WEB 2015

The Third International Conference on Building and Exploring Web Based Environments

ISBN: 978-1-61208-407-7

May 24 - 29, 2015

Rome, Italy

WEB 2015 Editors

Maria Del Pilar Angeles, Universidad Nacional Autonoma de Mexico - Del Coyoacan, Mexico

Pascal Lorenz, University of Haute-Alsace, France

WEB 2015

Forward

Third International Conference on Building and Exploring Web Based Environments (WEB 2015), held between May 24-29, 2015 in Rome, Italy, continued a series of events focusing on web-related theoretical and practical aspects, identifying challenges for building web-based useful services and applications, and effectively extracting and integrating knowledge from the Web, enterprise data, and social media.

The Web has changed the way we share knowledge, the way we design distributed services and applications, the way we access large volumes of data, and the way we position ourselves with our peers.

Successful exploitation of web-based concepts by web communities lies on the integration of traditional data management techniques and semantic information into web-based frameworks and systems.

The conference had the following tracks:

- Services and applications
- Features

We take here the opportunity to warmly thank all the members of the WEB 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to WEB 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the WEB 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that WEB 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of Web-based environments. We also hope that Rome, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the historic beauty of the city.

WEB 2015 Chairs

WEB 2015 Advisory Chairs

Irina Astrova, Tallinn University of Technology, Estonia
Zhe Wu, Oracle, USA
Mariusz Trzaska, Polish-Japanese Institute of Information Technology, Poland

Maria Del Pilar Angeles, Universidad Nacional Autonoma de Mexico - Del Coyoacan, Mexico
Davide Rossi, University of Bologna, Italy
Adriana Martin, National University of Austral Patagonia (UNPA), Argentina

WEB 2015

Committee

WEB 2015 Advisory Chairs

Irina Astrova, Tallinn University of Technology, Estonia
Zhe Wu, Oracle, USA
Mariusz Trzaska, Polish-Japanese Institute of Information Technology, Poland
Maria Del Pilar Angeles, Universidad Nacional Autonoma de Mexico - Del Coyoacan, Mexico
Davide Rossi, University of Bologna, Italy
Adriana Martin, National University of Austral Patagonia (UNPA), Argentina

WEB 2015 Technical Program Committee

Rajendra Akerkar, Western Norway Research Institute, Norway
Maha Alabduljalil, University of California - Santa Barbara, USA
Patrick Albert, IBM CAS France - Paris, France
Mohammed Alodib, Qassim University, Saudi Arabia
Remi Arnaud, AMD, France
Irina Astrova, Tallinn University of Technology, Estonia
Sofia Athenikos, IPsoft, USA
Judie Attard, University of Bonn, Germany
Karim Baïna, ENSIAS / Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes - Rabat, Morocco
Henri E. Bal, Vrije Universiteit, The Netherland
Khalid Belhajjame, Paris Dauphine University, France
Carlos Bobed, University of Zaragoza, Spain
Tasneem Brutch, Samsung Electronics – San Jose CA, USA
Coral Calero Muñoz, University of Castilla-La Mancha, Spain
Delroy Cameron, Wright State University, USA
Irene Celino, CEFRIEL, Italy
Sunil Choenni, Research and Documentation Centre / Ministry of Security and Justice, The Netherlands
Paolo Ciancarini, University of Bologna, Italy
Christophe Claramunt, Naval Academy Research Institute, France
Rodrigo Capobianco Guido, São Paulo State University, Brazil
Giovanni Luca Ciampaglia, Indiana University Bloomington, USA
Joseph Corneli, KMI, Open University / PlanetMath, UK
Roberta Cuel, University of Trento, Italy
Dana Dannells, University of Gothenburg, Sweden

Cláudio de Souza Baptista, University of Campina Grande, Brazil
Jeremy Debattista, University of Bonn, Germany
Ramona Enache, Chalmers University of Technology and University of Gothenburg, Sweden
Feroz Farazi, University of Trento, Italy
Cécile Favre, University of Lyon - ERIC Lab, France
Emilio Ferrara, Indiana University Bloomington, USA
Kenneth K. Fletcher, Missouri University of Science and Technology - Rolla, USA
Jakub Flotynski, Poznan University of Economics, Poland
Raffaella Folgieri, Università degli Studi di Milano, Italy
Naoki Fukuta, Shizuoka University, Japan
Christos K. Georgiadis, University of Macedonia, Greece
Abigail Goldsteen, IBM Research - Haifa, Israel
Dorian Gorgan, Technical University of Cluj-Napoca, Romania
Thomas Gottron, Institute for Web Science and Technologies/Universität Koblenz-Landau, Germany
Gregory Grefenstette, Exalead, France
Tudor Groza, The University of Queensland, Australia
João Guerreiro, INESC-ID - Lisbon and Technical University of Lisbon / Instituto Superior Técnico - Lisbon, Portugal
Allel Hadj Ali, LIAS/ENSMA, Poitiers, France
Mirjana Ivanovic, University of Novi Sad, Serbia
Hermann Kaindl, Vienna University of Technology, Austria
Rajaraman Kanagasabai, Institute for Infocomm Research, Singapore
Roula Karam, Università degli Studi di Brescia, Italy
Dimka Karastoyanova, University of Stuttgart, Germany
Hassan Karimi, University of Pittsburg, USA
Randi Karlsen, University of Tromsø, Norway
Jinho Kim, Kangwon National University, South Korea
Styliani Kleanthous Loizou, Open University of Cyprus, Cyprus
Christoph Kögler, T-Systems Multimedia Solutions GmbH, Germany
Fotis Kokkoras, TEI of Thessaly, Greece
Ivan Koychev, University of Sofia, Bulgaria
Efstratios Kontopoulos, Information Technologies Institute (ITI) | Centre for Research & Technology, Hellas (CERTH), Greece
Petr Kremen, Czech Technical University in Prague, Czech Republic
Wolfram Laaser, WWEDU World Wide Education GmbH, Germany
Steffen Lamparter, Siemens AG - München, Deutschland
Christoph Lange, University of Bonn / Fraunhofer IAIS, Germany
Sergio Luján Mora, University of Alicante, Spain
Xiangfeng Luo, Key Lab of Grid Technology - Shanghai, China
Vincenzo Maltese, University of Trento, Italy
Erik Mannens, Ghent University, Belgium
Maristella Matera, Dipartimento di Elettronica, Informatica e Bioingegneria (DEIB) - Politecnico di Milano, Italy

Abdul-Rahman Mawlood-Yunis, Canada Revenue Agency - Ottawa, Canada
Marios Meimaris, Research Center ATHENA, Greece
Michele Melchiori, Università degli Studi di Brescia, Italy
Manfred Meyer, Westfälische Hochschule - University of Applied Sciences, Bocholt, Germany
Jean-Claude Moissinac, Telecom ParisTech, France
Louise Moser, University of California - Santa Barbara, USA
Grzegorz J. Nalepa, AGH University of Science and Technology, Poland
Viorel Negru, West University of Timisoara, Romania
Bo Ning, Dalian Maritime University, China
Talal H. Noor, Taibah University, Saudi Arabia
Francesco Saverio Nucci, Engineering SpA - R&D, Italy
Sandra Sendra Compte, Universitat Politècnica de València, Spain
Tope Omitola, University of Southampton, UK
Mourad Ouziri, Université Sorbonne Paris Cité, France
Luca Panziera, Open University - Milton Keynes, UK
Razan Paul, The University of Queensland - Brisbane, Australia
Tassilo Pellegrini, University of Applied Sciences St. Pölten, Austria
Srinath Perera, University of Moratuwa, Sri Lanka
Silvio Peroni, University of Bologna, Italy
Edson Pinheiro Pimentel, UFABC - Federal University of ABC, Brazil
Michel Plantié, Ecole des Mines d'Ales - Institut Mines-Télécom, France
Laura Po, Università di Modena e Reggio Emilia, Italy
Hemant Purohit, Wright State University, USA
Jose Javier Ramasco, IFISC (CSIC-UIB), Spain
Isidro Ramos, Polytechnic University of Valencia, Spain
Tarmo Robal, Tallinn University of Technology, Estonia
Christophe Roche, Université de Savoie - Le Bourget du Lac, France
Marco Rospocher, Fondazione Bruno Kessler (FBK), Italy
Davide Rossi, University of Bologna, Italy
Gustavo Rossi, UNLP, Argentina
Georgios Santipantakis, University of Piraeus, Greece
Carmen Santoro, ISTI-CNR-Pisa, Italy
Cristina Sarasua, Institute for Web Science and Technologies (WeST) - University of Koblenz-Landau, Germany
Luigi Sauro, University of Naples Federico II, Italy
Monica Scannapieco, National Institute for Statistics - Rome, Italy
Alexander Schill, Technische Universität Dresden, Germany
Erich Schweighofer, Vienna University, Austria
Saeedeh Shekarpour, Universität Leipzig, Germany
Michael Sheng, The University of Adelaide, Australia
Eddie Soulier, Université de Technologie de Troyes (UTT) - France
George Spanoudakis, City University London, UK
Kostas Stefanidis, Institute of Computer Science, Foundation for Research and Technology - Hellas (ICS - FORTH), Greece

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Automated Elicitation of Functional User Requirements for Supporting Cloud Service Search

Andrea Horch, Constantin Christmann and Holger Kett
Fraunhofer Institute for Industrial Engineering IAO
Stuttgart, Germany
{andrea.horch, constantin.christmann, holger.kett}@iao.fraunhofer.de

***Abstract*—The use of cloud services can generate tremendous benefits for companies, especially for small and medium-sized enterprises (SMEs). When searching for a cloud service the specification of the required functions is an important aspect for finding appropriate services. SMEs which may not have access to the knowledge of an information technology (IT) expert can have difficulties in formulating their functional requirements and current cloud service search engines do not provide comfortable assistance for the specification of functional requirements like service features. The contribution of this paper is a technique for the automated identification of service features for supporting cloud service search. The technique uses an ontology to perform an analysis of the functions provided by cloud services or on-premise software which is currently used by the SMEs. The technique was implemented and evaluated in the context of the crafts domain.**

Keywords–Functional Requirements; Requirements Elicitation; Cloud Services; Service Search.

I. Introduction

The adoption of cloud computing holds vast potential for companies. In particular small and medium-sized enterprises (SMEs) can exploit the benefits of cloud computing like such as the outsourcing of resources through the use of a completely external infrastructure or the improvement of processes by using readily available services, which can be accessed from anywhere [11]. Due to its numerous advantages cloud computing already plays an important role for business. Moreover, for small and medium-sized enterprises (SMEs) which may not dispose of high IT expertise it can be difficult to formulate their functional requirements which are an important aspect when searching for suitable software tools. Hence, there is a need to assist the SMEs during the process of functional requirements elicitation when searching for appropriate cloud services.

Current cloud service registries and search engine projects like Cloud Search Portal [3] or the service directories of cloudbook [2] and Cloud Showplace [13] are not adequate in supporting the elicitation of functional requirements during the search process (see Section II). This paper contributes a technique to automatically identify required software features, which is based on an ontology. The ontology represents the domain of software features in the context of the crafts sector. The method was implemented in form of a search engine for the identification of required cloud service features in the crafts domain. The evaluation has verified the suitability of the technique for supporting users during the process of the identification of required cloud service features for cloud service search. The paper is organized as follows: Section II presents related work and Section III is introducing the approach. The implementation of the technique is presented in Section IV. The approach of the evaluation and the results are shown in Section V. We conclude in Section VI and we propose the future work in Section VII.

II. Related Work

Table I gives an overview of previous projects in the field of cloud service search and the tasks they focus on. The cloud portal introduced in [5] implements as a main feature a cloud service search engine. The search engine is based on a domain ontology describing the main features of a cloud service like service type, vendor details, technical information or cost and time details. Cloud service providers use the terms of the ontology to describe their services. Additionally, they have the possibility to add some keywords to the service description. Service consumers, which are searching for an appropriate service, also use the terms of the ontology to describe their requirements for service search. Furthermore, they have the option to add keyword for the search. The matching is done by similarity reasoning based on the domain ontology. The cloud service search engine CloudRecommender presented in [14] is based on an ontology called Cloud Computing Ontology (CoCoOn). The project focuses on the identification and selection of infrastructure services, which belong to the infrastructure as a service (IaaS) model of cloud computing. Users can search by selecting basic configuration parameters for the searched cloud service, which are provided by the domain ontology. Additionally, regular expressions can be used to search for appropriate services. The domain ontology designed in [6] describes cloud services. The ontology is used to assist cloud service providers to describe the cloud services when registering a cloud service in a cloud service registry. The ontology acts as a standard terminology for the service descriptions of the different providers. They have implemented a keyword-based cloud service search engine, which matches the keyword query to the terms of the ontology and infers suitable services. In [7], Cloud Service Crawler Engine (CSCE) is presented which is a crawler engine for cloud service discovery and categorization. CSCE crawls the internet by using the application programming interfaces (APIs) of conventional search engines like Google, Bing or Yandex, as well as a domain ontology describing cloud service characteristics. The crawler uses the concepts in the first levels of the ontology as query keywords to feed the search engines and the `is-a` and `is-not-a` relations between the concepts for the decision whether a search engine result is a cloud service or not. The ontology is also used for the categorization of the service providers of the discovered cloud services into infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), IaaS+PaaS, IaaS+SaaS, PaaS+SaaS and all.

The contribution of [10] is an approach for the generation of semantic cloud service descriptions. The proposed technique transforms a natural language description of a cloud service into a semantic service description by using a domain ontology, as well as natural language processing (NLP) tools. Based on the semantic description of the cloud services a search index is generated, which is used for processing a keyword-based search.

There are further recent approaches which pursue the goal to ease the use and reuse of cloud services. One of those projects is the Artist (Advanced software-based seRvices provisioning and migraTIon of legacy SofTware) project [1] which offers a set of methodologies and tools to owners and developers of cloud services for transforming their software in a way that it can exploit the benefits of cloud features. Another project called REMICS (REuse and MIgration of legacy applications to Interoperable Cloud Services) [9] directly translates natural user requirements to Java code by transforming natural language phrases into the Requirements Specification Language (RSL). Since the approach of the Artist project needs a deeper IT know-how and thus, is more suitable for owners and developers of cloud services than for non-experts it will be not further considered in this paper. The approach of the REMICS project offers the possibility to the users to easily define their software requirements in natural language, but it does not give them a hint if they do not completely know their requirements. As our approach aims to assist the users in identifying their functional requirements it needs to use other techniques than proposed in [1] and [9]. [1] and [9] are not compared to the cloud service search approaches in Table I since they do not mainly focus on cloud service search.

As illustrated in Table I the previous projects presented in this section deliver solutions for cloud service discovery, the description of cloud services as well as the elicitation of non-functional user requirements for cloud services. The elicitation process for the required features is only partially solved by some projects. These projects, which deliver a partial solution for the investigation of the required service features only consider the service type like IaaS or some technical requirements like the available storage size, but they do not consider the denotations of functions nor determine the functions offered by a service. Thus, there is a deficit in the assistance of users during the elicitation process of their required service features when searching for appropriate cloud services, which is filled by the technique introduced in this paper.

III. Approach

Our approach for an automated identification of required service features is based on the assumption that the users already have a software in use, which they want to substitute with an appropriate cloud service. Another possible scenario is that the user knows an on-premise software having all required functions and they are looking for a corresponding cloud service. Figure 1 shows the steps of our approach for automated elicitation of required service features.

In the first step, the user enters the search query, which is the URI of the software website, which should be substituted by the cloud service, or the URI of a similar software having the required functions. Having received the website URI, the content of the website and of all pages linked on it have to be crawled. In the next step, the content is checked against a domain ontology, which contains software function classes. These classes include several labels of function terms. Each class of the ontology is checked for its occurrence in the crawled content. Classes whose labels match a term in the content, are added to the result set. After that step the result set is transformed into a hierarchical order corresponding to the hierarchy of the function classes in the domain ontology. Finally, the hierarchical result list of software functions is displayed to the user.

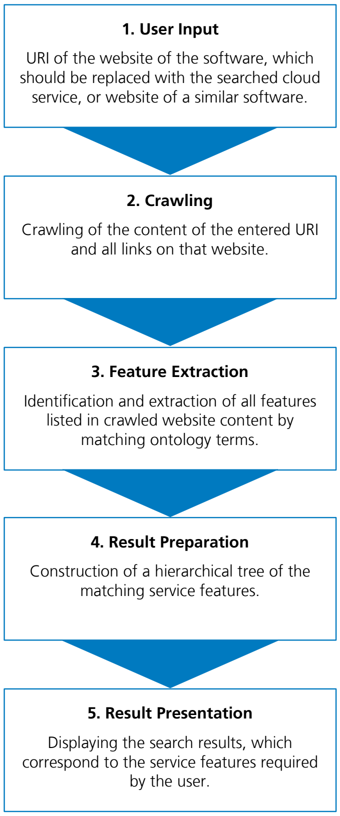

1. User Input
URI of the website of the software, which should be replaced with the searched cloud service, or website of a similar software.
2. Crawling
Crawling of the content of the entered URI and all links on that website.
3. Feature Extraction
Identification and extraction of all features listed in crawled website content by matching ontology terms.
4. Result Preparation
Construction of a hierarchical tree of the matching service features.
5. Result Presentation
Displaying the search results, which correspond to the service features required by the user.

Figure 1. Approach for automated elicitation of required service features

IV. Implementation

The implemented system consists of a user interface, a domain ontology, a component for web content crawling, a component for feature extracting and a component for processing the results.

A. Ontology Creation

For the implementation of the approach a domain ontology is needed for identifying terms of functions and interfaces in the crawled content of the software websites. For the creation of the ontology we used the ontology editor Protégé [12] and followed the Ontology Development 101 methodology described in [8]. The Ontology Development 101 methodology comprises the following steps:

1) Determination of domain and scope of the ontology.

TABLE I. OVERVIEW OF CLOUD SERVICE SEARCH PROJECTS

	cloud service discovery	generation of cloud service description	identification of required functions	identification of other features
[5] J. Kang and K. M. Sim, 2011	●	●	◐	●
[14] M. Zhang et al., 2013	○	●	◐	●
[6] V. S. K. Nagireddi and S. Mishra, 2013	○	●	○	○
[7] T. H. Noor et al., 2013	●	○	○	○
[10] R. Sahandi, A. Alkhalil and J. Opara-Martins, 2013	○	●	○	○

Legend: ○ no solution, ◐ partial solution, ● solution

2) Consideration of reusing existing ontologies.
3) Enumeration of important ontology terms.
4) Definition of the classes and class hierarchy.
5) Definition of the properties of classes (slots).
6) Definition of the facets of the properties (cardinality, value type, domain and range).
7) Creation of instances.

The domain of the ontology covers functional cloud service requirements such as functions and interfaces. It shall be used for the elicitation of the functional requirements by analysing the functional range of a currently used software tool or a similar software from the software description on the website of the provider. We restricted the ontology to describe cloud software in the crafts domain, but it was designed to be easily extended to other domains. Concerning step 2 of the methodology, we have analysed the ontologies of previous work whether these can fit our needs. For this purpose we examined the ontologies used in [4]- [7]. All these ontologies offer several meta data for describing software, especially cloud software. However, they do not include terms or a hierarchy for describing the concrete functions and interfaces of software. Hence, we decided to create a new ontology which includes terms and a hierarchical structure of software functions and interfaces. Important ontology terms where identified by scanning the websites of five different software services of the crafts domain for terms of functions and interfaces and adding them to the ontology in a hierarchical structure. The ontology consists of a class hierarchy (is-a relationships) and different labels for the classes to express synonyms.

Since we plan to extend the ontology to store additional knowledge about the data which is processed by each function in order to be able to increase our system for the search for trusted cloud services it was created in the Resource Description Framework Schema (RDFS) which is a standard of the W3C. For the search for trusted cloud services it is important to know if a service processes or stores critical data like personal data of customers or business secrets.

