displayed, helping humans guess properties of the state machines based on the graphical animations.

Specification animation has been actively studied. Specification animation means making formal specification exe-

cutable by translating formal specifications into executable programs because most formal specification languages are not executable. Specification animations have been used to inspect formal specifications [9], to monitor software through formal specification animation [10], to validate formal models by refinement animation [11] and to make a specification-based testing better [12]. Maude is inherently executable and then it is unnecessary to translate Maude specifications into executable programs.

May Thu Aung, et al. [13] uses a mutual exclusion protocol called Qlock to conduct a case study in which some properties of Qlock have been guessed by observing graphical animations of Qlock generated with SMGA, model checked with Maude and theorem proved with CafeOBJ [14]. One piece of our future work is to theorem prove the properties guessed in this paper with CafeOBJ.

Computer networks have been grown and intricate. Social networking service (SNS) has been used by many people over the world and then networks constituted of those SNS users have become very complex. Visualization is one promising way to comprehend such complex networks and then network visualization has been intensively studied. Tools, such as Gephi [15] and Cytoscape [16], have been developed. Some visualization techniques used in those tools could be used for our purpose. Because state machines cannot be necessarily expressed as networks, however, those network visualization tools cannot be directly used to graphically animate state machines.

IX. CONCLUSION

We have described graphical animations of the Suzuki-Kasami protocol with SMGA. Observing them has made us guess some properties of the state machine formalizing the Suzuki-Kasami protocol. We have used the Maude reachability analyzer (the search command) to confirm that the guessed properties are invariant with respect to the state machine when there are three nodes and each node enters its critical section once. The case study demonstrates that state machine graphical animations could make humans perceive state machine properties. Another piece of our future work is to generate graphical animations of some other distributed protocols, such as Paxos [17], guess some properties of those protocols by observing the graphical animations, confirm the guessed properties by model checking and theorem prove them.

REFERENCES

- [1] T. T. T. Nguyen and K. Ogata, "Graphical animations of state machines," in *15th IEEE Intl Conf. Depend., Auton. & Secure Comput.*, 2017, pp. 604–611.
- [2] I. Suzuki and T. Kasami, "A distributed mutual exclusion algorithm," *ACM Trans. Comput. Syst.*, vol. 3, no. 4, pp. 344–349, 1985.

- [3] T. T. T. Nguyen and K. Ogata, “Graphically perceiving characteristics of the MCS lock and model checking them,” in *7th Intl Wksh SOFL+MSVL*, ser. Lecture Notes in Computer Science, vol. 10795. Springer, 2017, pp. 3–23.
- [4] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott, *All About Maude – A High-Performance Logical Framework: How to Specify, Program and Verify Systems in Rewriting Logic*, ser. Lecture Notes in Computer Science. Springer, 2007, vol. 4350.
- [5] K. Ogata and K. Futatsugi, “Formal analysis of Suzuki & Kasami distributed mutual exclusion algorithm,” in *5th Intl Conf. Formal Methods for Open Object-Based Dist. Sys.*, ser. IFIP Conference Proceedings, vol. 209. Kluwer, 2002, pp. 181–195.
- [6] —, “Analysis of the Suzuki-Kasami algorithm with the Maude model checker,” in *12th Asia-Pacific Soft. Eng. Conf.* IEEE Computer Society, 2005, pp. 159–166.
- [7] —, “Comparison of Maude and SAL by conducting case studies model checking a distributed algorithm,” *IEICE Trans.*, vol. 90-A, no. 8, pp. 1690–1703, 2007.
- [8] S. Santiago, C. L. Talcott, S. Escobar, C. A. Meadows, and J. Meseguer, “A graphical user interface for Maude-NPA,” in *9th Spanish Conf. Prog. & Lang.*, 2009, pp. 3–20.
- [9] M. Li and S. Liu, “Integrating animation-based inspection into formal design specification construction for reliable software systems,” *IEEE Trans. Reliability*, vol. 65, no. 1, pp. 88–106, 2016.
- [10] H. Liang, J. S. Dong, J. Sun, and W. E. Wong, “Software monitoring through formal specification animation,” *Innov. in Sys. & Soft. Eng.*, vol. 5, no. 4, pp. 231–241, 2009.
- [11] S. Hallerstede, M. Leuschel, and D. Plagge, “Validation of formal models by refinement animation,” *Sci. Comput. Program.*, vol. 78, no. 3, pp. 272–292, 2013.
- [12] F. Nagoya and S. Liu, “A comparative study of a GUI-aided formal specification construction approach,” in *17th Intl Conf. Comput. Sci. & App.*, ser. LNCS, vol. 10404. Springer, 2017, pp. 273–283.
- [13] M. T. Aung, T. T. T. Nguyen, and K. Ogata, “Guessing, model checking and theorem proving of state machine properties – a case study on Qlock,” *Intl J. Softw. Eng. & Comput. Sys.*, vol. 4, no. 2, pp. 1–18, 2018.
- [14] R. Diaconescu and K. Futatsugi, *Cafeobj Report - The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, ser. AMAST Series in Computing. World Scientific, 1998, vol. 6.
- [15] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: An open source software for exploring and manipulating networks,” in *3rd International Conference on Weblogs and Social Media*.
- [16] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, “Cytoscape: A software environment for integrated models of biomolecular interaction networks,” *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, 2003.
- [17] L. Lamport, “The part-time parliament,” *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, 1998.