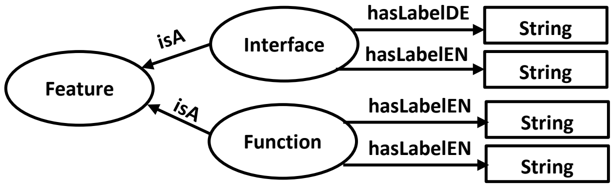


Figure 2. Structure of the software feature ontology

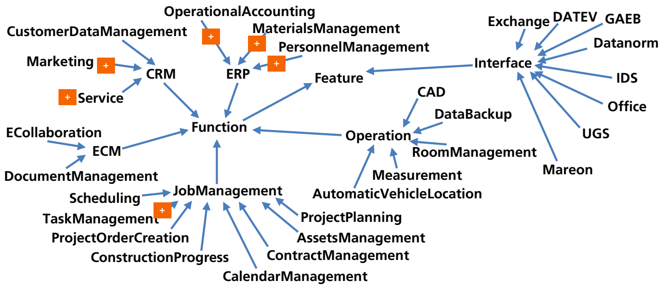


Figure 3. Extract of the software feature ontology

Figure 2 presents the structure of the ontology, whereas Figure 3 shows an extract of the ontology. The class `feature` describes all features of a software and includes the subclasses `function` and `interfaces`, which are features of a software. The class `function` contains several subclasses, which represent concrete functions of a cloud software in the crafts domain like Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) and their subclasses. The class `interface` is a meta class of software interfaces including several subclasses describing concrete interfaces of cloud software in the crafts domain like an office interface or a DATEV interface for transferring product data.

Moreover, the ontology contains the meta classes `FunctionMetaclass` for the description of software functions and a meta class `InterfaceMetaclass` for software interfaces. Another meta class is an extension of the system class `Slot` called `SlotWithLabel` which was designed to be able to add multilingual labels in the form of class properties to a function and an interface class. The ontologies includes the slots (class properties) `labelDE` and `labelEN` which both have the domains `Function`, `FunctionMetaclass`, `Interface`, `InterfaceMetaclass` and `SlotWithLabel`. The properties use the value type `String` for adding English and German labels to a function or interface class in order to define several English and German synonym terms for describing the name of a service function or interface. The ontology includes 36 classes for service functions and 10 different interface classes. It consists 118 German labels and 111 English labels for describing the different function classes. It has 16 German labels as well as 16 English labels for naming the interface classes.

B. System Built

After building the ontology the system for the identification and extraction of required service functions was created. Figure

4 presents the system design.

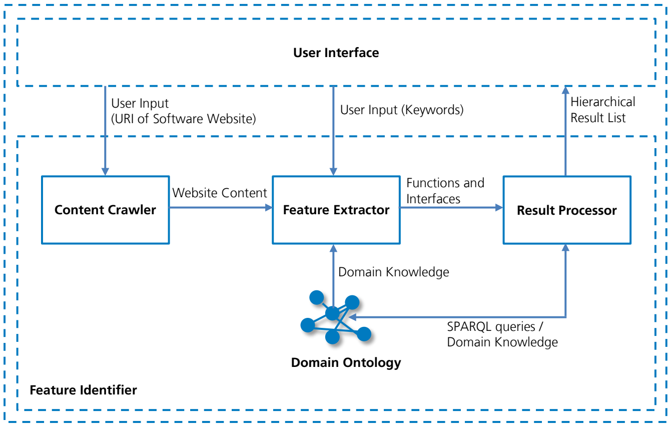
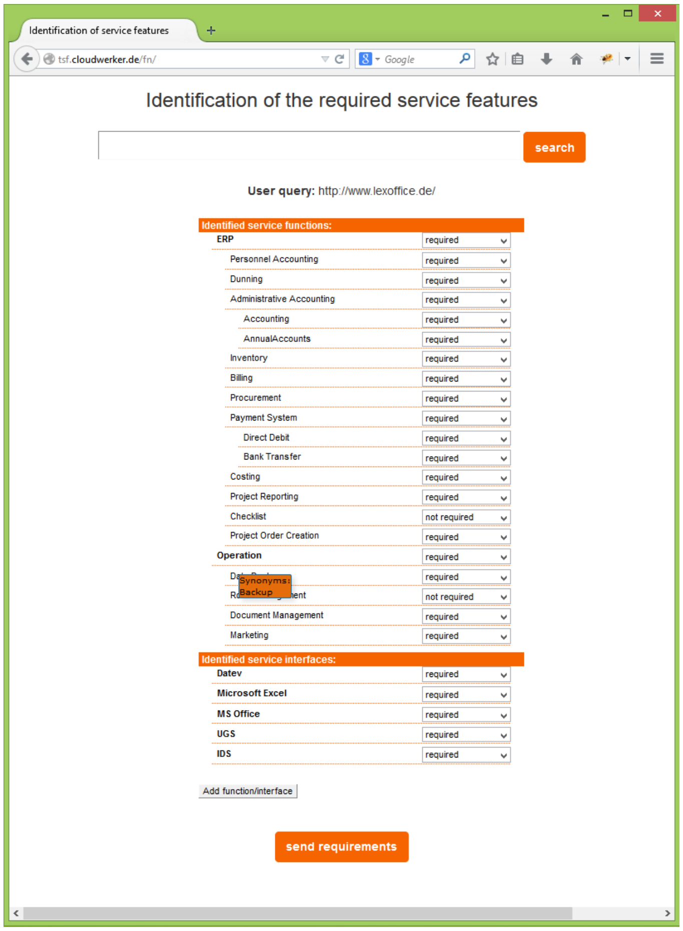


Figure 4. System design

- **User Interface:** The *user interface* is a web application where the users can enter a search query in form of a URI of the software to substitute or a software, which offers a similar set of functions and interfaces as the searched service. We designed the system to use the online description of an actually used software (on-premise software or web service) on the website of the software provider or retailer since we assume that the user will already use a software.
  Having processed the search the results are presented in the *user interface* in the form of hierarchical lists of the identified functions and interfaces. The users can decide, which of the listed features they require and send their selection to the service search engine. An example of displayed results is shown in Figure 5.
- **Content Crawler:** If the user query is the URI of the website the *content crawler* crawls the content of that website and all websites of the same domain, which are linked on it. After the crawling process the *content crawler* transfers the crawled content to the *feature extractor*.
- **Feature Extractor:** The *feature extractor* uses the *domain ontology* to compare the class labels (terms of functions and interfaces) with the website content. All matching ontology classes are added to the result list, which is referred to the *result processor*. Since the crawling is done in real-time during the search operation and some website linking many other domain websites we decided to interrupt the crawling after three minutes. That means, the functions and interfaces only occurring on websites, which are not crawled within three minutes, will not be identified by the system. The interruption after a time of three minutes was chosen because a three minutes time space for crawling has provided good results in our tests - even for bigger websites. We plan to reduce the processing time of three minutes for later versions of the system (e.g. by incremental result generation or a parallel programming approach).
  If the user query is formulated as keywords the *feature extractor* checks the ontology classes directly against the keywords.
- **Result Processor:** The *result processor* takes the result list from *feature extractor* and prepares it for the presentation on *user interface*. It processes SPARQL Protocol And RDF Query Language (SPARQL) queries on the *domain ontology* to get information about the hierarchical structure of the results and converts the result list to hierarchical lists of functions and interfaces, which are transmitted to the *user interface*, which is shown in Figure 5.



Identification of the required service features

User query: http://www.lexoffice.de/

Identified service functions:	
ERP	required
Personnel Accounting	required
Dunning	required
Administrative Accounting	required
Accounting	required
AnnualAccounts	required
Inventory	required
Billing	required
Procurement	required
Payment System	required
Direct Debit	required
Bank Transfer	required
Costing	required
Project Reporting	required
Checklist	not required
Project Order Creation	required
Operation	required
Da[illegible]	required
R[illegible]ent	not required
Document Management	required
Marketing	required
Identified service interfaces:	
Datev	required
Microsoft Excel	required
MS Office	required
UGS	required
IDS	required

Figure 5. Result lists of the identification of required service features

Figure 5 shows an example for a result list from the system for a search for suitable functional service requirements which can be used for a cloud service search in order to substitute a currently used service.

V. Evaluation

The evaluation of our approach is separated into two parts. In the first part, we measure the adequacy of the approach in eliciting the functional user requirements from software websites for being able to run a precise search for relevant cloud services. The second part of the evaluation validates the results of the elicitated functional requirements for running a cloud service search.

ISBN: 978-1-61208-407-7

TABLE II. CONSTRUCTION WEBSITES

software	**URL**
pds abacus	http://www.pds.de/cms/produkte/pds-abacus/
myfactory	http://www.myfactory.com/
scopevisio	https://www.scopevisio.com/
moser	http://www.moser.de/produkte/mosaik/
kwp-bnWin.net	http://www.kwp-info.de/

TABLE III. CONTROL WEBSITES

software	**URL**
lexoffice	http://www.lexoffice.de/
midcom	http://www.midcom.de/
cas pia	http://www.cas-pia.de/
reporta	http://reporta.ag/de/
salesking	http://www.salesking.eu/

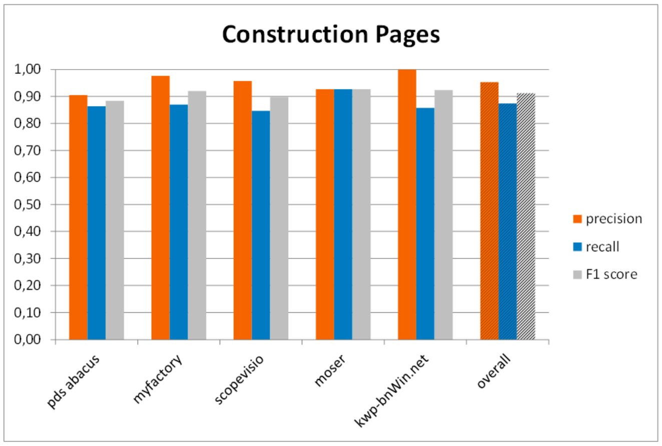


Figure 6. Precision, recall and F1 score for the construction pages

A. *Elicitation of Functional Service Requirements*

The first part of the evaluation analyses the quality of our system for the elicitation of the functional user requirements from software websites by measuring precision, recall and F1 score for the identification of the functions and interfaces offered by software systems, which are described on (1) the software websites we used to build up the domain ontology (construction pages) and (2) other software websites of the crafts domain (control pages), which act as control group.

Criteria for the selection of the construction and control pages were the following:

- **Construction pages:** The construction pages should contain a description of a cloud service of the crafts domain including as many functions and interfaces as possible.
- **Control pages:** The control pages should also contain a description of a cloud service of the crafts domain. The main criteria for a control page was that the service described should differ a lot in respect to structure and terms from those described in the construction pages.

The websites for ontology construction are given in Table II whereas the Uniform Resource Locators (URLs) for the control websites are quoted in Table III.

For this purpose we manually collected all functions and interfaces described on the mentioned websites and compared these results with the result lists delivered by our system. The evaluation results for the software websites used to build up the ontology are shown in Figure 6.

For the construction pages we could achieve an average value for precision of 0.95. We could reach an overall value for recall of 0.87 and for the F1 score of 0.91. Reasons for not achieving 1.00 for all values are the following:

- Crawling interruption: We interrupt the website crawling during the identification process after three minutes. Functions and interfaces only occurring on web pages, which are not crawled could not be identified.
- Wording on websites: Some software functions are described on the website using a sentence and not a single term of the ontology. There is no match between the terms of the ontology and the text on the website, which describes the function.
- Asynchronous JavaScript and XML (AJAX): Some websites use AJAX for displaying some information. This information is only displayed after triggering a specific action. Since we do not trigger actions during the crawling process functions and interfaces described in text, which only occurs after an action, does not become identified.
- No semantic analyses of content: Our process does not analyse the semantics of the crawled website content. There are sentences like ”You can pay the service fee by bank transfer or direct debit”, which specifies how a customer can pay for the use of the software described on the website. However, the system gets a match between the words ”bank transfer” and ”direct debit” in the sentence on the website and the corresponding ontology classes describing the software functions ”bank transfer” and ”direct debit”.

The results for the control pages are shown in Figure 7.

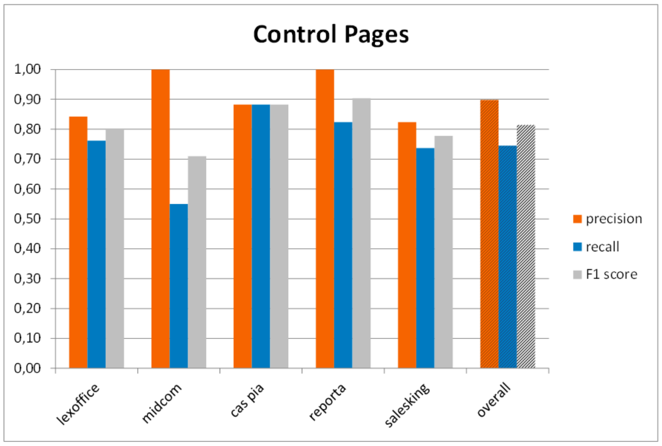


Figure 7. Precision, recall and F1 score for the control pages

For this type of pages we achieved an average value for precision of 0.90, and overall recall of 0.74 and an F1 score of 0.81. This means, on average the system can identify 74 %

of the features listed on software website not used to build the ontology, which is a sufficient result for a system, supporting users to identify their required service features and interfaces.

B. Cloud Service Search

In this part of the evaluation, the suitability of our approach for cloud service search is validated. For this purpose we take the results from the prior step of functional service requirements elicitation for each of the control websites and use the identified service functions and interfaces as input for a cloud service search by using the search engine Google. We used the Google search engine for our approach since the previous approaches described in Section II were not available online. The search query for the cloud service search is created by the leading keyword "cloud service" followed by the required service functions identified in the requirements elicitation step followed by the determined service interfaces. Thus, a search query for the evaluation has the following structure:

"cloud service *<function 1> <function 2> <function 3> ... <interface 1> <interface 2> ...*"

For the service discovery only the search results of the first result page of the Google search were taken into account and maximum one link from search result page to reach the website of an appropriate cloud service was followed. For empty result sets of a keyword search or not sufficient result sets the keyword for a minor interface or function was deleted and a new search was started. For finding the cloud services of Table IV a maximum of three keywords from the results of the requirements elicitation step was deleted.

Table IV presents the results of a Google search for cloud services by using the functional requirements reached by the first step of the approach as search keywords. The table shows the results for five different searches for a substitution of the software services "lexoffice", "midcom", "cas pia", "reporta" and "salesking" by appropriate cloud services. Each search has returned a minimum of three possible services. The results were checked for their suitability to substitute the initial service by comparing the functions and interfaces of the detected service with those of the initial service. In Table IV the suitability for substitution of a detected service is described as "unsuitable" if no or only one function, which is not the main function of the initial service, of the detected service matches the functions of the initial service. The suitability for substitution of a service identified during the search is called "partially suitable" if some functions of the initial service can be covered by the service, but not all. The suitability for substitution of a service is denoted as "suitable" since all functions of the initial service can be substituted by the discovered service. The results show that for each service of the control group at least one cloud service could be found which can totally substitute the initial service. For most services even two or more cloud services for the service replacement could be discovered. The outcome of our evaluation has demonstrated the adequacy of our approach for the automated elicitation of functional requirements to search for appropriate cloud services.

TABLE IV. RESULTS OF THE CLOUD SERVICE SEARCH

Cloud service	**Suitablility for substitution**
lexoffice	
Exact Online http://www.exactonline.de/	●
Microsoft Dynamics http://www.microsoft.com/de-de/dynamics/default.aspx	●
Sage Office Online https://www.sage-office-online.de/	●
midcom	
Salesforce https://www.salesforce.com/	○
umantis Talent Management http://www.umantis.com/	○
karg-edv emis http://www.karg-edv.de/	●
cas pia	
Exact Online http://www.exactonline.de/	●
Salesforce https://www.salesforce.com/	○
SuperOffice CRM Online http://crmonline.superoffice.de/crmonline_062014-1/	●
pds abacus http://www.pds.de/cms/produkte/pds-abacus/uebersicht.html	●
Microsoft Dynamics http://www.kumavision.com/microsoft-dynamics-crm	●
reporta	
Salesforce https://www.salesforce.com/	○
zep http://www.zep.de/	●
LogMyTime http://www.logmytime.de/	◐
Odoo OpenERP http://www.ife.de/	●
TimeNote http://www.timenote.de/	◐
timr http://www.timr.com/	◐
salesking	
Exact Online http://www.exactonline.de/	●
Salesforce https://www.salesforce.com/	○
SuperOffice CRM Online http://crmonline.superoffice.de/crmonline_062014-1/	◐
CRM on Demand http://www.crm-on.de/	◐
projectfacts https://www.projectfacts.de/	●

Legend: ○ unsuitable, ◐ partially suitable, ● suitable

VI. Conclusion

The contribution of this paper is a technique for the automated identification of required (cloud) service functions. We have implemented the approach in a system for the automated identification and extraction of required cloud service functions in the crafts domain. The system is based on a domain ontology, which was manually built up by collecting the functions listed on the websites of several cloud services for the crafts domain. We have evaluated the system by measuring its precision, recall and F1 score according to two different test cases. In the first test case the system has identified the functions and interfaces listed on the websites the domain ontology was built up from. In the second case the system had to examine the functions and interfaces from websites of other crafts software as control group. For the control group websites, we have got an average value for recall of

0.74. This evaluation result is sufficient for a system, which supports users to identify required service features. Moreover, we have performed a cloud service search by using the elicited functional service requirements as keywords for running a Google search. The results of the Google searches for the control group services have returned adequate cloud services. Thus, we could demonstrate the suitability of our approach for functional requirements elicitation to support searches for suitable cloud services.

VII. Future Work

There are some ideas which can improve our approach and which are planned for future work. Currently, the users can only enter an URI or keywords into the system for identifying required (cloud) service features. We plan to offer the possibility to process a keyword search after having received the results for an URI-based search to provide an iterative search process to the users. Another idea is the possibility to search for the required features of several software. That means the users can enter the URIs of multiple software websites to get the all functions and interfaces offered by all software tools together. An additional consideration is to extend the domain ontology automatically by software descriptions of a repository or by user input. The case of extending the ontology based on user input could be implemented into the user interface. A possibility could be to implement a dynamic result where the users can add own classes of functions and interfaces in form of terms into the hierarchy of the result list. Thus, the terms entered by the users could be automatically added into the hierarchy of the domain ontology. As mentioned in Section IV-B we plan to reduce the waiting time for the users e.g. by incremental result generation or a parallel programming approach. The final version of the system shall be evaluated for its practical usage by running a user study.

Acknowledgement

This work was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference “01MD11041”. The authors take the responsibility for the contents.

References

[1] Athens Technology Center, “Artist Project Website”, 2015 retrieved on March 8, 2015 from http://www.artist-project.eu/.

[2] The Cloudbook Community, “cloudbook.net”, 2013 retrieved on March 8, 2015 from http://www.cloudbook.net/directories/product-services/cloud-computing-directory.

[3] CloudSearchPortal, “CloudSearchPortal.com”, 2015 retrieved on March 8, 2015 from http://www.cloudsearchportal.com/.

[4] H.-J. Happel , A. Korthaus, S. Seedorf and P. Tomczyk, “KOntoR: An Ontology-enabled Approach to Software Reuse”, Proceedings of the 18th International Conference on Software Engineering and Knowledge Engineering, 2006, pp. 349-354.

[5] J. Kang and K. M. Sim, “A Cloud Portal With a Cloud Service Search Engine”, Proceedings of the 2011 International Conference on Information and Intelligent Computing IPCSIT, vol. 18, IACSIT Press, Singapore, 2011, pp. 1-8.

[6] V. S. K. Nagireddi and S. Mishra, “An Ontology Based Cloud Service Generic Search Engine”, Proceedings of the 8th International Conference on Computer Science & Education (ICCSE), 2013, pp. 335-340.

[7] T. H. Noor, Q. Z. Sheng, A. Alfazi, A. H. H. Ngu and J. Law, “CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web”, Proceedings of the 20th International Conference on Web Services (ICWS), 2013, pp. 443-450.

[8] N. F. Noy and D. L. Mcguinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, tech. report, Stanford University, 2001.

[9] REMICS, “REMICS Project Website, Reuse and Migration of legacy applications to Interoperable Cloud Services”, 2011 retrieved on March 8, 2015 from http://www.remics.eu/home.

[10] M. Á. Rodríguez-García et al., “Semantic Annotation and Retrieval of Services in the Cloud”, Distributed Computing and Artificial Intelligence, vol. 217, Springer International Publishing, 2013, pp. 69-77.

[11] R. Sahandi, A. Alkhalil and J. Opara-Martins, “Cloud Computing From SMEs Perspective: A Survey-based Investigation”, Journal of Information Technology Management, vol. 24, No. 1, University of Baltimore, 2013, pp. 43-49.

[12] Stanford Center for Biomedical Informatics Research, Stanford University, “Protégé”, 2014 retrieved on March 8, 2015 from http://protege.stanford.edu/products.php.

[13] THINKstrategies, “Cloud Computing Showplace”, 2012 retrieved on March 8, 2015 from http://www.cloudshowplace.com/.

[14] M. Zhang et al., “Investigating Techniques for Automating the Selection of Cloud Infrastructure Services”, International Journal of Next-Generation Computing, vol. 4, no. 3, 2013, pp. 759-764.

Temporal Exception in Web Service Composition

Bey Fella, Bouyakoub Samia, Belkhir Abdelkader
Institute of Electronics & Computer Sciences
USTHB University
Algiers, Algeria
Bey-Fella@hotmail.com
belkhir@lsi-usthb.dz
bouyakoub.s@gmail.com

***Abstract*— Web service composition is studied by many works and constitutes the heart of a great research activity. However, the majority of this work does not take into account temporal exception handling. Consequently, the results do not answer the needs and the temporal preferences of customers and suppliers. Incorporating temporal constraints in Web service composition results in a more complex model and addresses the crucial need for exception handling. In this paper, we present H-Service-Net model for Web service composition and policies of handling exceptions. We validated our proposed approach in an implementation called H-Service-Editor tool.**

Keywords-Composition of Web services; Petri network; temporal constraints; handling of exception.

I. Introduction

Due to its capability for dynamic composition and easy reuse, Service Oriented Architecture (SOA) has become a popular framework for software development in many application domains. An important process in SOA is service composition [9].

A major part of the interest created by so-called Web services is their possibility to distribute processing capabilities across a number of loosely coupled functional entities communicating over standardized messages [6]. Companies such as Google, Amazon or PayPal have been making an increasing part of their revenue by exposing and selling their functionalities as instances of services, reusable as background components by third-party application developers [6].

A composite Web service invokes one or more other Web services and combines their functionality. In contrast, a Web service that does not invoke other Web services is called a basic Web service. The process of developing a composite Web service is referred to as a Web service composition [10].

The objective of Web services composition is to determine a combination of services according to the customer's request. This composition will seem to the customer as a single service because it is transparent to him. In composition, Web services collaborate by exchanging messages. In addition to this exchange, other factors affect this composition. We are interested in the time factor which is crucial and at the same time very complex.

Due to failures that can occur using Web services and their composition, several solutions of exception handling have been proposed in order to recover from these exceptions.

Exceptions are critical in Web services. Therefore, it is essential to take into account the handling of exceptions, especially if their execution relates to the continuation of the composed service. The rest of this paper is organized as follows. Section II defines some related work. Section III presents our model called H-Service-Net. Section IV addresses the policies of handling exception with a demonstrative example. Section V presents the H-service-editor tool, its architecture and usage. Conclusions close the article.

II. Related work

In the literature, several theories have been proposed to explain exception handling, which is a critical case in Web services, therefore, to have a consistent execution, it is essential to take into account exceptions. In service composition it is important to define the mechanism of handling exceptions in order to have a coherent and consistent composition even in the presence of exception.

A number of approaches have been proposed to deal with exception handling in Web service composition. First, Caoqing et al. [1] describe the integration model which contains the normal process and exception handling logic based on Petri net integration technologies. The result shows that this model can realize formal modeling of exception handling in Business Process Execution Language (BPEL), and further provides support for analysis and verification of properties relating to exception handling. A novel architecture for exception handling has been proposed by Thirumaran et al. [2] for focusing upon and verifying the manageability of a web service. But this solution does not provide methods of handling exceptions, it only detects if an exception is manageable or not. The approach proposed by Wang et al. [3] describes the policy-based exception handling approach for BPEL processes, which is a new framework designed for exception handling in BPEL processes to provide a flexible language mechanism. The approach of Erradi et al. [5] proposes a set of extensible recovery policies to declaratively specify how to handle and recover from exception in Web services composition. The identified constructs were incorporated into a lightweight service management middleware named Manageable and Adaptive Service Composition (MASC) to transparently enact the fault management policies and facilitate the monitoring. Hamadi et al. [4] propose Self-Adapting Recovery Net (SARN), an extended Petri net model, for specifying exceptional behavior in business processes. SARN adapts the structure of the underlying Petri net at run time to handle exceptions. The approach of Christos et al. [7] introduces the Service Relevance and Replacement Framework (SRRF) whose main concept is resolving exceptions by finding relevant Web tasks, exploiting qualitative and functional metadata semantics. Finally, Lau et al. [8] propose an approach to server-side exception handling by composite Web services that capture Unavailability and Time-out exceptions and provide Retry as recovery action, or Throw to propagate exceptions.

Exception handling is generally tedious and error prone. The issue of exception handling has not been carefully considered in existing service composition works [9].

Although several studies have indicated that exception handling is critical, little attention has been given to time out exception.

III. H-SERVICE-NET: A TEMPORAL MODEL FOR WEB SERVICE COMPOSITION

The H-Service-Net model (an acronym for Hierarchical Service Net) is a time Petri net-based model. It allows the modeling of time-critical aspects in the field of Web Services. It allows incremental composition of services, as well as consistency checking after each modification. It introduces a new type of places named composite places. A composite place is an abstract place represented by a sub-network, allowing a degree of independence between the parts of the H-Service-Net. Indeed, a composite or single place in H-Service -Net depends only on the subnet to which it belongs. In other words, the modification of a component can affect its subnet or the subnets of the same hierarchy. This representation allows for incremental modeling of the H-Service-Net. This will allow for easy correction of errors, an exact location of conflicts between the subnet elements and support rapid changes. Thus, the H-Service-Net model is well suited for modeling the synchronization constraints in a temporal scenario. As a result, it was chosen to model the composition of Web services. We present in what follows the different elements of the H-Service-Net model:

A. Places in H-Service-Net

There are two main types of places of H-Service-Net: simple and composite." Then, Table I shows places in H-SERVICE-NET, simple place are modeled by circle shape and composite place are modeled by triangle shape.

TABLE I. PLACES IN H-SERVICE-NET

Place	Modeling	Description
Ordinary Place	○	It models a basic element (Web service) and its execution time.
Virtual Place	◌	It models a temporal constraint.
Silent Place	◎	It models a place without any specific task, which is used to handle exceptions.
Parallel Place	△ P	It models a set of elements of the same group that operate simultaneously, which is used to model concurrent Web service.
Sequential Place	△ S	It models a sequence of elements of the same group that run in sequence. It is used to model sequential Web services.
Root Place	△ R	It represents the root of the global Petri net and behaves like a sequential composite place.
Loop Place	△ L	It is an element that runs in a loop and is used to model a recursive Web service.

An H-Service-Net can be seen as a tree where the parallel and sequential composite places represent intermediate nodes, and atomic places associated with Web services are the leaves.

B. Transitions in H-Service-Net

An operation performed by a Web service is modeled by a transition. Table II shows transition in H-SERVICE-NET that define the different termination semantics :

TABLE II. TRANSITIONS IN H-SERVICE-NET

Transition	Modeling	Description
Simple transition	▌	It is fired when all its input places are active and have available tokens.
Master transition	M	It is fired as soon as the place associated with the Master arc is active and has an available token.
First transition	F	It is fired when one of its input places is active and has a free token.

C. Tokens and arcs in H-Service-Net

H-Service-Net defines state tokens and exception tokens and there are two types of arcs in the model, then Table III shows tokens and arcs in H-Service-Net:

ISBN: 978-1-61208-407-7

TABLE III. TOKENS AND ARCS IN H-SERVICE-NET

Arc \ token	Modeling	Description
State Token	●	It defines the state of the Web service associated with the place
Exception Token	▲	It is used to handle exceptions
Simple arc	→	Control the firing of a simple transition
Master arc	➡	Control the firing of a Master transition.

The modeling of Web services in H-Service-Net is given as follows:

— A simple or atomic Web service is represented by an ordinary simple place.

— A composite Web service is represented by a sequential, parallel or loop composite place.

D. *Example of the TimeOut policy:*

Let us consider the following scenario:

A person wants to go on holiday, and wants to make a campsite for two days in a forest at the end of the holidays. Inclusive vacations are offered by a travel agency that can offer a composition of the following Web services for this request:

Ws1, Ws6: two Web services of payment by credit card.
Ws2: Visa Service.
Ws3: Sales service ticket travel agency.
Ws4, Ws5: two Services of booking rooms in two different hotels.

Figure 1 shows the time constraints (date) in the network H-Service-Net of the example.

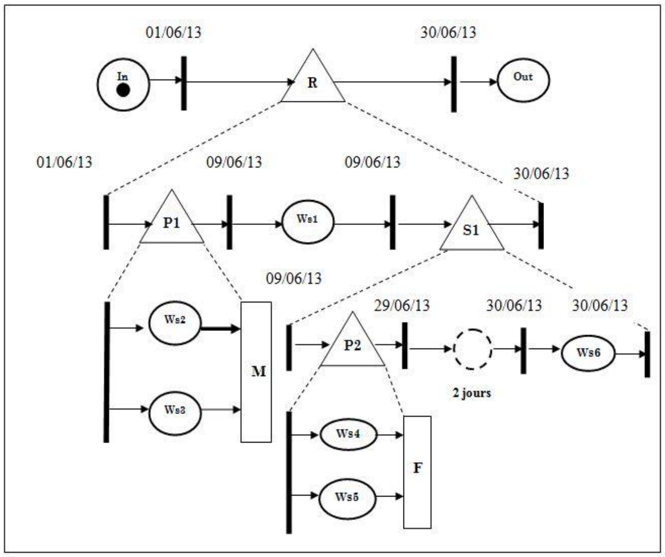


Figure 1. Temporal Constraints in H-Service-Net for the example.

For the modeling of these Web Services in a single H-Service-Net, we add the following composite places:

- The visa services ws2 and the sales service ticket travel agency ws4 can run in parallel, so we represent them in H-Service-Net by a parallel composite place P1.
- The Web Services of booking rooms ws4 and ws5 can run in parallel, so they are modeled in H-Service-Net by a parallel composite place P2.
- The composite services P2 and Web Payment service banking ws6 are modeled with the sequential composite place S1.
- The Composite services P1 and S1, and the Web Service Payment by credit card ws1 run in sequence, so we have modeled this set by the root place R.

IV. SOLUTION TO TEMPORAL VIOLATIONS BY A HANDLING OF EXCEPTION

Exception handling is critical in Web services composition. It is essential to consider exception handling for the robustness of the composition, especially if the exception relates to the continuation of the composition.

In this section, we present the policies of exception handling in H-Service-Net. For each exception, a set of methods is executed to handle the exception. We were inspired by Rachid et al. [4] to define the methods of exception handling. Table IV shows all used methods:

TABLE IV. BASIC OPERATIONS FOR EXCEPTION HANDLING

Method	Description
CreatePlace(namews , type-place, type-transitionIn, type-transitionOut)	Allows creating a simple or composed place with an input transition and output transition.
CreateSilentPlace(nameSilentPlace , type-transitionIn, type-transitionOut)	Allows creating a silent place without any task between two types of transition.
AddExceptionToken (place)	Allows adding an exception token to a silent place.
DisableWS(ws)	Allows canceling the running Web service.
AddTo(P, ws)	Allows to add the place ws to the composed place P
AddBefor (ws, ws’)	Allows adding the place ws' before the place ws.
AddAfter(ws, ws’)	Allows adding the place ws' after the place ws.
ReplacePlace(ws ,ws’)	An existing place ws is replaced by another place ws'.
RemoveToken(ws)	A token of state is removed from the Web service ws.

A. *The TimeOut policy:*

In a Web service composition, both the client and the service provider may have temporal constraints in their interaction. For example, the service provider may cancel the service if it does not receive a response after a time T_max and similarly for the service requester.

To handle the time out exception, each Web service is associated with a time limit T_max. If the execution of a Web service exceeds this time limit, a time-out exception is triggered.

A policy of handling TimeOut (WebServiceWsTO, time T) exception is defined in Figure 2:

PSEUDO CODE 1. The TimeOut policy

```
1. DisableWs(WsTO). //Cancels the Web service which reached the timeout.
2. Compensate (WsTO) or RollBack (WsTO). //Optional
3. RepeatAfter(WsTO). //Repeat the Web service which has reached the timeout (defined in the next section).
4. Compensate (WsTO) or Rollback (WsTO). // Optional.
5. RepeatAfter(WsTO, 2*T).// If another exception of time out is raised, Repeat the Web service till the doubling timeout expired.
6. Compensate(WsTO) or Rollback (WsTO). // Optional
7. OtherWS(TimeOut, WsTO, OtherWsTO). //If another exception of time out is raised, search another alternative service (defined in the next section).
```

Figure 2. Pseudocode of the TimeOut policy.

In order to handle the time out exception, the Web service which has reached a timeout is cancelled. Then, the Web service is repeated using the RepeatAfter policy and if another timeout exception is captured Repeat the Web service until the doubling timeout expired and expect that the service will not miss this deadline with T = 2* T (by doubling the time out). If this strategy does not work, the exception can be handled by calling another equivalent and compatible Web service.

Compensate(WsTO) : undoes the effect of task execution result of the Web service (WsTO).

In compensation, information must be added in the SOAP header in order to define that the Web service can be compensated or not, <nsto :compensate enable= “True” /> or <nsto :compensate enable= “False” /> .

Rollback functionality RollBack (WsTO) is executed in the reverse order of their forward execution and rolled back to (WsTO) original state.

The difference between compensate and rollback is that compensate is a Forward Recovery and rollback is a Backward Recovery, compensation or rollback are not require in read-only Web service.

Compensation is run on the service provider's side by a call to service. In our solution, compensation is considered as a service call cancellation

If the temporal constraint assigned to the Web service is higher than the execution time of the timeout policy (RepeatAfter (Ws, T) + RepeatAfter (Ws, 2*T)) then all the timeout policy is executed. Otherwise if the temporal constraint is higher than the execution time of the RepeatAfter (Ws, T) policy then only the stage one, two, three, four and seven are executed in order to satisfy the temporal constraint. In default only the stage one, two and seven are executed to handle the critical Web service.

B. *The RepeatAfter policy:*

When an exception event is captured, the policy of handling RepeatAfter allows to repeat the execution of a Web service after its execution. A policy of handling exception RepeatAfter (Exception e, WebServiceRepeatWs), when a corresponding event of exception e is captured is defined in Figure 3:

PSEUDO CODE 2. The RepeatAfter policy

```
1. CreateSilentPlace (S, simpleTransition, simpleTransition). // Create a silent place S between two simple transitions
2. CreatePlace (Sra, Seq, simpleTransition, simpleTransition) // Create a new sequential composed place Sra between two simple transitions.
3. ReplacePlace (RepeatWs, Sra). // Replace the composed place RepeatWs by the place Sra.
4. RemoveToken (RepeatWs). // A token of state is removed from the place RepeatWs .
5. AddExceptionToken (S). // Add an exception token to a silent place S.
6. AddBefor (RepeatWs, S). // Add the place S before the repeated place RepeatWs and run the new H-service-net in order to handle the exception.
```

Figure 3. Pseudocode of the repeatAfter policy.

C. *The OtherWs policy*

The failure of a Web service Ws requires a search for another Web service OtherWs offering, at least, the same functionalities. The policy of handling OtherWs allows running another alternative Web service in case the Web service fails.

A policy of handling of exception OtherWs (Exception e, WebServiceWs, WebServiceotherWs) when a corresponding event of exception e is captured is defined as in Figure 4:

PSEUDO CODE 3. The OtherWs policy

```
1. DisableWs (Ws).  // Cancel the fail Web service Ws.
2. CreatePlace (otherWs, simple, simpleTransition, simpleTransition). // Create a new place OtherWs between two simple transitions.
3. RemoveToken (Ws). //Remove the token of state in the place Ws
4. AddExceptionToken (otherWs). // Add a token of exception to the place otherWs.
5. CreatePlace (So, seq, simpleTransition, simpleTransition) // Create a new composite place So between two sequential simple transitions
6. ReplacePlace (Ws, So). // Replace the place Ws by the composite place So.
7. AddTo (So, Ws). //Ajouter the service Web to be cancelled in the made up place So.
8. AddAfter (Ws, otherWs) // Add after the place Ws the place otherWs and run the new H-service-net in order to handle the exception.
```

Figure 4. Pseudocode of the OtherWs policy.

During the execution of the network in Figure 1, an exception of time out is raised from the flight ticket reservation service Ws3. In order to handle the TimeOut exception first, the Web service Ws3 is cancelled, and then a repetition of the same service is performed using the RepeatAfter policy and the network H-Service-Net will become like Figure 5.

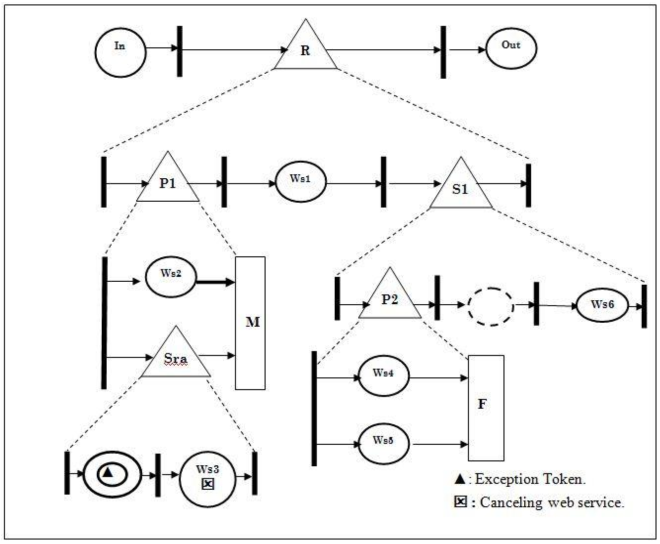


Figure 5. H-Service-Net before the execution of RepeatAfter policy.

After the execution of the RepeatAfter policy if another exception of time out is raised, the TimeOut policy calls another Web service using the OtherWS policy and the H-Service-Net network will become as Figure 6. We note that during the execution of the exception, the token of exception has become a token of state.