A study on the interaction of Environmental Information in the Protection of Shanghai Guangfulin Cultural Heritage

Chuanhong Xu

School of Art And Design
Shanghai University of Engineering Science
Shanghai, China
chuanghongxu@sina.com

Mengying Kong

School of Art And Design
Shanghai University of Engineering Science
Shanghai, China
chuanghongxu@sina.com

Abstract—This paper explores the interaction design of Environmental Information of cultural heritage in China from the perspective of Shanghai Guangfulin's "marketplace". It argues that "Guangfulin's marketplace culture" is rarely discussed on social media based on the analysis of the Guangfulin's official website and other social media. From the information, data and technology propagation and communication activities presented by the existing Guangfulin cultural heritage, it is suggested that some disadvantages in the protection of cultural heritage has been found, such as absence of interpersonal interaction and single interaction mechanism. This paper proposes that the academia should facilitate the study on Guangfulin marketplace culture, enhance communication and rebuild the mechanism of environmental information interaction of Guangfulin Culture..

Keywords- Environmental Information, Cultural Heritage, Marketplace Culture

I. INTRODUCTION

Based on archaeological research, ancestors have drilled wood for fire, living and working in Songjiang Guangfulin area in Shanghai, China since 5000 years ago. It was officially coined as "Guangfulin Culture". The Guangfulin Culture has not only the trace of Liangzhu culture from the south, but also includes the track of Longshan culture from the northern Yellow River. Thus, the archaeologists suggest that Guangfulin reflects the earliest stage of the "immigrant culture" in Shanghai.

About 4000 years ago, the ancestors from the Yellow River in the Central Plains(Shandong province, Southeastern Henan province and Northern Anhui province) came to settle in the Guangfulin area at the southern foot of Jiufeng, the remaining vein of Tianmu Mountain. They collided and merged with the native southern civilization, constructing trees as nests, building homes, picking fruits, fishing, hunting and farming. With the increasing population, the ancestors need to constantly exploit new residences and dig wells for survival.

In the site of the excavation of Guangfulin, there are hundreds of wells have been unearthed according to statistics. In the site, J14 well is one of the symbolic relics, which were excavated from 2001 to 2002 and the wellhead is nearly

circular; the diameter is approximately 2.43-2.50 meters and the depth of the well is 3.10 meters; the inclined straight wall is recessed; the bottom is flat and the diameter of the bottom hole is 0.3-0.5 meters. Meanwhile, a basket-shaped bamboo piece was also excavated in the well. Based on the discussion above, experts identified J14 well as a cultural relic in the last stage of Liangzhu culture.

A large number of wells found in Guangfulin site demonstrate that Guangfulin area already had the characteristics of large settlements and cities as early as the Eastern Zhou Dynasty (Figure 1). Some researchers even consider Guangfulin as the center of politics, economy, transportation and culture in the eastern part of Wu country. Some examples can be demonstrated as following:

Figure 1. Excavation Site of Guangfulin Site

Firstly, the wells of the Eastern Zhou Dynasty found in the Guangfulin site have a special shape. And the well cribs were created in coil method by residences. For materials, the well cribs were made of unprocessed stone, which were taken from the Chen Mountain. Secondly, the bottom of the well was paved with wooden boards. The distribution of numerous wells, combined with the surrounding unearthed remnants of bronze ware, bronze knife, original pottery, large bronze vases and a large number of relics of Han Dynasty, Song Dynasty and Yuan Dynasty. These discoveries have revealed Guangfulin formed the larger cities in the Zhou dynasty and Han dynasty. By the Yuan, Ming and Qing Dynasty, Guangfulin was a populous area with merchants, and further developed into a renowned Guangfulin bazaar. Guangfulin was the distributed center of commodity trade and the North-South transitional station in Qing dynasty. Thus, these samples concludes that

Guangfulin's "wells" and "cities" were the origin of Shanghai's ancient civilization and the initial stage of Shanghai's city.