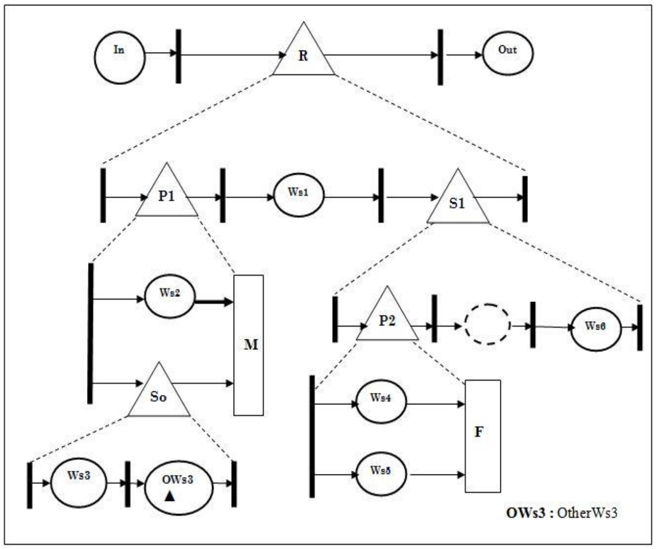
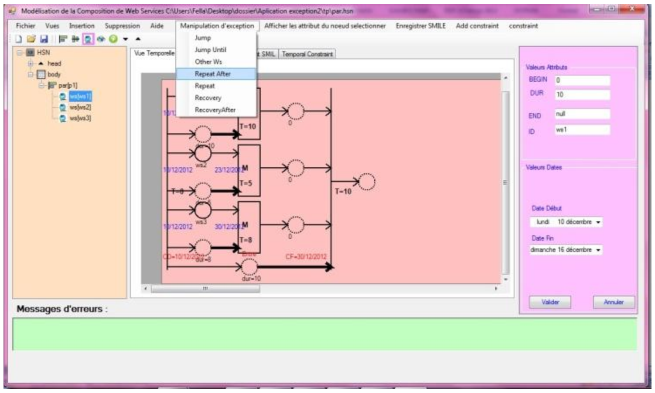


Figure 6. H-Service-Net before the execution of OtherWS

V. IMPLEMENTATION

We developed H-Service-Editor to illustrate the viability of the proposed composition and exception handling techniques presented above. H-Service-Editor is a Web service composition modeling tool with simulation capability; it supports the creation of policies for handling exceptions through the System of handling exception as depicted in Figure 7.



BEGIN 0
DUR 10
END null
ID ws1

Date Début: lundi 10 décembre
Date Fin: dimanche 16 décembre

Figure 7. Depicts a screen shot of H-Service-Editor

The authoring environment offers different views that allow the composition of Web services and handling of exception. The different views are:

— The graphical view: it displays the handling of exception for all the policies introduced in this article in the form of H-Service-Net model. The user can view the different sequences and temporal relations between simple or composite Web services before and after the manipulation.

— XML View: it displays the composition file of the handling of exception in the Extensible Markup Language (XML) standard format document

— The hierarchical view: It allows the representation of services composition in the form of a tree structure. The different services are represented in this area by hierarchical structures similar to the H-Service-Net.

— Error message view: if any temporal conflict is found, the tool displays an error message in order to offer earliest error detection within the editing process of handling exception.

VI. CONCLUSION AND PERSPECTIVE

In order to complete the H-Service-Net model, we have proposed to apply to the H-Service-Net model a manipulation of exception in a hierarchical way. This allows an easy management of time out exception while maintaining the fundamental design of Petri net simple and easy.

Finally, an application (H-Service-Editor tool) of the proposed approach is presented for the modeling of all the policies of handling exception based on our model.

Until now the policies of handling exception are executed manually. The next stage of our research will be to automate the execution of the policies. Further research is needed to consolidate this approach.

REFERENCES

[1] J. Caoqing, Y. Shi, H. Shanming, X. Hui and Q. Yueming “A Formal Model for Exception Handling in BPEL Process”, 2nd International Conference on Computer Science and Network Technology, 2012.

[2] M. Thirumaran, P.Dhavachelvan, K.Seenuvasan and G.Aranganayagi, “A Novel Approach for Run Time Web Service Exception Handling”, Third International Conference on Advanced Computing, 2011.

[3] Q. Wang, S. Ying, J. Wen, and G. Lv ,“Policy-based exception handling for BPEL processes”, IEEE International Conference on Information Science and Technology Wuhan, Hubei, China, 2012

[4] H. Rachid, B. Boualem and M. Brahim,“Self-adapting recovery nets for policy-driven exception handling in business processes” Journal Distributed and Parallel Databases vol. 23 Issue 1, pp. 1 – 44, 2008.

[5] A. Erradi, P. Maheshwari, and V. Tosic. “Recovery policies for enhancing Web services reliability.” In IEEE International Conference on Web Services (ICWS’06), 2006.

[6] S. Hall “Model-Based Simulation of SOAP Web Services From Temporal Logic Specifications Sylvain”, 16th IEEE International Conference on Engineering of Complex Computer Systems, 2011.

[7] K. Christos, V. Costas, G and Panayiotis1, “Towards Dynamic, Relevance-Driven Exception Resolution in Composite Web Services” University of Athens ,University of Peloponnese, 2006

[8] K. Lau and C. Tran ,“Server-side Exception Handling by Composite Web Services”, School of Computer Science, the University of Manchester 2008

[9] J. Huang, W. Zhu and J. Fu “Automated Exception Handling in Service Composition Using Holistic Planning”, IEEE 15th International Conference on Computational Science and Engineering, 2012.

[10] R. Prakash, R. Raja “Evaluating Web Service Composition Methods with the Help of a Business Application”, International Journal of Engineering Science and Technology Vol. 2(7), pp. 2931-2935, 2010.

Simulation of Web-Based Multi-Modal Transportation with Multi-Criteria Walking for Smart Cities

Monsak Socharoentum
National Electronics and Computer Technology Center
National Science and Technology Development Agency
Pathumthani, Thailand
e-mail: monsak.soc@nectec.or.th

Hassan A. Karimi
Geoinformatics Laboratory, School of Information
Sciences, University of Pittsburgh
Pittsburgh, USA
e-mail: hkarimi@pitt.edu

***Abstract*— While current transportation simulations evaluate vehicle trips or neighborhood walkability, none can be utilized to evaluate trips that require multi-modal transportation when walking is always one mode. In this paper, we address this gap by introducing the Multi-Modal Transportation (MMT) with Multi-Criteria Walking (MMT-MCW) concept. MMT-MCW simulation can be used to evaluate various aspects of smart cities, such as walkability. The premise of MMT-MCW is based on the observations that: (a) walking can be performed for other purposes besides merely reaching destinations, such as to maintain or improve health and (b) traveler's characteristics and preferences play an important role in determining optimal route choices. Selected MMT-MCW scenarios were used to evaluate walkability of several cities with respect to three criteria: inter-modal transfer locations (parking lots and bus stops), elevation of walking routes; and walking distance. Results of simulations using these criteria are discussed and analyzed.**

Keywords-smart cities; walkability; multi-modal transportation; routing; multi-criteria walking.

I. INTRODUCTION

Multi-Modal Transportation (MMT) with Multi-Criteria Walking (MCW) is proposed as a new concept where walking is always considered as one mode and can be performed for other purposes (multi-criteria) in addition to travelling to a destination. Two sets of factors impacting MCW are environmental factors and traveler factors. Environmental factors, when compared to driving cars or riding public transportation, may have a greater impact on walking. For example, people may prefer driving cars or riding buses over walking due to rain, snow, hilly terrain, or air pollution. Location is also an environmental factor that influences walking, for instance, fastest walking routes may be based on flat and short routes which take priority over steep and longer routes. However, when walking is for exercise, the steeper and/or longer route may be preferred. Traveler factors, such as individuals' characteristics, also have an impact on choosing walking routes [15]. Several studies, such as [11], reported a correlation between individual behaviors and walking. Studies by Leslie et al. [8] and Ewing et al. [3] are examples related to the urban area evaluation in terms of neighborhood walkability. Despite the benefits of MMT-MCW for evaluating transportation options in smart cities, currently, there is no research that is focused on evaluation of city' transportation infrastructures and utilities (e.g., parking locations and walking routes).

To fill this gap, MMT-MCW simulation is proposed to evaluate three basic options: (a) inter-modal transfer locations (parking lots and bus stops); (b) elevation of walking routes; and (c) walking distance. The first option is related to MMT, and the latter two are related to MCW.

MMT-MCW may be implemented in several ways for smart cities, for example, as a new service for individuals interested in finding routes that include walking components. [6] developed a prototype service (called Route2Health) that recommends walking sessions, if feasible, for any trip. By taking origin, destination, and traveler's individual conditions as input, Route2Health recommends a sequence of transportation modes along with specific details about each mode that is most optimal (personalized). MMT-MCW can also be implemented to simulate the design and evaluation of smart cities. Existing surveys, analyses, simulations on or related to neighborhood walkability and urban design, such as [2] - [5][7][9][10][14][15], but none addresses the issues and scenarios as those possible with MMT-MCW.

The paper's contribution is a novel integration of new and existing Web techniques and technologies for evaluating and analyzing transportation options for smart cities. An information management tool (simulation) is developed to analyze mushing up data and find solutions based on existing Web services (Google Map APIs). The rest of the paper is organized as follows. Section 2 describes MMT-MCW. Sections 3 and 4 discuss MMT simulation and its results. The paper ends with a summary and suggestions for future research in Section 5.

II. MMT-MCW

MMT-MCW is concerned with finding: (a) multi-modal transportation routes with walking as one mode and finding (b) optimal walking paths by considering multiple criteria. Walking transfer node and route score are the two main factors that MMT-MCW considers in finding optimal solutions (routes).

Three modes of transportation are considered in MMT-MCW: walking, driving, and riding (bus). We define "walking transfer node" as a location where travelers switch from a pedestrian network to a vehicular network, or vice versa. In MMT-MCW, walking transfer nodes play an important role in finding suitable (personalized) routes. For example, change of one parking lot to another (as a walking transfer node) may result in a different (and desired) solution. With respect to public transportation, the choice of a bus stop (as a walking transfer node) determines a specific bus route. To identify a suitable walking transfer node, traveler's desired walking distance is separated into estimated upper and lower limits. The upper limit excludes walking transfer nodes that are located beyond a traveler's maximum preferred distance. The lower limit excludes walking transfer nodes that are located closer than the desired minimum walking distance. Accordingly, one or more suitable walking transfer nodes are identified.

Route score is used to quantify suitability of a walking route in meeting traveler's criteria. To compute a route score, a relevant criterion must be identified and used to formulate its associated metric function. Examples of route score computation are based on: (1) traveler's desire to burn a desired amount of calories by walking and (2) traveler's preference for a certain level of elevation variation. Accordingly, two route scores are required: (1) calories burnt on walking and (2) elevation variation.

To calculate the calories burnt on walking, the ACSM walking equation [12] can be used:

$$EE = (0.1 \cdot S + 1.8 \cdot S \cdot G + 3.5) \cdot BM \cdot t \cdot 0.005 \qquad (1)$$

where *EE* is walking energy expenditure (kilocalories), *S* is walking speed (meters/minute), *G* is grade (slope) in decimal form (e.g., 0.02 for 2% grade), *BM* is traveller's body weight (kilograms), and *t* is walking time (minutes).

To calculate elevation variation, walking surface roughness is used. The walking surface roughness refers to the standard deviation of the elevations along an entire walking route. The standard deviation of a flat walking route is zero, and the higher value of walking surface roughness refers to higher variation of elevations along the walking route.

III. Simulation and Data Collection

MMT-MCW scenarios for several geographic areas (cities) in the US were simulated. The two attributes used to categorize and select the cities were population density and elevation range. Population density was simulated to explore the influence of high and low population density on walking routes and walking transfer nodes. Different elevation ranges were simulated to explore the effect of topography on walking routes and walking transfer nodes. The US Office of Management and Budget uses population density to define a statistical area. A statistical area contains one or more cities (and/or counties) and can be classified as metropolitan (high-density population) or micropolitan (low-density population). Elevation range was classified into hilly (elevation range ≥ 100 meters) and flat (elevation range ≤ 50 meters); where elevation range = max. elevation - min. elevation. The two threshold values (50 and 100 meters) were chosen for separating between hilly and flat terrains.

Two different MMTs were simulated: driving-walking and riding-walking. A driving-walking trip usually comprises (in sequence) driving, parking, and then walking, and return in the reverse sequence. Unlike the driving-walking, travellers do not have to begin with the vehicular (riding) mode in a riding-walking trip. The trip may start by walking from origin to a nearby bus stop then taking bus to destination. Walking can also be in the middle to connect two different bus routes, and the return trip can be in any sequence. For simplicity, the return trips were not considered, and walking was assumed as the mode connecting the walking transfer nodes and destinations. To this end, walking transfer nodes and walking routes were used in the simulation. The vehicular route computation between origin and walking transfer node was not considered since it is not the MMT-MCW's main contribution. Parking lots, bus stops, walking routes, sidewalk slopes, and points of interest in several cities were considered in the simulation. The data, programs, and parameters used in the simulation are described below.

The desired walking distance between walking transfer node and destination was assumed to be one kilometer. Point of interest (POI) locations were selected from OpenStreetMap [1], and 100 destinations within each city were randomly selected (in case the number of POIs in a city was less than 100, all POIs were used). To identify suitable parking lots, a buffer (inner radius: 0.5 kilometer; outer radius: 1.5 kilometer) around the destinations of interest was created. For each destination, up to 20 parking lots within a buffer were selected as suitable walking transfer nodes (note that 20 here is an arbitrary number and a suitable number may be determined for each). Bus stops and bus routes data were collected from Google Transit Feed Specification [16]. For each suitable parking lot and bus stop, up to three candidate walking routes were generated (ordered by their travel time). Parking lot locations and walking routes were retrieved from Google Place API and Google Direction API, respectively. Once all candidate routes were computed, elevation of points along the walking route of interest is retrieved from Google Elevation API, and then (1) was used to find calories burned for each candidate walking route. Walking surface roughness was also calculated using the elevations of route segments. To simulate multiple traveler's characteristics, four body weights (60, 80, 100, and 120 kilograms) and three walking speeds (60, 80, and 100 meters/minute) were used.

IV. Simulation Results

Figure 1 shows the selected cities (on X-axis), based on the criteria discussed in Section 3, the numbers of destinations and the counts of destinations that have one or more suitable parking lots (on Y-axis). The following abbreviations are used in Figure 1: Micropolitan (Mi), Metropolitan (Me), Hilly (H), and Flat (F). Most cities in metropolitan areas have a large number of destinations with suitable parking lots except Bossier and McAllen. Four cities (Barre, Kappa, Scottsbluff, and Bossier) have zero or only one destination with a suitable parking lot, which are considered outliers and excluded from the analyses.

ISBN: 978-1-61208-407-7

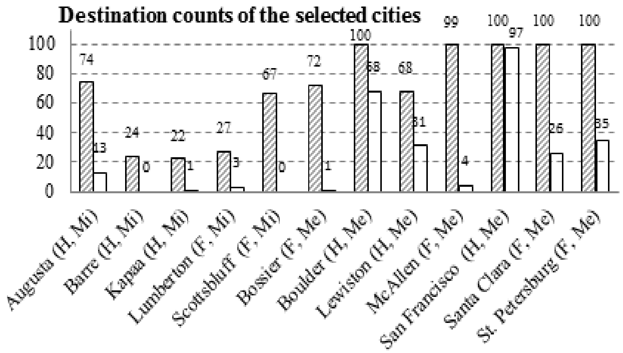


Figure 1. Number of destinations and the counts of destinations that have ≥ 1 suitable parking lots.

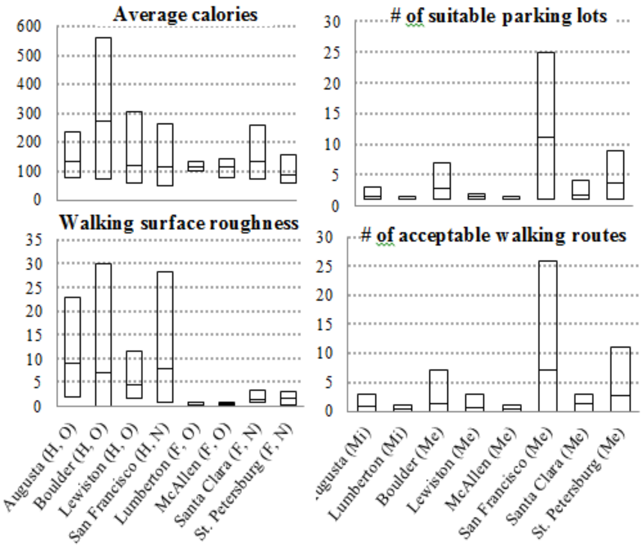


Figure 2. Comparison of attributes related to PK routes in different cities.

Figure 2 shows maximum, minimum, and average number of calories burned (top left) and walking surface roughness (lower left) for walking routes that connect to parking lots (PK routes). On X-axis, the first four cities are hilly, and the latter four are flat. The graphs indicate that hilly cities have wider ranges of both calories burnt and walking surface roughness. This is because both calories burnt and walking surface roughness are directly related to the elevation range of hilly cities and the walking routes. An interesting observation is that most cities (except Boulder) in the left figure have a similar average calories burnt regardless of the elevation range. Although Boulder has a similar walking surface roughness compared to other hilly cities, its average calories burnt is significantly higher than the others. This indicates that walking routes in Boulder are better in terms of burning calories.

Figure 2 (top right) shows the number of suitable parking lots and (lower right) shows the number of acceptable walking routes. The acceptable walking routes refer to walking routes that have their distance fall within 0.9 and 1.1 kilometer (±10% of the 1 kilometer desired walking distance). The graphs show that San Francisco has the highest average number of suitable parking lots, the highest number of acceptable walking routes, and the largest range on both attributes (largest variation of results); this is expected for a metropolitan city where transportation infrastructures are dense. Note that San Francisco is the 13th most populous city in the United States [13].

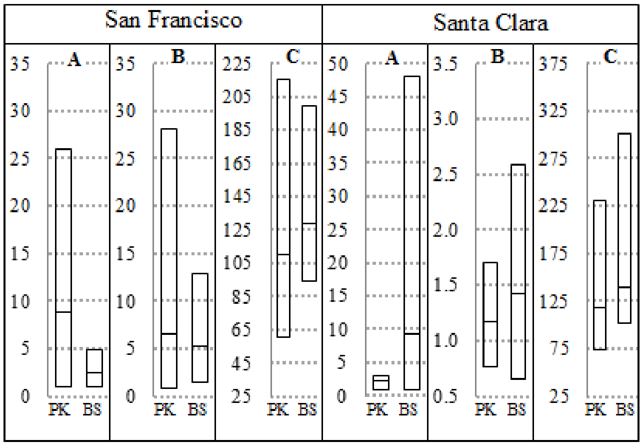


Figure 3. Comparisons between PK routes and BS routes; A: Number of acceptable walking routes; B: Walking surface roughness; C: Average calories burnt.

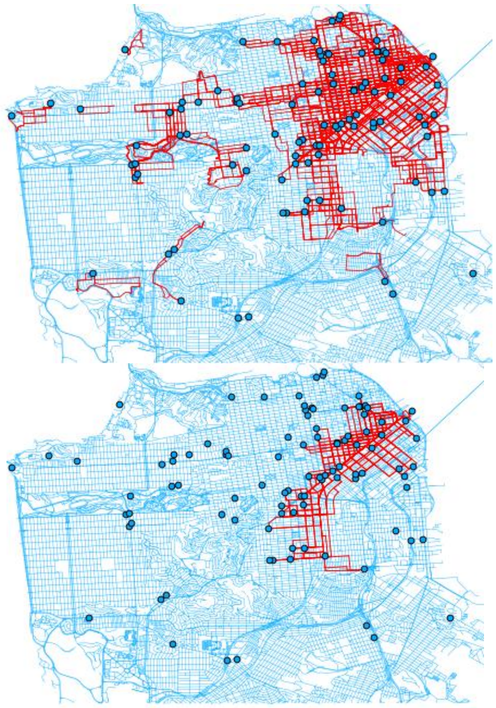


Figure 4. San Francisco: PK routes (upper) and BS routes (lower).

Figure 3 shows the comparisons between PK routes and BS routes (walking routes that connect to bus stops) in San Francisco and Santa Clara which were the only two cities (among the selected cities) that publish their transit data.

Each bar graph represents maximum, minimum, and average values. Considering the number of acceptable walking routes in San Francisco and Santa Clara, the PK routes in San Francisco have higher average value than BS routes, while the opposite behavior is revealed in Santa Clara's graphs. This indicates that PK routes and BS routes are not necessarily correlated. Considering walking surface roughness, BS routes in both cities have narrower ranges than PK routes, meaning that BS routes in both cities are more similar with respect to walking surface roughness. BS routes in both cities are also similar with respect to amount of calories burnt. PK routes in San Francisco have lower average calories than BS routes, and vice versa for Santa Clara.

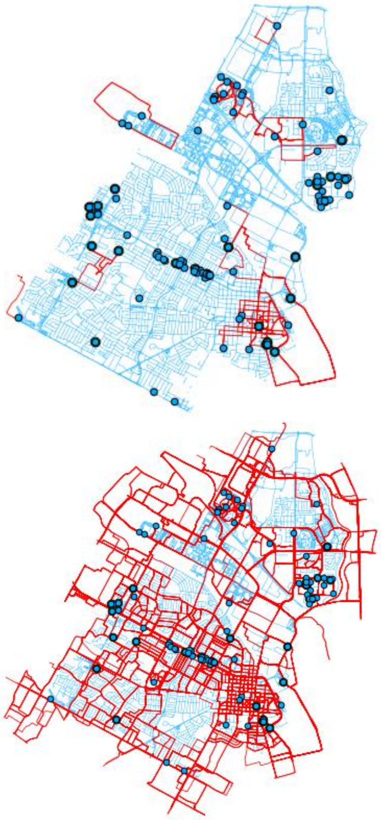


Figure 5. Santa Clara: PK routes (upper) and BS routes (lower).

Figures 4 and 5 show the spatial distribution of destinations and the coverage of PK and BS routes overlaid on the cities' road network. The maps indicate that PK routes have more coverage than BS routes in San Francisco and vice versa in Santa Clara.

V. Summary and Future Research

This paper presented a new simulation approach for evaluating smart cities. Scenarios in some cities were simulated. The simulation results show that: (a) despite similar elevation range, cities may have significantly different average calories burnt for the walking routes generated and (b) two cities in metropolitan areas (San Francisco and Santa Clara) show that PK routes and BS routes are not necessarily correlated.

Considering that enhancing health and wellbeing of people, among other things, is one objective for building smart cities, our proposed approach can be used to evaluate smart cities for their environment infrastructures (roadways and sidewalks) and transportation infrastructures (different modes) and as a simulation tool to design new smart cities. Some future research directions are:

- Investigating and developing MCW optimization algorithms for travelers, such as people with disabilities (e.g., wheelchair users and people who are blind or visually impaired), people with special physical conditions (e.g., people with joint problems), and people with health conditions (e.g., people who must be less exposed to air pollution or sun light).
- Investigating and developing a predictive MMT-MCW methodology that allows route request well in advance and can monitor the recommended route up to minutes before the route is taken and update the recommendation based on changes of environmental and individual factors.
- Investigating and developing MMT-MCW simulation platforms for different purposes and applications, such as those described above.

References

[1] J. G. Benner and H. A. Karimi, Geo-Crowdsourcing. In Advanced Location-Based Technologies and Services (Ed: H. A. Karimi). Taylor and Francis, 2013.

[2] B. B. Brown, et al., "Mixed land use and walkability: Variations in land use measures and relationships with BMI, overweight, and obesity," Health and Place, 15(4), 2009, pp. 1130-1141.

[3] R. Ewing, S. Handy, R. C. Brownson, O. Clemente, and E. Winston, "Identifying and Measuring Urban Design Qualities Related to Walkability," Journal of Physical Activity and Health, 3, 2006, pp. s223-s240.

[4] R. Ewing and S. Handy, "Measuring the unmeasurable: Urban design qualities related to walkability," Journal of Urban Design, 14(1), 2009, pp. 65-84.

[5] A. Forsyth and M. Southworth, "Cities Afoot—Pedestrians, Walkability and Urban Design," Journal of Urban Design, 13(1), 2008, pp. 1-3.

[6] H. A. Karimi and M. Socharoentum, "Route2Health: A Novel Routing Service to Assist in Increasing Physical Activity," In Proceedings of the 7th International Conference on health Informatics, Angers, France, 2014, pp. 43-51.

[7] A. C. King, et al., "Aging in neighborhoods differing in walkability and income: associations with physical activity and obesity in older adults," Social Science and Medicine, 73(10), 2011, pp. 1525-1533.

[8] E. Leslie, et al., "Residents' perceptions of walkability attributes in objectively different neighbourhoods: a pilot study," Health and Place, 11(3), 2005, pp. 227-236.

[9] J. D. Marshall, M. Brauer, and L. D. Frank, “Healthy neighborhoods: walkability and air pollution,” Environmental Health Perspectives, 117(11), 2009, pp. 1752-1759.

[10] K. R. Smith, et al., “Walkability and body mass index: density, design, and new diversity measures,” American Journal of Preventive Medicine, 35(3), 2008, pp. 237-244.

[11] J. Sun, M. Walters, N. Svensson, and D. Lloyd, “The influence of surface slope on human gait characteristics: a study of urban pedestrians walking on an inclined surface,” Ergonomics, 39(4), 1996, pp. 677-692.

[12] S. J. Tharrett and J. A. Peterson (Eds.), ACSM’s health/fitness facility standards and guidelines. Human Kinetics, 2012, pp. 141.

[13] U.S. Census Bureau: State and County QuickFacts. (2010). Retrieved April 10, 2015 from http://www.census.gov/2010census/popmap/.

[14] D. V. Dyck, B. Deforche, G. Cardon, and I. D. Bourdeaudhuij, “Neighbourhood walkability and its particular importance for adults with a preference for passive transport,” Health and Place, 15(2), 2009, pp. 496-504.

[15] A. W. Agrawal, M. Schlossberg, and K. Irvin, “How far, by which route and why? A spatial analysis of pedestrian preference,” Journal of Urban Design, 13(1), 2008, pp. 81-98.

[16] GTFS Data Exchange. Retrieved April 10, 2015 from http://www.gtfs-data-exchange.com/

An Approach for a Web-based Diagnosis Cockpit for the Field of Neurology

Mirco Josefiok, Oliver Norkus and Jürgen Sauer

Deparment of Computer Science
Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstr. 114-118, 26129 Oldenburg
Email: {mirco.josefiok, oliver.norkus, juergen.sauer}@uni-oldenburg.de

Abstract—**In this paper, we present an approach for a Web-based diagnosis cockpit for the field of neurology which can be deployed either in a Web-based environment or as an on-premise solution. We strongly emphasize self service capabilities by adding additional metadata. Often, diseases manifest themselves through varying combinations of symptoms, especially in the field of neurology. Supporting medical staff in the process of finding a correct diagnosis to a hypothesis in a timely manner is very desirable to improve a patients outcome.**

Keywords*–*Expert systems in neurology, diagnosis support systems, Web-based information system

I. Introduction

Medical diagnoses in general, and neurological diagnoses, in particular, manifest themselves in different and varying combinations of symptoms. Practitioners rely on their experience and knowledge to find a diagnosis based on a hypothesis. They have to deal with various different symptoms and need to decide whether to perform further diagnostics or to begin an appropriate treatment. Stroke is one of the most common diseases in Germany. About six billion Euros are spent each year for the treatment of the most frequent types[1]. Especially for stroke patients a rapid and early diagnosis significantly improves their prognosis and allows for taking appropriate measures. It is therefore very desirable to offer support to medical staff as much as possible.

Hospitals and resident doctors have to deal with limited resources and minimal information technology capacities. Acquisition and operation of large scale solutions is often not an option. Expert interviews also indicate a demand for a solution which is easily accessible, needs minimal maintenance and blends seamlessly into the the daily work of the medical staff. Recently, we performed several expert interviews with doctors, controllers and information technology (IT) managers from local hospitals [1]. On foundation of those interviews, we were able to create a theoretical concept with a corresponding business process for support systems, which we developed in close cooperation with the interviewed experts [2]. In addition, we propose an approach for Web-based analytical information systems with self-service of business users in mind. We evaluated the developed prototype in several scenarios with business users who had different levels of previous knowledge [3]. The overall feedback was very positive - therefore we adapted architecture, structure and approach for developing the diagnosis cockpit.

In this paper, we present the first approach for a Web-based diagnosis cockpit for the field of neurology. Our approach comprises an expert system with an underlying knowledge base, an interface to provide additional metadata and a HTML5 frontend. With the introduction of additional metadata, we strive to make our system more accessible.

This paper is organized as follows. Section II introduces the foundations of our work, namely expert systems in neurology and Web based decision support systems (DSS). Section III analyzes the problem statement and introduces the architectural requirements for a Web-based diagnosis cockpit. Section IV presents our approach and gives an overview of the developed prototype and its architecture. Finally, the paper concludes with Section V, in which we point out remaining research and development challenges.

II. Foundation

Expert systems are software solutions utilizing specialists knowledge in order to support the decision-making process [4]. Contrary to other knowledge-based systems, which expert systems belong to, a knowledge base is derived from human experts. However, expert systems seek to reduce the need for human experts in practice. This is one main advantage, because human experts are often very costly [5]. In the healthcare domain, arious attempts of applying expert systems exist and the possible benefits are widely recognized [6, 7, 8]. Research and development of medical expert systems started in the 70s and systems are still evolving. MYCIN [9] and CADUCEUS [10] represent early attempts of expert systems with an inference engine. They were superseded by systems able to deal with imprecise or incomplete information. At present, computer systems are used in various steps of a patients treatment, e.g. in detecting adverse drug events [11]. But diagnosis related expert systems are rarely used in practice [12, 13, 14].

A. Expert Systems in Neurology

In a recent survey, we searched PubMed, Mendeley and Google Scholar for the terms *neurology*, *expert systems*, *diagnosis support systems*, *decision support systems* and *healthcare* in various combinations. We identified more than 15,000 references. After further filtering, we were able to reduce this set by 14,920 references. Afterwards we identified and removed 21 duplicates, 41 unrelated references and 11 reviews and editorials. Finally, we were left with 59 full text references from which 43 turned out to be unrelated work and 11 were identified as reviews or editorials. Subsequent to this, only eight full text references, actually dealing with diagnosis support systems in neurology, still remained [15].

[1] see diagnosis code I60, I61, I63, I65 at www.gbe-bund.de - last visited 2015-01-14

TABLE I. RESULTS OF LITERATURE ANALYSIS

Name	Last update	Number of diseases	Target diseases	Knowledge base	Method	Provision	Reference
Bunke	1988	unknown	Neurological diseases	Specialized literature	Database, rule-based	unknown	[16]
Bickel	2006	400	Neurological diseases	unknown	Database, statistical	Desktop Application	[17]
Roses	2009	6	Neurological diseases	unknown	Rule-based	Web-based	[18]
Reimers	2009	unknown	Neurological diseases	Specialized literature	Database, rule-based	Desktop Application	[19]
Borgohain	2012	4	Neuromuscular disorders	Expert interviews	Rule-based	Desktop Application	[20]
Borgohain	2012	1	Cerebral Palys Diagnosis	Expert interviews	Rule-based	Desktop Application	[21]
Ghahazi	2014	1	Multiple Sclerosis	unknown	Fuzzy logic	Spreadsheet	[22]

Table I gives an overview of our findings. We discovered different approaches for expert systems in neurology, but we found no references which show or hint at a routine application of any expert system. This emphasizes our previous findings [2]. In all cases, no considerations regarding user interface, comprehensibility of suggestions, expandability and customizability of knowledge are published. For practitioners in healthcare it is crucial that the results of the experts are comprehensible.

B. Web-based Decision Support Systems

Usually DSS consist of three basic components: A data management component, a knowledge base and a user interface. In case of Web-based DSS, they can be probably deployed on individual servers. DSS are very close to database management systems (DBMS) and are often built on top of a DBMS. More recent approaches tend to integrate the DSS into a broader application [23]. Those integrated DSS allow for an improved workflow and they are likely to be used more frequently. Most DSS, and also integrated DSS, are regular desktop applications and therefore need to be installed on every computer, where access is required. Furthermore, they do not offer an expandable knowledge base. Rare Web-based DSS offer access through common Web browser and mobile devices, which will be particularly important in hospitals.

III. Requirements and Business Drivers

During the expert interviews, we were able to derive the following requirements for a diagnosis cockpit:

Performance: Performance comparable to common desktop solutions.

Modularity and expandability: It should be possible to subsequently add complementary functionality to the prototype.

Expandability of knowledge: It should be possible to modify and expand the underlying knowledge base.

Metadata: Meaningful metadata should be provided to increase efficiency and accessibility when performing analyses.

User empowerment: It should be possible for inexperienced users to perform complex analyses.

Web standards: The prototype should use Web standards and should conform to the RESTful architecture paradigm. Representational State Transfer (REST) is widely accepted as a simper alternative to service approaches [24].

A recent study [7] lists additional success factors for clinical DSS, which include seamless integration in the workflow of clinicians, integration of the DSS into a broader system, provision of recommendations rather then decisions, minimal data input, easy access and clear presentation of the results.

In Section IV we show how we approach the requirements and in Section V we will point out what we intend to do to fulfill them.

IV. Approach for a Diagnosis Cockpit

In this section, we present our approach for a Web-based diagnosis cockpit. In Section IV-A we describe the server side of our application and especially how we utilize various technologies for our purpose. In Section IV-B, we explain the implementation of our client, with particular emphasis on self-service and the usage of metadata.

The framework conditions for our approach are determined by the theoretical concept and the corresponding business process. Both were developed in close cooperation with experts from the healthcare domain. Figure 1 depicts an excerpt of the business process. It shows how case data, meta data and patient data are prepared and provided.

A. Server side Prototype

For developing the server side part of our prototype, we chose Spring Boot, which offers rapid application development capabilities whilst being well extensible. This complies with our requirement *Modularity and expandability*. Our prototype does not require a dedicated application server but only a Java Virtual Machine. Moreover, it relies on a relational database for storing data. In future iterations we'd like to add a NoSQL database for storing / caching unstructured data. Figure 2 shows the structure of the application. We implemented a structure conforming to the Representational State Transfer (REST) software architecture style. RESTful Web services for data exchange between client and server are implemented. These standards were chosen to comply with the business driver *Web standards*. Moreover, we added the capability to work securely in a multi-client environment.

Concering the *Recommender* component, we have not yet finally decided which approach to follow. We are evaluating whether to use FuzzyLogic or to rely on crisp rules and emulate a fuzzy approach. We created small knowledge based on the expert interviews and implemented the knowledge base with Fuzzy Control Language (FCL) and JBoss Drools Domain Specific Language. Drools [25] offers a comprehensive environment for developing expert systems with a crisp rule set. There are approaches using fuzzy logic with Drools, but they are not mature, yet [26]. FCL was standardized by IEC 61131-7 and different Java implementations exits, eg. jFuzzyLogic [27].

At the moment we carry out experiments to decide which approach complies most with our requirements (see section III), especially *Performance*. Listing 1 shows an FCL example for Neuritis Vestibularis. There are four possible symptoms as inputs and the diagnosis as an output. Altogether,

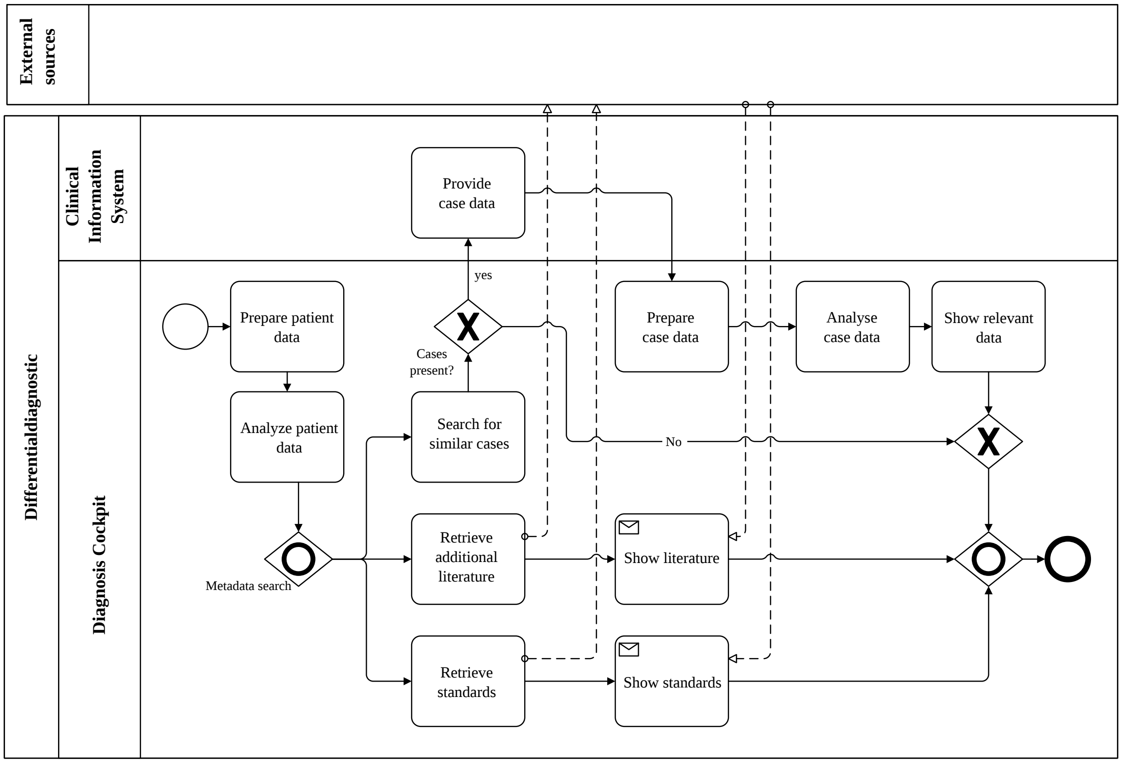


Figure 1. Preparation and Provision of Case Data, Metadata and Patient Data

we deducted seven rules from expert interviews, but only the one which produces *sure* is depicted as an example.

Additional, meaningful metadata is provided through the integration of third-party services. Our prototype fetches domain specific knowledge, regarding diagnoses, symptoms and clinical pathways, from the ”Deutsche Gesellschaft für Neurologie” [28]. We search PubMed for supplementary literature to the recommendations produced by our *Recommender*. PubMed is a search engine access provider to the MedLine database of medical publications. In future iterations we plan to integrate WikiData to access correct medical classification codes. Through the *ServiceFactory* we offer a unified access point for possible clients as a RESTful interface. Data is serialized into JavaScript Object Notation (JSON), before sending it to the client.

```
FUNCTION_BLOCK Neuritis_vestibularis
VAR_INPUT
        rotary_vertigo : REAL;
        tendency_to_fall : REAL;
        spontaneous_nystagmus : REAL;
        unterberger_test : REAL;
END_VAR
VAR_OUTPUT
        Neuritis_vestibularis : REAL;
END_VAR
FUZZIFY rotary_vertigo
        TERM sudden := (0, 1) (2, 1) (4, 0) ;
        TERM attack := (2, 0) (4, 1) (6, 1) (8, 0);
        TERM constant := (6, 0) (8, 1) (10, 1);
END_FUZZIFY
FUZZIFY tendency_to_fall
        TERM left := 1;
        TERM right := 2;
END_FUZZIFY
FUZZIFY spontaneous_nystagmus
        TERM left := 1;
        TERM right := 2;
END_FUZZIFY
FUZZIFY unterberger_test
        TERM left := 1;
        TERM right := 2;
END_FUZZIFY
DEFUZZIFY Neuritis_vestibularis
        TERM possible := (0, 1) (4, 0) ;
        TERM probable := (1, 0) (4,1) (6,1) (9,0);
        TERM sure := (6, 0) (9, 1);
        METHOD : COG;
        DEFAULT := 0;
END_DEFUZZIFY
RULEBLOCK No1
        AND : MIN;
        ACT : MIN;
        ACCU : MAX;
        RULE 1 : IF rotary_vertigo IS  sudden AND tendency_to_fall IS left AND
                spontaneous_nystagmus IS right AND unterberger_test IS left THEN
                Neuritis_vestibularis IS sure;
        ...
END_RULEBLOCK
END_FUNCTION_BLOCK
```

Listing 1. FCL Example for Neuritis Vestibularis

B. Client side Prototype

Our prototypical client consumes the Web services of the described server side prototype. It was developed using HTML5/JavaScript, especially AngularJS [29]. In order to contribute to the business driver *Modularity and expandability*, the application is organized in a modular way with reusable components representing different aspects of the view.