Since then, Guangfulin gradually developed from a market village with the multicultural culture into a prosperous ancient city.

II. THEORETICAL APPROACHES: ENVIRONMENTAL INFORMATION AND INTERACTIVE DESIGN OF CULTURAL RELICS

In recent years, some achievements have been presented by many scholars in the design of environmental information and interaction of cultural sites.

In 2007, Professor Lu Xiaobo at Tsinghua University in Beijing, China suggests that the design of environmental information and interaction is refined in the process of "program development and actual evaluation" (Lu, 2007). According to scholar Zhang Lie and Guan Yan, digital entertainment technology helps to enhance the narrative of cultural heritage exhibition content and space. After that, Wu Yajie, Han Xiaohan, He Yujin and other scholars discussed the exhibit design in cultural sites, including interactive experience design, human-computer interaction and interactive media art; Huang Jiancheng emphasized that the marvelous space displays are the combination of vivid high-tech application and touching artistic expression; Sun Wei studied the interactive media narrative of "non-thread" information. Meanwhile, Wu Weiguang proposed digital technology and diversified methods of information dissemination to lead exhibition design to comprehensive experience. Liu Shuo suggests that the interaction of environmental information should be centered on the For the future research, Zhang Lie proposed that virtual experience design is "content-foremost and user-centered theory" (Zhang, 2008); while Hao Jianying, He Xiaoyou and Sun Minglu suggest that virtual exhibition is the main direction of exhibition development of cultural heritage museums in the 21st century and interaction is the "user-centered design". Zhuge Yang and Lei Yue suggest the development perspective of "experience and interactive design" by using digital technology in the future.

In 2017, Fu Xinyi, Xu Yingqing and other scholars put forward the trend of "big data visualization", while Professor Lou Yongqi at Tongji University, Shanghai, China argues that "the display of environmental interaction design deals with interaction from the point of view of human feelings NHCAS perspective of human-computer interaction, and sustainability and design are the keys" (Lou, 2017).

Murray (1997) suggests that the most essential thing in interactive narrative environment is communicating and immersive narrative. Sparacino's research focuses on the future of digital museums using interactive technologies to facilitate communication between people and objects, and proposes a set of principles and modeling methods to guide the effective and interactivity of digital museums (Sparacino, 2004). Similarly, Terry Cao (2015) argues that "the core essence of interactive technology remains engagement". In *Technology and Digital Initiatives: Innovative Approaches for Museums*, Juilee Decker (2015) discusses future trends in public participation, online experience, analytics, cloud-client computing evaluation of digital museum exhibits.

Martin Schr (1994) argues that museum exhibits are essentially "the relationship between human and substance". At the International Conference on Museum Themes of Interactive Narrative held in Darmstadt, German researchers reached a consensus on the establishment of museum models on users' emotions and experiences. British Scholar Jenny Kidd discusses information communication in cross-media museum, digital interaction, and the evaluation of 3D technology, wearable technology, immersive experience etc (Kidd, 2014). Furthermore, he suggests that the design trend of "user-centered" in the interaction of exhibition and display Information in the digital museum. In 2015, Norwegian scholar Ana Luisa Sánchez Laws has studied the feedback and supply mechanism of museum digital space and proposed the development trend of mobile technology, Interactive Museum website and Social Media Integration based on public participation. The well-known Japanese Museum director Takeshi Hetian suggests the idea that museums are the combination of human and objects.

III. TRANSMISSION ANALYSIS: GUANGFULIN CULTURAL HERITAGE WEBSITE AND SOCIAL MEDIA

<http://www.gflpark.com/gateway/homepage/showHomePage> (the website of Guangfulin Cultural Heritage) shows:

The propagation of exhibition environmental information of Guangfulin cultural heritage includes the part of "Entering Fulin". "Fulin impression" has only 1062 Chinese characters which primarily introduces the excavation and archaeology, inscription and monument-building of Guangfulin Ancient Cultural Site from two aspects of "archaeological discovery, excavation and research" and "protection and Archaeology"

"Fulin Characteristics" presents eight images including "architectural culture", "religious culture", "farming culture", "museum culture", "scientific education", "inheritance and creative culture", "food culture" and "Leisure Tourisms Culture". "Panorama Park" adopts virtual reality digital sand table, 360° panoramic mode, (screen size: 7.0 inch; pupil distance: 63.5 mm; set head display: Cardboard B.)