Figure 3 shows a screenshot of the main view. ① shows a selected patient. Basic information about the patient is provided (e.g. name, address etc.). Possible other fields to display would be age, sex etc. Additional information about medication, planned treatments etc can be seen at ②. ③ shows a fever curve and other corresponding information with customizeable charts is displayed. The process of filtering and selecting a patient is accessible through a dedicated modal. ④ shows the selected symptoms with a short description. Possible diagnoses with additional information can be seen at ⑤. Again the process of selecting symptoms is accessible through a modal. Additional information in form of literature references is provided at ⑥. We constantly work on the user interface to

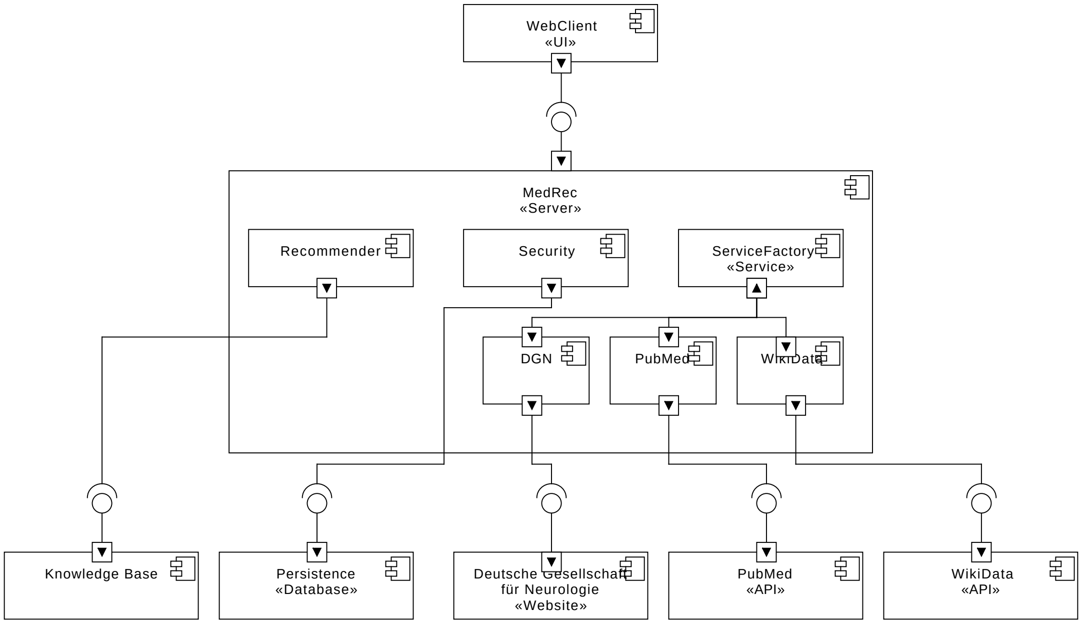


Figure 2. Server Structure

comply with our requirements and make the application easily accessible and integrate our approach into the daily routines of practitioners.

V. Conclusion and further Challenges

We presented an approach for a Web-based diagnosis cockpit. According to early feedback from doctors and medical staff our approach is very promising and a significant improvement over existing solutions. The dynamic addition of metadata, the easy access, the traceability and comprehensibility of results are most important for medical staff. But there was some criticism we need to work on in further iterations, mostly regarding our operation concept. Whilst we developed only a research prototype, a possible application accessible by end users, must take security issues into account. In addition we will rework parts of our operation concept, especially regarding the processes of selecting patients and symptoms. Feedback from IT managers was very positive as well. They considered the low hurdle to usage and the minimal effort needed to maintain our system very promising. At the moment, we mostly work with sample data and an incomplete and limited rule set. We are currently conducting a series of expert interviews with doctors from a local hospital to build a larger knowledge base. We aim to provide a reasonable set of rules with which comprehensible results can be produced and will continuously evaluate our work with practitioners from local healthcare facilities. With a larger knowledge base we will have the opportunity to perform more complete tests and finally chose an approach for our knowledge base. Another planned improvement concerns the modification of the underlying knowledge base by users at runtime. At the moment it is possible to modify the knowledge base with an external tool and afterwards the server side part needs to be recompiled. A future version could incorporate the needed user interface elements which should allow users to modify the knowledge base.

References

[1] M. Josefiok, “Measuring and Monitoring the Rehabilitation of Patients on Monitoring Stations via the Analyses of poly–structured Data,” in *Informatik 2014, 44. Jahrestagung der Gesellschaft für Informatik e.V. (GI),*, ser. LNI. GI, 2014.

[2] J.-P. Weiß, M. Josefiok, T. Krahn, and H.-J. Appelrath, “Entwicklung eines Fachkonzepts für die klinische Entscheidungsunterstützung durch Analytische Informationssysteme,” in *Proceedings of the 12th International Conference on Wirtschaftsinformatik (WI2015)*, 2015.

[3] *An Approach for a Web-based Analysis Solution with MUSTANG*, ser. The Seventh International Conferences on Advanced Service Computing. IARIA XPS Press, 2015.

[4] F. Puppe, *Einführung in Expertensysteme (Studienreihe Informatik) (German Edition)*. Springer, 1988.

[5] C. Beierle and G. Kern-Isberner, *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen (Computational Intelligence) (German Edition)*, 5th ed. Springer Vieweg, 2014.

[6] E. Berner, “Clinical decision support systems: state of the art,” *AHRQ Publication*, no. 09, 2009.

[7] D. Dinevski, U. Bele, T. Šarenac, U. Rajkovič, and O. Šušteršič, “Clinical decision support systems,” *Studies in health technology and informatics*, pp. 217–238, Dec. 2013.

[8] N. K. J. Adhikari, J. Beyene, J. Sam, and R. B. Haynes, “Effects of Computerized Clinical Decision Support Systems on Practitioner Performance,” vol. 293, no. 10, pp. 1223–1238, 2014.

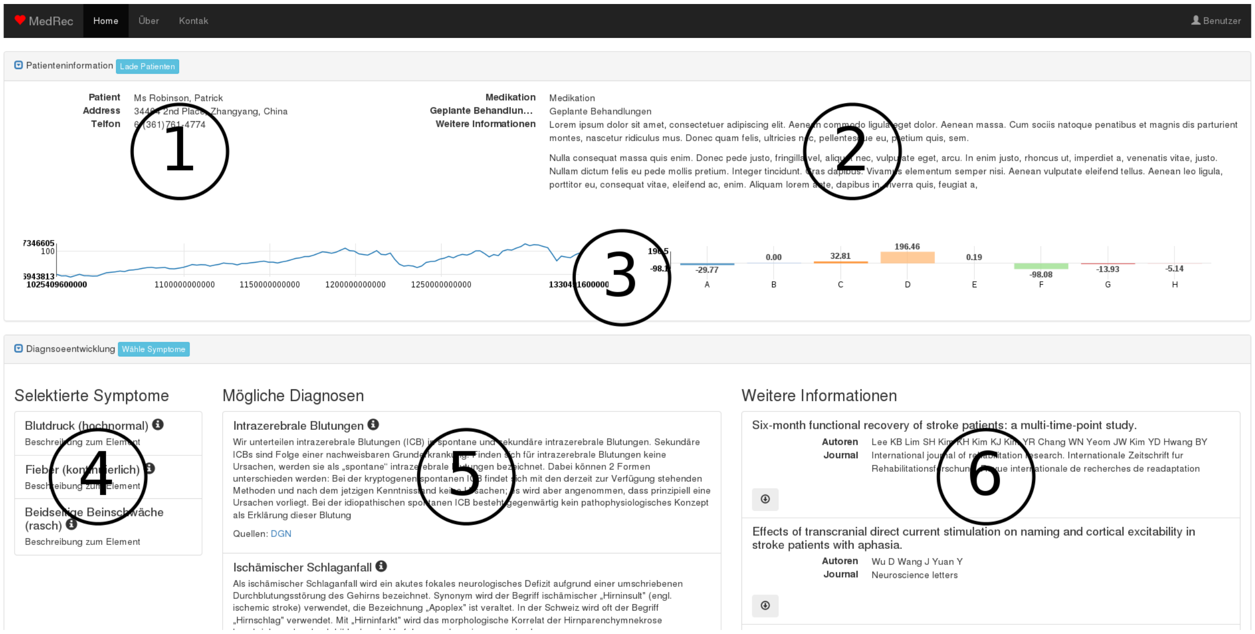
[9] P. Van Remoortere, “Computer-based medical consultations: MYCIN,” pp. 385–386, 1979.

[10] G. Banks, “Artificial intelligence in medical diagnosis: the INTERNIST/-CADUCEUS approach.” *Critical reviews in medical informatics*, vol. 1, pp. 23–54, 1986.

[11] T. Krahn, M. Eichelberg, S. Gudenkauf, E. Laleci, B. Gokce, and A. Hans-Jürgen, “Adverse Drug Event Notification System – Reusing clinical patient data for semi–automatic ADE detection,” in *Proceedings 27th International Symposium on Computer–Based Medical Systems (IEEE CBMS 2014)*, 2014.

[12] W. Ishak and F. Siraj, “Artificial intelligence in medical application: An exploration,” *Health Informatics Europe Journal*, 2002.

[13] C. Angeli, “Diagnostic Expert Systems: From Expert’s Knowledge to Real-Time Systems,” in *Advanced Knowledge Based Systems: Model, Applications & Research*, 2010, vol. 1, ch. Chapter 4, pp. 50–73.



Patient Ms Robinson, Patrick
Address 34404 2nd Place, Zhangyang, China
Telfon 61(361)761-4774

Medikation Medikation
Geplante Behandlun... Geplante Behandlungen
Weitere Informationen Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a,

Selektierte Symptome

Blutdruck (hochnormal)
Beschreibung zum Element

Fieber (kontinuierlich)
Beschreibung zum Element

Beidseitige Beinschwäche (rasch)
Beschreibung zum Element

Mögliche Diagnosen

Intrazerebrale Blutungen
Wir unterteilen intrazerebrale Blutungen (ICB) in spontane und sekundäre intrazerebrale Blutungen. Sekundäre ICBs sind Folge einer nachweisbaren Grunderkrankung. Finden sich für intrazerebrale Blutungen keine Ursachen, werden sie als „spontane“ intrazerebrale Blutungen bezeichnet. Dabei können 2 Formen unterschieden werden: Bei der kryptogenen spontanen ICB findet sich mit den derzeit zur Verfügung stehenden Methoden und nach dem jetzigen Kenntnisstand keine Ursachen; es wird aber angenommen, dass prinzipiell eine Ursachen vorliegt. Bei der idiopathischen spontanen ICB besteht gegenwärtig kein pathophysiologisches Konzept als Erklärung dieser Blutung

Quellen: DGN

Ischämischer Schlaganfall
Als ischämischer Schlaganfall wird ein akutes fokales neurologisches Defizit aufgrund einer umschriebenen Durchblutungsstörung des Gehirns bezeichnet. Synonym wird der Begriff ischämischer „Hirninsult" (engl. ischemic stroke) verwendet, die Bezeichnung „Apoplex" ist veraltet. In der Schweiz wird oft der Begriff „Hirnschlag" verwendet. Mit „Hirninfarkt" wird das morphologische Korrelat der Hirnparenchymnekrose

Weitere Informationen

Six-month functional recovery of stroke patients: a multi-time-point study.
Autoren Lee KB Lim SH Kim KH Kim KJ Kim YR Chang WN Yeom JW Kim YD Hwang BY
Journal International journal of rehabilitation research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation

Effects of transcranial direct current stimulation on naming and cortical excitability in stroke patients with aphasia.
Autoren Wu D Wang J Yuan Y
Journal Neuroscience letters

Figure 3. Client Prototype

[14] T. Sharma and S. Jain, “Survey on Expert System,” vol. 2, no. 1, pp. 343–346, 2012.

[15] M. Josefiok, T. Krahn, and J. Sauer, “A Survey on Expert Systems for Diagnosis Support in the Field of Neurology,” in *Intelligent Decision Technologies - Proceedings of the 4th International Conference on Intelligent Decision Technologies (IDT’2015)*, R. J. Howlett and L. C. Jain, Eds. Springer, 2015.

[16] H. Bunke, F. Flückiger, H. P. Ludin, and F. Conti, “Diagnostic Expert Systems in Neurology - A Comparison between a Rule Based Prototype and a Database Oriented Approach,” in *Expert Systems and Decision Support in Medicine*. Springer, 1988, pp. 209–212.

[17] A. Bickel and M. Grunewald, “Ein Expertensystem für das Fachgebiet Neurologie-Möglichkeiten und Grenzen,” *Fortschritte der Neurologie· . . .*, pp. 723–731, 2006.

[18] S. Y. M. Vaghefi and T. M. Isfahani, “ROSES: AN EXPERT SYSTEM FOR DIAGNOSING SIX NEUROLOGIC DISEASES IN CHILDREN Title,” in *Int. Conf. Intelligent Systems & Agents*, 2009, pp. 259–260.

[19] C. D. Reimers and A. Bitsch, “Differenzialdiagnose Neurologie,” *Kohlhammer: Stuttgart*, 2008.

[20] R. Borgohain and S. Sanyal, “Rule Based Expert System for Diagnosis of Neuromuscular Disorders,” *arXiv preprint arXiv:1207.2104*, pp. 1–5, 2012. [Online]. Available: http://arxiv.org/abs/1207.2104

[21] ——, “Rule Based Expert System for Cerebral Palsy Diagnosis,” *arXiv preprint arXiv:1207.0117*, pp. 1–4, 2012. [Online]. Available: http://arxiv.org/abs/1207.0117

[22] M. A. Ghahazi, M. H. Fazel Zarandi, M. H. Harirchian, and S. R. Damirchi-Darasi, “Fuzzy rule based expert system for diagnosis of multiple sclerosis,” in *Norbert Wiener in the 21st Century (21CW), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–5.

[23] R. Elliott, “Strathprints Institutional Repository,” *Psychotherapy Research*, vol. 22, pp. 1753–1759, 2008.

[24] Alex Rodriguez, “RESTful Web services: The basics,” http://www.ibm.com/developerworks/library/ws-restful/, 2015, last visited 2015-01-22.

[25] Red Hat Inc, “Drools - Drools - Business Rules Management System (Java^{TM}, Open Source,” http://www.drools.org, 2015, last visited 2015-01-22.

[26] N. Wulff and D. Sottara, “Fuzzy reasoning with a Rete-OO rule engine,” in *Rule Interchange and Applications*. Springer, 2009, pp. 337–344.

[27] P. Cingolani and J. Alcala-Fdez, “jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation.” in *FUZZ-IEEE*. Citeseer, 2012, pp. 1–8.

[28] Deutsche Gesellschaft für Neurologie e.V., “Deutsche Gesellschaft für Neurologie e.V.” http://www.dgn.org, 2015, last visited 2015-01-22.

[29] AngularJS Team, “AngularJS - HTML enhanced for web apps!” https://angularjs.org/, 2015, last visited 2015-01-22.

ZLOC: Detection of Zombie Users in Online Social Networks Using Location Information

Jing Deng, Lixin Fu, Yi Yang

Department of Computer Science
University of North Carolina at Greensboro
Greensboro, NC 27410, U.S.A.
Email: jing.deng@uncg.edu, lfu@uncg.edu, y_yang6@uncg.edu

***Abstract*—Online social networks serve as a promising platform for social eliteness and financial gain. With such a promise or dream, zombie accounts, behind which stand no real users, become prevalent. The detection of such accounts has been games of cat and mouse, with more and more sophisticated methods used by zombie account managers. In this work, we propose a new zombie account detection technique called ZLOC, Zombie detection using Location information. ZLOC uses the location information of an account's friends or followers. More specifically, we investigate the follower accounts of suspected zombie accounts in SINA WeiBo, one of the two most popular microblogging websites in China. Our scheme is based on a natural social behavior that many of one's friends are usually in the vicinity of his/her location. Our analysis shows that the proposed ZLOC scheme has some salient features that help with zombie detection.**

***Keywords*–Online Social Networks; Zombie Detection; Fake Users; Detection Accuracy.**

I. Introduction

With the prevalent usage of Internet, online social networks have quickly become the center of human interactions. Younger generations, as well as the old ones, use different kinds of online tools to communicate and to obtain information. Web pages supported by HTML 5.0 can be very dynamic and have various features, but Facebook, Twitter [1][2], LinkedIn, to name a few, are the place to make friends, share information or news, or establish professional connections.

For any online website with a large number of users, there also come the zombie accounts, or the faked users. Different from legitimate users who have real people behind to communicate, a software or a zombie manager may be controlling these accounts. The reasons for the existence of such zombie accounts can vary. They range from financial exploitation to phantom fame purposes. For instance, in most of daily deal websites, each deal is usually ranked by the number of thumbs-up and thumbs-down. With the control of a large number of zombie accounts, it is then possible to control the hotness of a deal by directing some of the zombie accounts to thumb up a particular deal (by a sponsored company or website) and/or thumb down another deal (maybe by a different company or website). Some online advertisement companies may exploit the effectiveness on the popularity of certain accounts to achieve financial gains. Even worse, the spammers may even steal information from profiles on network or direct users to phishing websites.

It is thus essential to detect such zombie accounts from online social networks. Unfortunately, the task can be hindered by the adaptive behavior of the zombie managers (the people controlling the zombie accounts). It is further complicated by the sometimes less than normal active behavior pattern by the real user accounts: the difference of an abandoned account and an occasionally used zombie account can just be tiny. Nevertheless, zombie detection has seen some interesting progresses in recent years, with techniques ranging from simply checking number of friends, to number of active posts or activity, to more sophisticated statistical analysis among different herds of users [3][4]. Interestingly, none of these prior arts has investigated the use of location information of the user accounts.

In this work, we focus on the detection of zombie accounts in SINA WeiBo, one of the two most popular twitter-like microblogging websites in China. Similar to Twitter, WeiBo allows each user to follow a number of other users. Since any real person is unlikely to be able to read large amount of feeds, it posts a limit of 2000 follows for each user. The number of followings, however, can be as large as millions, depending obviously on the popularity of the accounts.

Our work is based on the following natural social bond observation. Most people interact with their friends or relatives who live within the same city or the same province (state) where they live. While anyone may have friends or followers from other cities or provinces (states), the ratio of followers who are in the same city or province (state) should be at least higher than a certain threshold. Instead, zombie accounts tend to have followers from a very diversed geographical locations. Our proposed scheme ZLOC, Zombie detection based on Location information, is based on such an salient observation.

Our paper is organized as follows. In Section II, we discuss prior arts and their differences with our proposed scheme. We illustrate and explain our ZLOC scheme in Section III, followed by the performance evaluation in Section IV. In Section V, we conclude our work and point out several directions for future work.

II. Related Work

There have been some work in fake followers for online social networks in the technical literature. We review them in the following:

Rumors can propagate easily in social networks. Sun et al. [5] proposed an effective rumor classifier that categorized rumors into four types, one of which was text-picture unmatched event rumors. They designed special features that may be used to build a classifier to differentiate rumors from ordinary posts. Thomas et al. [1] researched on suspended Twitter accounts to find their lifetime events and behavior. McCord and Chuah [6] presented a detection method with user-based and content-based features and applied traditional classifiers such as Random Forest, Naive Bayesian, Support Vector Machine, and K-nearest neighbors.