There are 15 scenarios including Fulin impression, Chen Zilong Memorial Hall, Guangfulin Cultural Exhibition, Ancient Pottery Art Museum, Sanyuan Palace, Flynn Tower, Zhiyi Zen Temple, Wood Art inheritance Exhibition Hall, Fulin Zhaobi, Guangfulin site Core, Tianniu Bridge, Guandi Temple, Guangfulin City God Temple, Bone Needle Square, Cloud College.

The Guangfulin Culture site (Phase I) was officially commissioned on June 26, 2018. For the convenience of tourists, the administration has set up an "online customer service" on the website, where the tourists can "speak" to a robot and have interactive consultation on 20 popular questions, such as "how can I buy a ticket?", "does the elders get a free ticket?", and the robot can response the questions immediately.

By February 18, 2019, an online search of "Guangfulin culture site" on Baidu has about 482,000 results, including 12,194 images and more than 10,000 videos.

The most of video files are from Youku video, Tencent video, Sina video, Sohu Video, Goodview video, Leshi video,

Chinanews.com, Tudou, Iqiyi, Miaobaike, Miaopai, 56, website, Weibo, and Toutiao.

As far as the number of broadcasts is concerned, the top three are from Youku. The video files are: Guangfulin Cultural Relics: Ancient Pottery Collection Hall Shanghai Morning, 1.08 million broadcasts; Guangfulin Cultural Relics Phase I has entered the stress test Shanghai Morning, 1.10 million broadcasts; Guangfulin Cultural Relics: Mudiao Ming Qing Collection Hall, 1 million broadcasts.

More than 10,000 broadcasts also included "Guangfulin Cultural Relics" (87,000 broadcasts, Tencent videos), "Guangfulin Cultural Relics: Wood Carving Ming and Qing Collection Museum" (56,000 broadcasts, Tencent videos) and so on.

The video file was uploaded earlier as "the Guangfulin Culture Park in Songjiang District, Shanghai: Light Lake scenery and Dream Wonderland" (2014-01-24, Broadcast Network) ; In 2018, the largest number of uploaded videos was over 95%.The latest videos are "aerial photos of China: sensational archaeological sites of human culture, Songjiang Guangfulin, Shanghai" (2019-01-28 tencent video) and "Guangfulin Culture site. Thames Town" (6 days ago, Youku) .

IV. PROBLEMS AND SOLUTION

By Feb.18.2019, Retrieval "Guangfulin marketplace Culture" in <http://nvsm.cnki.net/>, there was no items. However, search Guangfulin Culture in the website, the results were 43, but 19 papers such as "New proof of Ancient History", "prehistoric Culture", "wise man guides my academic path" have nothing to do with "Guangfulin Culture". Of the 24 related documents, 4 are general tourism reports, such as "looking for Guangfulin", "the Development and Exploration of Guangfulin Cultural elements in the Design of Tourism Derivatives", and 11 archaeological excavations. For example, "significant achievements in archaeology at Guangfulin site in Shanghai in 2012" and "Review on the Development of Archaeology in Shanghai in 2014"; there are only six academic studies. For instance, "Environmental Analysis of Guangfulin site in Songjiang District, Shanghai", "Archaeological Exhibition Hall and Landscape Design of Shanghai Songjiang Guangfulin Cultural Site Protection area", and the other three articles. As it can be seen that the study of Guangfulin culture, this topic has not yet attracted the attention of the academic community.

According to the documents, "jing" and "shi" are closely related in Chinese culture.

The word "shi" first appeared in the bronze inscriptions of the Yi wares in the Shang and Zhou Dynasty. according to the ancient classic book "Yijing", "shi" is refer to an activity which means people from all over the country were gathered an area for trade at noon. and it is also have interpretation in the book called "Lv Shi Chun Qiu".

As a place for people to exchange products, "shi" developed into a market controlled by the government in the Western Zhou Dynasty. In the following hundreds of years, the establishment or revocation of the "shi" was decided by the government. The institution of "shi fang" was once popular among residence. The "shi" was a commercial district where prohibit build dwelling, the "fang" was a residential district where forbid set up shops.