In social networks, there are spammers, the detection of which is essential and can impact real users [7]. Hu et al. gave a unified model for detecting social spammers in microblogging by integrating both social network information and content information [8]. Lee et al. [9] deployed social honeyspots to harvest suspicious spam profiles and then classified them using machine learning. Lin et al. [10] collected a set of spammer samples by proactive honeypots and keyword based searching, and designed an online system for identifying spammers. They found three abnormal behaviors of the spammers: aggressive advertising, repeated reposting, and aggerssive following. Chu et al. mainly focussed on the detection of large-scale spam campaigns on Twitter rather than screening individual tweets [11]. They clustered the collected dataset of 20 million tweets into different campaigns according to their same final URLs. They presented a classification system based on a set of features generated from campaign data.

The detection of non-real users in social networks has shown to be tricky because of their evasive and ever-changing behaviors. Shen et al. [12] proposed a binary classifier to detect fake followers by their extracted major features in SINA WeiBo and presented their classifier's performance. Guo et al.[13] collected more than 20 million profiles of users and researched their posting behaviors. Marionette users like puppets are fabricated for fake popularity or financial gain. Wu et al. integrated both individual user tweeting behavior information and the social interactions among users to develop a semi-supervised probabilistic model in order to distinguish marionettes from normal users [14]. Due to lack of user-generated contents, it is difficult to capture the profiles of lurking users. Zhang et al. [15] presented a unified social context graph model and an algorithm to generate profiles of the lurking users to effectively detect them. Wang and Lu [3] introduced a star sampling method by taking all the neighbors as valid samples. They used it to identify ten thousands of top bloggers on Weibo. To analyze Twitter sphere, Black et al. [2] proposed an elegant architecture to perform Twitter studies. Jiang et al. [4] proposed CATCHSYNC that used and measured two suspicious behaviors: the first measure is "sync" behavior of zombies, that is, they often have similar behavior; another is "norm", that is, their behavior is different from other normal users.

Armed with a very large dataset with 54 million users and 1.8 billion tweets and a manually labeled collection of 1,065 users, Benevenuto et al. [16] carefully examined a large set of features, such as fraction of tweets with URLs, hashtags, and spam words, number of replies, number of followees and followers, account life time, number of tweets received, etc., to differentiate spammers from normal users. They also used Support Vector Machine and Chi-square method to classify and characterize the spammers. Liu et al. proposed ProZombie [17], a two-stage cascading model for detecting zombies. They also came up with new features to give a more refined description of Weibo users, improving the modeling efficiency without loss of accuracy. Zombies are essentially the same as Socialbots or Sybil accounts, which have received attention from the perspective of Turing machine/human classification [18][19][20][21].

TABLE I. Variables.

SAMEP	Ratio of followers who share the same province (state) with the user
SAMEC	Ratio of followers who share the same province (state) and city with the user
FER	Number of followers of a user
FING	Number of users that a user is following

To conclude, while quite some work has been performed on zombie detection in online social networks, the detection of such still remains inaccurate and/or requires too much extra information. None of the above prior arts considered user registration location specifically for zombie detection, although some used similar features in the big picture of user classification [11][17]. Instead, our work focuses on the use of such location information make accurate detection decisions.

III. The ZLOC Scheme

We describe the ZLOC scheme in this section. The ZLOC scheme requires the follower information of a suspected account. Once the list of all followers is obtained, the registration information of each of the followers will be retrieved through a simple web access. Such registration location information is then compared with the registration location information (such as Guangzhou, Guangdong). Then the following two numbers are generated: the number of followers of the suspected account having the same city and province (state) information as the suspected account, denoted as SAMEC; and the number of followers of the suspected account having the same province (state) information as the suspected account, denoted as SAMEP. Note that it is obviously true that SAMEC $<$ SAMEP. We present all variables on Table I, as add "_TH" to denote the threshold in a comparison, e.g., FER_TH is the threshold for FER.

These two numbers are then compared to two thresholds, SAMEC_TH and SAMEP_TH, respectively. When a suspected account satisfies SAMEC $<$ SAMEC_TH and SAMEP $<$ SAMEP_TH at the same time, it is considered a zombie based on location information.

Due to the fact that some users may register under locations different from their real residential location, other perspectives should be checked to improve the detection accuracy. A simple and low-cost perspective is the number of followers, defined as FER, and the number of followings, defined as FING. At least in the early days, zombie accounts made large number of follows so that they might get followed back. Therefore, they usually have close-to limit FING numbers. On the contrary, it is hard to find real user accounts to follow them back, except that the zombie accounts can be directed to follow themselves (orchestrated by one zombie manager or even several zombie managers). It is thus easy to see that any real user account should have FING lower than a threshold FING_TH unless his/her FER number is greater than a threshold FER_TH (such as some highly popular accounts). Hence, the following additional detection rule:

When FER $<$ FER_TH and FING $>$ FING_TH, a suspected account is considered a zombie.

The reason to include a threshold for the number of followers is that some accounts may have so many followers and the users may prefer to follow (or counter-follow) a certain percentage of these followers. Therefore, it is possible that these real user accounts may have larger numbers of followings.

In fact, we use the above basic detection rule, termed FER-FING, to demonstrate the effectiveness of our location-based zombie detection strategy in Section IV. The basic FER-FING scheme is

$$\begin{aligned}
&C1: \text{FER} < \text{FER\_TH} \\
&C2: \text{FING} > \text{FING\_TH} \\
&\text{Rule: } C1 \cdot C2 \text{==TRUE} \rightarrow \text{zombie} \qquad (1)
\end{aligned}$$

We describe the ZLOC scheme in the following:

$$\begin{aligned}
&C1: \text{FER} < \text{FER\_TH} \\
&C2: \text{FING} > \text{FING\_TH} \\
&C3: \text{SAMEP} < \text{SAMEP\_TH} \\
&C4: \text{SAMEC} < \text{SAMEC\_TH} \\
&\text{Rule: } C1 \cdot (C2 + C3 \cdot C4) \text{==TRUE} \rightarrow \text{zombie} \quad (2)
\end{aligned}$$

IV. Performance Evaluation

In this section, we present the performance evaluation of our ZLOC scheme.

A. Data Collection and Performance Metrics

First, we explain how we obtained our dataset. The data was collected through a web crawler that started with a real user account and found the followers and the followers' followers, and so on. For each user, we retrieved the list of all followers that are available from the WeiBo webpage. Our crawler stopped when the number of accounts reached 10,000. For all of these 10,000 accounts and all their known followers, we retrieved their registration location information (together with other information that we did not use, such as registration time/date and last post time/content).

Note that, in SINA WeiBo, only about 200 followers are now disclosed to anyone other than the user himself/herself. This might have been posted for privacy reason. It has an interesting effect on our evaluation. First of all, such a limit means that the follower list that ZLOC processes and makes decision upon is incomplete. Thus, the accuracy of zombie detection can be questionable. However, we argue that such a large snapshot of the follower list is already quite revealing, as demonstrated by our results. Secondly, it also affected the list of users that we crawled in a way that might have changed the ratio of zombie/real accounts in the dataset. Since the goal here is to evaluate how accurate the ZLOC scheme is, the ratio of the dataset does not actually matter.

The 10,000 accounts were then passed through the ZLOC scheme and the basic FER-FING scheme. A decision of either zombie or real account would be reached at the end. We randomly sampled more than 100 accounts and checked (through human reading) whether they were really zombies or real user accounts. We evaluated these two schemes in the following performance metrics (all based on the sampled pool):

Successful Detection Ratio: This is the ratio of the number of zombie accounts that are detected as zombie users divided by the number of all zombie accounts.

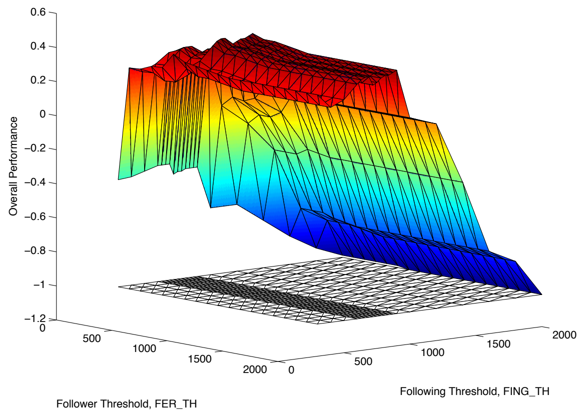


Figure 1. Overall performance of the basic FER-FING scheme with different FER_TH and FING_TH values.

Missed Detection Ratio: This is the ratio of the number of zombie accounts that are detected as real user accounts divided by the number of all zombie accounts.

False Alarm Ratio: This is the ratio of the number of real user accounts that are detected as zombie accounts divided by the number of all real accounts.

Overall Performance: This is computed as Successful Detection Ratio minus Missed Detection Ratio as well as False Alarm Ratio.

Among these metrics, the last calls for some explanations. In any classical detection problem, it is rather easy to increase successful detection ratio while ignoring missed detection and false alarm ratios, or vice versa. A practical scheme should indeed balance all three. In fact, different weights (positive or negative) for such ratios can be applied to these ratios and one can try to maximize the combined return. In this work, we choose the simple subtraction as the final return and leave more complex return weight to our future work.

B. FER_TH and FING_TH Selection

First of all, we need to choose the best FER_TH and FING_TH for the basic FER-FING scheme as well as our ZLOC scheme.

We present our investigation of the basic FER-FING scheme for its best FER_TH and FING_TH values (see Figure 1). When these thresholds are too low or too high, the overall performance of the FER-FING scheme is rather low. When they are in the range of 700-900, the basic FER-FING scheme works the best, at least for the data points that we sampled. Therefore, we will choose FER_TH and FING_TH values as 700 and 900, respectively. All simulations in Section IV-C are based on these values. Note that other similar parameters would produce similar comparisons, as shown below.

C. Accuracy of ZLOC

First of all, we plotted the SAMEP and SAMEC values of all sampled accounts (hence, we have manually checked whether they are zombie accounts or real user accounts) in Figure 2. All zombie accounts are represented by a red circle on its (SAMEP, SAMEC) position. Note that we have added a

ISBN: 978-1-61208-407-7

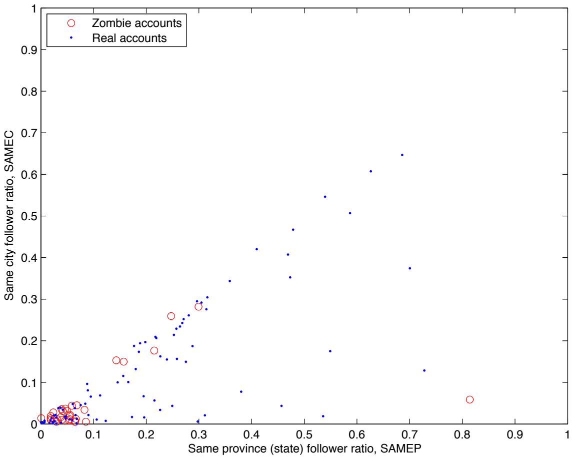


Figure 2. Distribution of sampled accounts. Here we plot sampled accounts on a 2-D surface using their SAMEP and SAMEC values. Zombie accounts and real user accounts are distinguished by the different symbols.

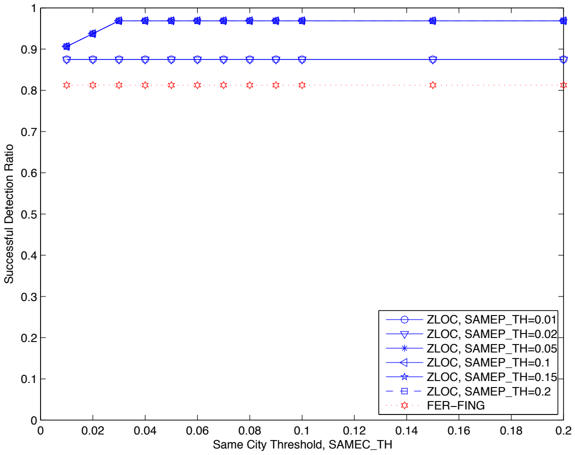


Figure 3. Successful detection ratio for the FER-FING scheme and ZLOC scheme.

slight perturbation for each point in order to show those at the exact same locations, which were caused by the small numbers of followers and hence, the same SAMEP values and the same SAMEC values.

From Figure 2, the pattern of zombie users is clearly demonstrated as most of them stay on the lower left corner of the region, except for a few data points. On the other hand, real user accounts are more diverse and vastly spread. Such an observation has served as the inspiration for our work. Also note that all data points satisfy SAMEC < SAMEP.

We present the successful detection ratio in Figure 3. It can be seen that the ZLOC scheme generally has better successful detection ratios than the basic FER-FING scheme. Within the ZLOC scheme, a very small SAMEP_TH is rather ineffective. When SAMEP_TH reaches 0.1 and 0.2, however, the successful detection ratio remains the same. It might have been caused by the fact that none of the users have more than 10% of their followers within the same province (state). Except for SAMEP_TH=0.01, the successful detection ratio shows slight increases as SAMEC_TH increases, as more and more accounts become eligible to be declared as zombies.

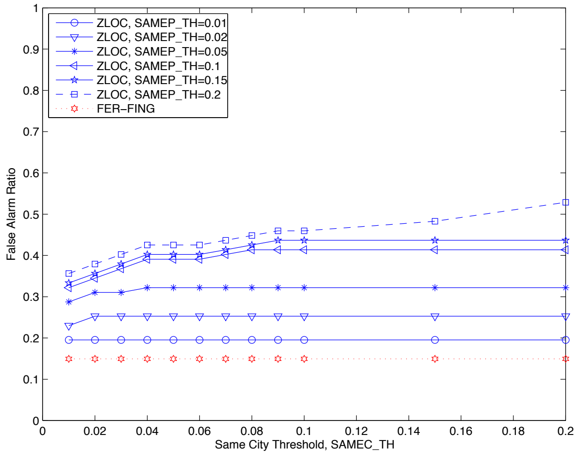


Figure 4. False alarm ratio for the FER-FING scheme and ZLOC scheme.

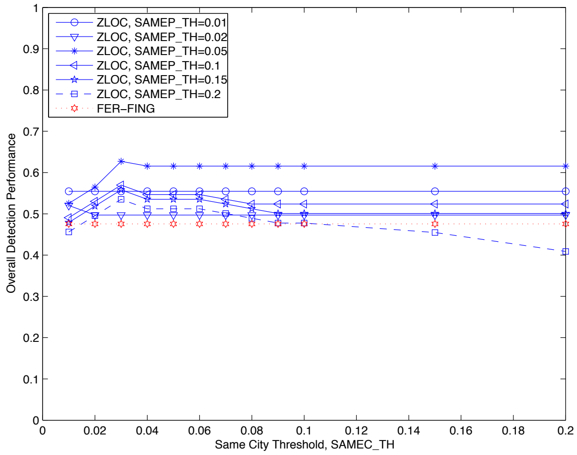


Figure 5. Overall performance comparison of the FER-FING scheme and ZLOC scheme.

The performance in false alarm ratio is shown in Figure 4. As SAMEP_TH increases, the false alarm ratio increases as more and more accounts fall within the detection threshold. Similarly, the increase of SAMEC_TH also raises false alarm ratio. In general, the basic FER-FING scheme has a lower false alarm ratio than the ZLOC scheme.

We compare the overall performance of the ZLOC scheme and the basic FER-FING scheme in Figure 5. The ZLOC scheme outperforms the basic FER-FING scheme except in a few places where the SAMEC_TH value is set to be too large. For several of the SAMEP_TH lines, an interesting up and down trend can be observed as SAMEC_TH increases, suggesting an optimum choice for SAMEC_TH. This is because of the joint impact of successful detection, false alarm, as well as missed detection. Overall, the results in Figure 5 suggest the best SAMEP_TH and SAMEC_TH to be 0.05 and 0.03, which are the parameters that we use in Section IV-D.

D. Zombie Ratios

Lastly, we present the ratio of accounts in our dataset that these two schemes detect as zombies. Among these 10,000 accounts, ZLOC detected 5,300 of them as zombies and the basic FER-FING scheme found 3,300 as zombies. The difference is significant, underlining the impact of our location-based detection approach.

V. Conclusions and Future Work

We have presented a new zombie detection scheme called ZLOC. The ZLOC scheme takes advantage of the fact that the location vicinity between a follower and the person whom he/she is following. ZLOC then compares the ratio of such followers of any suspected account. If the ratio is below a certain threshold, the account is more likely to be a zombie. Through our simulation studies, we have found that ZLOC could use two thresholds, one to compare with the ratio of the followers within the same city and the other to compare with the ratio of the followers within the same province (or state). With such additional ratios, the ZLOC scheme has been demonstrated to raise the successful detection of zombie accounts significantly. In addition, the overall performance, defined by successful detection minus false alarm ratio, as well as missed detection ratio, is also higher than other schemes.

In terms of applications, we believe that ZLOC can be used in combination with other techniques, such as those in [12], to further improve the accuracy of zombie user detections. Our scheme does not rely on detailed information of the suspicious user such as posting habits, timing, contents, etc., making it a great candidate for efficient detection.

We note that it is possible to adapt our ZLOC approach to detect phantom users in other online social networks such as Facebook, LinkedIn, and/or Twitter. For instance, in Twitter, many accounts show their current residential locations. Their followers can be checked as well and our ZLOC approach can be applied for detection. The story would be slightly different in Facebook, mainly because all connections are bi-directional instead of the directional following in microblogging websites. Instead of checking followers' location information, perhaps the location statistics of all friends of one account can be checked, although such friends are usually not viewable from a third party unless the privacy setting allows so.

In our future work, we will investigate the use of actual vicinity instead of hard-coded same city ratio. Therefore, followers in the neighboring cities will still be considered as close-by. The investigation of larger datasets from other social networks will be helpful as well.

Acknowledgment

J. Deng would like to thank the financial support from NSF, grant No. CCF-1320428.

References

[1] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: An analysis of twitter spam," in Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 243–258.

[2] A. Black, C. Mascaro, M. Gallagher, and S. P. Goggins, "Twitter zombie: Architecture for capturing, socially transforming and analyzing the twittersphere," in Proceedings of the 17th ACM International Conference on Supporting Group Work, ser. GROUP '12. New York, NY, USA: ACM, 2012, pp. 229–238.

[3] H. Wang and J. Lu, "Detect inflated follower numbers in osn using star sampling," in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ser. ASONAM '13. New York, NY, USA: ACM, 2013, pp. 127–133.

[4] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "Detecting suspicious following behavior in multimillion-node social networks," in Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion, ser. WWW Companion '14. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2014, pp. 305–306.

[5] S. Sun, H. Liu, J. He, and X. Du, "Detecting event rumors on sina weibo automatically," in Web Technologies and Applications, ser. Lecture Notes in Computer Science, Y. Ishikawa, J. Li, W. Wang, R. Zhang, and W. Zhang, Eds. Springer Berlin Heidelberg, 2013, vol. 7808, pp. 120–131.

[6] M. McCord and M. Chuah, "Spam detection on twitter using traditional classifiers," in Autonomic and Trusted Computing, ser. Lecture Notes in Computer Science, J. Calero, L. Yang, F. Mrmol, L. Garca Villalba, A. Li, and Y. Wang, Eds. Springer Berlin Heidelberg, 2011, vol. 6906, pp. 175–186.

[7] C. Sibona and S. Walczak, "Unfriending on facebook: Friend request and online/offline behavior analysis," in Proc. on System Sciences (HICSS), the 44th IEEE Hawaii International Conference, January 2011, pp. 1–10.

[8] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging," in 23rd International Joint Conference on Artificial Intelligence (IJCAI '13), August 3-9 2013.

[9] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, July 19-23 2010, pp. 435–442.

[10] C. Lin, J. He, Y. Zhou, X. Yang, K. Chen, and L. Song, "Analysis and identification of spamming behaviors in sina weibo microblog," in Proceedings of the 7th Workshop on Social Network Mining and Analysis, vol. 5, 2013, pp. 1–9.