Taking the top three video files of "Play Number" as an example, broadcasted 1080,000 times "Guangfulin Cultural Relics: Ancient Pottery Collection Museum". From Neolithic to Sui and Tang Dynasties, there were altogether 16 dynasties and 280 treasures, including "the largest pot in the world". It can be said that part of it is related to the Guangfulin Cultural Relics, but "the first phase of Guangfulin Cultural Relics has entered the stress test" is only a news report about the coming visitors. However, the "Guangfulin Cultural Relics: Wood Carving Ming and Qing Collection Museum" exhibits more than 300 pieces of wood carving collected from Huizhou area in southern Anhui Province, but there is no intersection with the "Guangfulin Culture".

As for the project "Guang Fulin's marketplace culture", it has not been disseminated in any form in the social media. According to the historic references which called "Zhou Dynasty Politeness and Craftsman" records: in the design of "shi", the left side is ancestor, the right side is temple, the front is the area where the emperor direction, and the back is the market. the reason why "shi" plan is feasible can be interpreted by two points: on the one hand, the front area is referred to the administrator where emperor worked, and the back area is service distinct especially for folks.in addition, due to the "origin of Chinese Characters", "shi "is means the place where people engage the business events, In ancient Chinese culture, "shi "is a pictograph which is express the composition of space.

ACKNOWLEDGMENT

The project was supported by Shanghai IV Summit Discipline in Design.

REFERENCES

- [1] Lu Xiaobo (2017), The method of interactive design in information design, Scientific News.
- [2] Song, Xinchao (2015), An analysis on smart museum construction, Chinese Museum.
- [3] Lou Yongqi (2017), Human-Computer Interaction, Sustainable and Design in NHCAS perspective. Zhuangshi.
- [4] Zhang, Lie (2008), Fundamental principles of Virtual Reality, Zhuangshi.
- [5] Moggridge, Bill (1990), Design Interactions.
- [6] Kidd, Jenny (2014), Museum in the New Mediascape: Transmedis, Participation, Ethics.
- [7] Decker, Juilee (2015), Technology and Digital Initiatives: Innovative Approaches to Museums.
- [8] Manzini, Ezio (2016), Design, when everyone designs.

Author Index

Bhanu Murthy Neti, Lalita	83
Bui, Dang Duy	125
Carneiro, Glaucio	63
Choy, Clifford Sze-Tsan	11
Ciancarini, Paolo	113
Coccoli, Mauro	95
Cooper, Kendra	30
Costagliola, Gennaro	51
Cuzzocrea, Alfredo	103
D'Arco, Raffaele	51
De Gregorio, Sabato	51
De Rosa, Mattia	51
Di Iorio, Angelo	113
Filho, Joao Werther	63
Fuccella, Vittorio	51
Hota, Chinmay	83
Ivanov, Vladimir	75
Khosravi, Hassan	30
Kong, Mengying	135
Kumar, Lov	83
Larionova, Daria	75
Laurini, Robert	1
Li, Qi	92
Liu, Huiyi	21
Liu, Yufeng	21
Lupo, Daniele	51
Maciel, Rita	63
Magrofuoco, Nathan	58
Maresca, Paolo	95
Molinari, Andrea	95
Mumolo, Enzo	103
Ng, Po Choy	11
Ogata, Kazuhiro	125

Perez-Medina, Jorge Luis	58
Peroni, Silvio	113
Poggi, Francesco	113
Pérez-Medina, Jorge-Luis	40
Roselli, Paolo	58
Strugar, Dragos	75
Succi, Giancarlo	75
Vanderdonckt, Jean	40, 58
Villarreal-Narvaez, Santiago	40, 58
Vitali, Fabio	113
Xu, Chuanghong	135
Xu, Chuanhong	92
Zeng, Xiaoqin	21
Zou, Yang	21

Additional Reviewers

Risi, Michele

DMSVIVA

**Lisbon, Portugal
July 8 - 9, 2019**

**Proceedings of the
25th International
Distributed Multimedia
Systems Conference on
Visualization and
Visual Languages**

Copyright © 2019
Printed by
KSI Research Inc. &
Knowledge Systems Institute
156 Park Square
Pittsburgh, PA 15238 USA
Tel: +1-412-606-5022
Fax: +1-847-679-3166
Email: seke@ksiresearch.org
Printed in USA, 2019
ISBN 1-891706-49-7 (paper)
ISSN 2326-3261 (print)