[11] Z. Chu, I. Widjaja, and H. Wang, "Detecting social spam campaigns on twitter," in Applied Cryptography and Network Security, ser. Lecture Notes in Computer Science, F. Bao, P. Samarati, and J. Zhou, Eds. Springer Berlin Heidelberg, 2012, vol. 7341, pp. 455–472.

[12] Y. Shen, J. Yu, K. Dong, and K. Nan, "Automatic fake followers detection in chinese micro-blogging system," in Advances in Knowledge Discovery and Data Mining, ser. Lecture Notes in Computer Science, V. Tseng, T. Ho, Z.-H. Zhou, A. Chen, and H.-Y. Kao, Eds. Springer International Publishing, 2014, vol. 8444, pp. 596–607.

[13] Z. Guo, Z. Li, H. Tu, and L. Li, "Characterizing user behavior in weibo," in Mobile, Ubiquitous, and Intelligent Computing (MUSIC), 2012 Third FTRA International Conference on, June 2012, pp. 60–65.

[14] X. Wu, Z. Feng, W. Fan, J. Gao, and Y. Yu, "Detecting marionette microblog users for improved information credibility," in Machine Learning and Knowledge Discovery in Databases, ser. Lecture Notes in Computer Science, H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezny, Eds. Springer Berlin Heidelberg, 2013, vol. 8190, pp. 483–498.

[15] Z. Zhang, B. Zhao, W. Qian, and A. Zhou, "Generating profiles for a lurking user by its followees' social context in microblogs," in Web Information Systems and Applications Conference (WISA), 2012 Ninth, Nov 2012, pp. 135–140.

[16] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), vol. 6, 2010, pp. 1–9.

[17] H. Liu, Y. Zhang, H. Lin, J. Wu, Z. Wu, and X. Zhang, "How many zombies around you?" in Proc. on Data Mining (ICDM), IEEE 13th International Conference on, Dec 2013, pp. 1133–1138.

[18] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: When bots socialize for fame and money," in Proceedings of the 27th Annual Computer Security Applications Conference, ser. ACSAC '11. New York, NY, USA: ACM, 2011, pp. 93–102.

[19] T. Hwang, I. Pearce, and M. Nanis, "Socialbots: Voices from the fronts," interactions, vol. 19, no. 2, Mar. 2012, pp. 38–45.

[20] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," CoRR, vol. abs/1407.5225, 2014, pp. 1–11.

[21] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," ACM Transactions on Knowledge Discovery from Data, vol. 8, no. 1, Feb. 2014, pp. 2:1–2:29.

The Bitcoin Network as Platform for Trans-Organizational Attribute Authentication

Jason Paul Cruz and Yuichi Kaji
Nara Institute of Science and Technology, Graduate School of Information Science
Nara, Japan
E-mail: jpmcruz@ymail.com, kaji@is.naist.jp

***Abstract*—The role-based access control (RBAC) is a natural and versatile model of the access control principle. In the real world, it is common that an organization provides a service to a user who owns a certain role that was issued by a different organization. However, such a trans-organizational RBAC is not common in a computer network because it is difficult to establish both the security that prohibits malicious impersonation of roles and the flexibility that allows small organizations/individual users to fully control their own roles. This study proposes a system that makes use of Bitcoin technology to realize a trans-organizational RBAC mechanism. Bitcoin, the first decentralized digital currency, is a payment network that has become a platform for innovative ideas. Bitcoin's technology, including its protocol, cryptography, and open-source nature, has built a good reputation and has been applied in other applications, such as trusted timestamping. The proposed system uses Bitcoin technology as a versatile infrastructure to represent the trust and endorsement relationship that are essential in RBAC and to realize a challenge-response authentication protocol that verifies a user's ownership of roles.**

Keywords-role-based access control; trans-organizational role; information security; Bitcoin; trusted-timestamping.

I. Introduction

A. Roles and Role-Based Access Control

Roles and titles are often used to distinguish the eligibility of people to access certain services. Such mechanism is modeled as the role-based access control (RBAC) [1] framework, which describes the access control relation among users and services. In RBAC, users are associated with roles, and roles are associated with services. This framework is compatible with the access control requirements of real-world organizations and is employed in the computer systems of many organizations/companies. However, it must be noted that RBAC is a versatile framework, and roles are often used in a trans-organizational manner. For example, students are often allowed to purchase computer software at an academic-discounted price. In this example, the "student" role that is issued by an organization (school) is used by another organization (computer shop) to determine if a guest is eligible to receive a certain service (discounted price). This kind of trans-organizational use of roles is common in face-to-face communication, but it is not obvious in computer networks. Even if one has a certain role (student role) that is issued by an organization (school), he/she has no systematic way of convincing a third-party organization (computer shop) that he/she really has that role.

To realize a trans-organizational RBAC mechanism in a computer network, a mechanism that prevents malicious users from disguising their roles is necessary. This requirement is naturally accomplished in real-world services with the use of physical certificates, such as passports and ID-cards, which are expected to be difficult to forge or alter. This problem, however, is not obvious in a computer system. Digital certificates [2] can be utilized as an analogue of physical certificates, but the use of digital certificates is not favorable because it requires considerable and continuous elaborations to maintain secure public-key infrastructures. Another less sophisticated approach to the security problem is to let a service-providing organization (the computer shop in the above example) inquire a role-issuing organization (school) about the user-role assignment. This approach works fine in some cases [3], but a focal point of this approach is the necessity for the agreed beneficial relationship among organizations. Consequently, it is difficult for a new organization to join the partnership, severely restricting the trans-organizational utilization of roles.

B. Bitcoin

Bitcoin is a decentralized global currency cryptosystem that has increased in value and popularity since its inception by Satoshi Nakamoto in 2008 [4]. As of March 2015, Bitcoin has a market capitalization of approximately 4 billion USD, market price per Bitcoin (BTC) of approximately 280 USD, and on average, 100,000 transactions daily [5]. The main purpose of Bitcoin is to enable a payment system and complete digital money that is secure and decentralized; that is, it is a peer-to-peer network powered by its users and with no central authority. To achieve this, transactions are publicly announced and the participants agree on a single history of these transactions. The transactions are grouped into blocks, given timestamps, and then published. The hash of each timestamp includes the previous timestamp to form a chain, making accepted blocks difficult to alter. Based on this security, Bitcoin features many favorable properties, including easy mobile payments, reliability, full control of one's own money, high availability, fast international payments, zero or low fees, protected identity, and privacy [6].

C. Bitcoin as an Infrastructure

The current study aims to develop a practical system that uses Bitcoin technology to realize the trans-organizational utilization of roles. We investigate a realization of a user-role assignment that is secure (users cannot disguise roles), user-oriented (users can disclose their roles to any organization), and open (anyone can verify if a user has a certain role that is managed and issued by another organization). The key ideas are to define correspondence between the roles issued by organizations and the users and to employ a challenge-response authentication protocol that will be used for verifying if a user really has an asserted role.

Bitcoin's protocol and cryptography make the proposed system suitable for the trans-organizational utilization and authentication of roles, and furthermore, allow flexible role management operations, such as the endorsement and management of roles, with relatively small realization cost.

The rest of this paper is organized as follows. Section II introduces RBAC and the different models associated with it. Section III discusses the Bitcoin protocol to show the security it provides. Section IV presents the structure and procedures of the proposed framework. Section V presents the role management features of the system. Section VI provides the conclusion and future work.

II. Models for the Role-Based Access Control

Among the many technical issues of the RBAC framework, this study mainly focuses on the realization of the user-role assignment in a trans-organizational scenario. Other issues of RBAC may be related to this study, but they are excluded from the scope of our discussion. To clarify the position of our study in the entire framework of RBAC, an abstract model of RBAC and its extension are discussed first.

In the simplest model of the RBAC [1], the access structure is defined by three sets and two relations; the set U of *users*, the set R of *roles*, the set S of *services*, a *user-role assignment* $\mathrm{UA} \subset U \times R$, and a *role-service assignment* $\mathrm{SA} \subset R \times S$. A user u is eligible to access a service s if and only if there is a role r such that $(u, r) \in \mathrm{UA}$ and $(r, s) \in \mathrm{SA}$. In real-world services, roles can be used in a trans-organizational manner. A role that was issued by a *role-providing entity* can be referred by a foreign *service-providing entity* to determine if a service should be given to an unknown guest. An interesting point here is that the role-providing entity is not always concerned about the service-providing entities. This suggests that the service-providing entity is not always allowed access to the user-role assignment, and thus, it needs to devise an alternative means to confirm if an unknown guest has a certain role or not. To deal with this kind of framework, we consider extending the basic model of RBAC by introducing a set of organizations.

The *trans-organizational RBAC* is defined similarly to the usual RBAC, but a set O of *organizations* is defined in addition to the sets of users, roles, and services. Furthermore, the set R of roles is partitioned into several subsets, with each subset of R associated with an element in O, that is, $R = R_{o_1} \cup \cdots \cup R_{o_n}$, where $o_1, \ldots, o_n \in O$ and $R_{o_i} \cap R_{o_j} = \emptyset$ if $i \neq j$. To make the relation among roles and organizations explicit, a role r in R_{o_i} is written as $o_i.r$. Similarly, the user-role assignment UA is partitioned into disjoint subsets; $\mathrm{UA} = \mathrm{UA}_{o_1} \cup \cdots \cup \mathrm{UA}_{o_n}$, where $\mathrm{UA}_{o_i} \subset U \times R_{o_i}$. Obviously, $o_i.r \in R_{o_i}$ means that the role $o_i.r$ is managed by the organization o_i and the assignment of users to $o_i.r$ is controlled by that organization o_i. In the trans-organizational RBAC, a user u demands a service s by asserting his/her role $o_i.r \in R_{o_i}$ that has been provided by a role-providing entity (organization) o_i. The service-providing organization provides the service to user u if and only if $(u, o_i.r) \in \mathrm{UA}_{o_i}$ and $(o_i.r, s) \in \mathrm{SA}$. Note that the test of $(o_i.r, s) \in \mathrm{SA}$ is easy for the service-providing organization because the assignment SA is defined by the organization itself. On the other hand, the test of $(u, o_i.r) \in \mathrm{UA}_{o_i}$, which is sometimes called an *authentication*, is not as obvious as the test of $(o_i.r, s) \in \mathrm{SA}$ because the assignment UA_{o_i} is defined by a foreign service-providing organization.

The trans-organizational RBAC will be realistic only if the authentication of roles $(u, o_i.r) \in \mathrm{UA}_{o_i}$ is accomplished. Physical certificates, such as passports and ID-cards, have been widely used for many years, but these certificates cannot be easily imported to the digitalized world over a computer network. Digital certificates have been studied for the replacement of physical certificates [2], but they are not widely accepted because of the cost issues for acquiring these certificates, keeping related keys secure, and maintaining a public-key infrastructure (PKI) [7][8] that should be always available. A less sophisticated but simpler approach is to arrange a mutual agreement between role-providing organizations and service-providing organizations. However, such a framework will be semi-closed and only include organizations that share identical benefits. The authors have studied another approach for realizing secure authentication of roles by utilizing a special cryptography known as hierarchical ID-based encryption [9]. The scheme in [9] offers some advantages over other existing approaches, but it necessitates several users with the same role to have an identical secret key, which is not favorable from a security viewpoint. Other comparable systems include decentralized multi-authority systems for attribute-based encryption (MA-ABE) and attribute-based signatures (MA-ABS) [10][11]. The MA-ABE in [10] is decentralized but requires a trusted setup of common reference parameters. The MA-ABS in [11] is also decentralized and does not necessitate a trusted setup, but it requires the setting of a public parameter for a prime order bilinear group and hash functions. Even though these schemes are decentralized, all users and organizations must agree on the parameters first. Confusion and implementation problems may arise if several communities use different parameters, and therefore, such systems must be initiated by somebody who has strong leadership, grand design, and sufficient financial power for implementation. Consequently, a scheme for a role authentication mechanism that is secure, practical, and easy to set up has yet to be established.

III. THE BITCOIN PROTOCOL

Bitcoin is a collection of cryptographic protocols that allow secure online transactions between users [12][13]. These users own digital wallets that handle the creation and storage of private keys and the corresponding public *Bitcoin addresses*. A user can send a certain value or amount of BTC to another user by creating a *transaction* with the sender's Bitcoin address/es as input/s and the receiver's Bitcoin address/es as output/s. Transactions are validated by miners and then recorded in a global public ledger called the *blockchain*. The validation of transactions requires some amount of computation, and the miner who succeeds in validating these transactions is rewarded or compensated with Bitcoins and the transactions fees for his/her efforts. Validated transactions cannot be altered unless an attacker has computation power that overwhelms the total computation powers possessed by all other miners.

A. Bitcoin Addresses

A Bitcoin address is 160-bit hash of the public key of an Elliptic Curve Digital Signature Algorithm (ECDSA). The public key undergoes several cryptographic processes (SHA-256, RIPEMD-160, and Base58 Encoding) to be converted into a valid Bitcoin address. A new ECDSA keypair is generated for each Bitcoin address. The private key is generated first, and then the public key is derived from the private key so that each private key corresponds to only one Bitcoin address. A user can create any number of Bitcoin addresses easily and for free, and thus, users usually use a digital wallet to store the private and public keypairs and the corresponding Bitcoin addresses (some wallet services create new Bitcoin addresses deterministically, i.e., using a seed). The users should backup and secure the private keys (or the wallet data file) because these private keys are needed to access the Bitcoins stored in the corresponding Bitcoin addresses. Each Bitcoin address contains a built-in checksum, and thus mistyped addresses can be detected. However, Bitcoins sent to a well-formed mistyped address with no owner are lost forever. Similarly, Bitcoins stored in Bitcoin addresses are lost if the owner loses control of the wallet file or the corresponding private keys.

A Bitcoin address is in the form of random numbers and letters, e.g., 19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6, and there are 2^{160} possible Bitcoin addresses that can be created. A Bitcoin address is considered to be "unique" as it is extremely unlikely for two users to independently generate the same Bitcoin address.

B. Transactions, Blocks, and the Blockchain

A transaction is a digitally signed data that is broadcasted to the Bitcoin network and then included in a block in the blockchain. A transaction contains the following information: transaction ID (used to identify the transaction in the blockchain); list of input addresses (addresses from which Bitcoins are transferred and should be outputs of previous transactions); list of output addresses (which contains the receiving addresses and the amounts of BTC being transferred). A sender of the transaction has to prove that he/she has control of the addresses in the list of input addresses by signing them (input addresses) with the corresponding private keys. Think of a Bitcoin address as a transparent vault where anyone can see or know the amount of Bitcoins inside, but only the one who has the key (private key) can spend or transfer the Bitcoins inside that vault.

C. Blocks and the blockchain

Transactions are grouped together in blocks and then recorded in the global ledger of the Bitcoin network. A block contains a record of transactions that have not been recorded in any previous block. Blocks are connected and linked together to form a blockchain, where a new block is added to the block that came before it. Every block contains the hash of the previous block, and thus, creating a chain that connects the first block (genesis block) to the current block.

A block contains, among other things, a hash of the previous block, a hash of the merkle root of valid transactions to be included in this block, and a nonce (a unique solution to a difficult mathematical puzzle), as shown in Figure 1. The entire blockchain and every transaction included in the blocks can be downloaded or viewed online using a blockchain browser.

D. Mining and Proof-of-Work

Blocks are added to the blockchain through the process called *mining*. This process uses a proof-of-work system wherein miners all around the world use special software to solve mathematical problems. The mathematical problem is inherently difficult to solve and requires a "brute force" solution; that is, miners scan and test for a nonce that gives a correct solution to the mathematical problem. During mining, the mining hardware of a miner (CPUs, GPUs, FPGAs, and ASICS) runs a cryptographic hashing function composed of two rounds of SHA256 on the block header. The mining software increments the nonce as the random element in the block header until a valid hash is found. To control the difficulty of mining, a parameter called a difficulty target is agreed upon by miners. The difficulty target can be regarded as a threshold that is used in such a way that a miner is required to find a nonce that makes the hash of the block lower than the difficulty target (equivalently, the hash values should start with a certain number of zeroes). To compensate for increasing hardware speeds, the difficulty target is adjusted every 2016 blocks so that it takes on average 10 minutes to find a valid nonce.

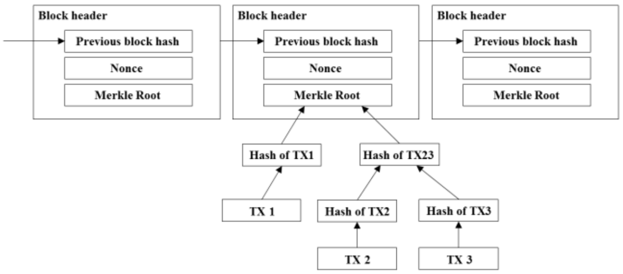


Figure 1. A simplified representation of the Bitcoin blockchain.

The difficulty target is expressed as the difficulty on creating the current block compared to generating the first block and is determined as follows:

$$\text{difficulty} = \frac{\text{difficulty\_1\_target}}{\text{current\_target}} \tag{1}$$

As of writing, the difficulty is 46,684,376,316.86 (the probability of each hash to be a valid solution is less than 10^{-20}) [5][14]. When a miner finds the correct nonce value that creates a hash value less than the target, it forwards the block to the rest of the nodes in the network. After validating the solution for the block, miners move on to solving for the next block.

Sometimes, more than one miner may find a valid solution at approximately the same time, consequently forking the blockchain and dividing the nodes to work on different versions of the blockchain. This inconsistency is resolved when the solution for the next block is found and one of the branches becomes longer. In the Bitcoin protocol, miners always work on the longest chain (the chain with the most work put into it), and thus, in case of a fork, the shorter chain is orphaned.

The miner who solves the block is awarded with newly "minted" Bitcoins (currently at 25 BTC) and the transaction fees of the transactions included in the solved block. This process of mining guides the issuance of Bitcoins and incentivizes miners to keep mining and approving transactions.

The security of Bitcoin relies on this proof-of-work system, which inherently means that a block cannot be modified without redoing the work spent on it, including the work spent on blocks chained after it. Given this design, as long as majority of the overall CPU power participating in the Bitcoin network is controlled by honest miners, an attacker will be outpaced by the honest miners, making it almost impossible to modify a published block.

E. *Attacks on the Bitcoin Network*

Some strategies, both theoretically and in practice, have been devised to attack the security of the Bitcoin protocol and possibly put it in danger. Some attacks have been designed for dishonest or rogue miners, i.e., those who do not follow the Bitcoin protocol, to get rewards higher than their contribution to the network.

These strategies include the pool hopping attack [15], the mining cartel attack [16], selfish mining [17], block withholding attack [18], and hardware attacks. These attacks are designed to infiltrate factors outside the blockchain, targeting the client side and stealing Bitcoins from them (pool and wallet infiltration). These attacks mainly aim to steal Bitcoins and/or gain higher rewards and not to modify transactions or the blockchain. Therefore, Bitcoin's security remains intact and "backed by math". For our purposes, the transactions between organizations and users will remain secure and unmodifiable because they are recorded in the blockchain.

IV. PROPOSED SCHEME

A. *Overview*

The proposed system is a non-conventional authentication mechanism that is suitable for the trans-organizational utilization of roles. The idea is to provide an irrefutable proof of the role of a user issued by an organization by verifying the connection of the user to the organization through the Bitcoin blockchain. Consider for example that A-university would like to manage a "student" role for its students. First, it would perform a Bitcoin transaction using its own public Bitcoin address/es as input/s and the corresponding students' public Bitcoin address/es as output/s. Upon request for a service from an unknown user who asserts that he/she possesses the student role of A-university, a service-providing organization, for example a restaurant, will verify the Bitcoin transaction containing the Bitcoin addresses of A-university and the student, which connects the student role managed by A-university to the output address in the transaction. After establishing the connection, the restaurant can verify (through a challenge-response protocol) if the unknown user has access to the output address in the transaction, which finally connects the student role from A-university to the unknown user.

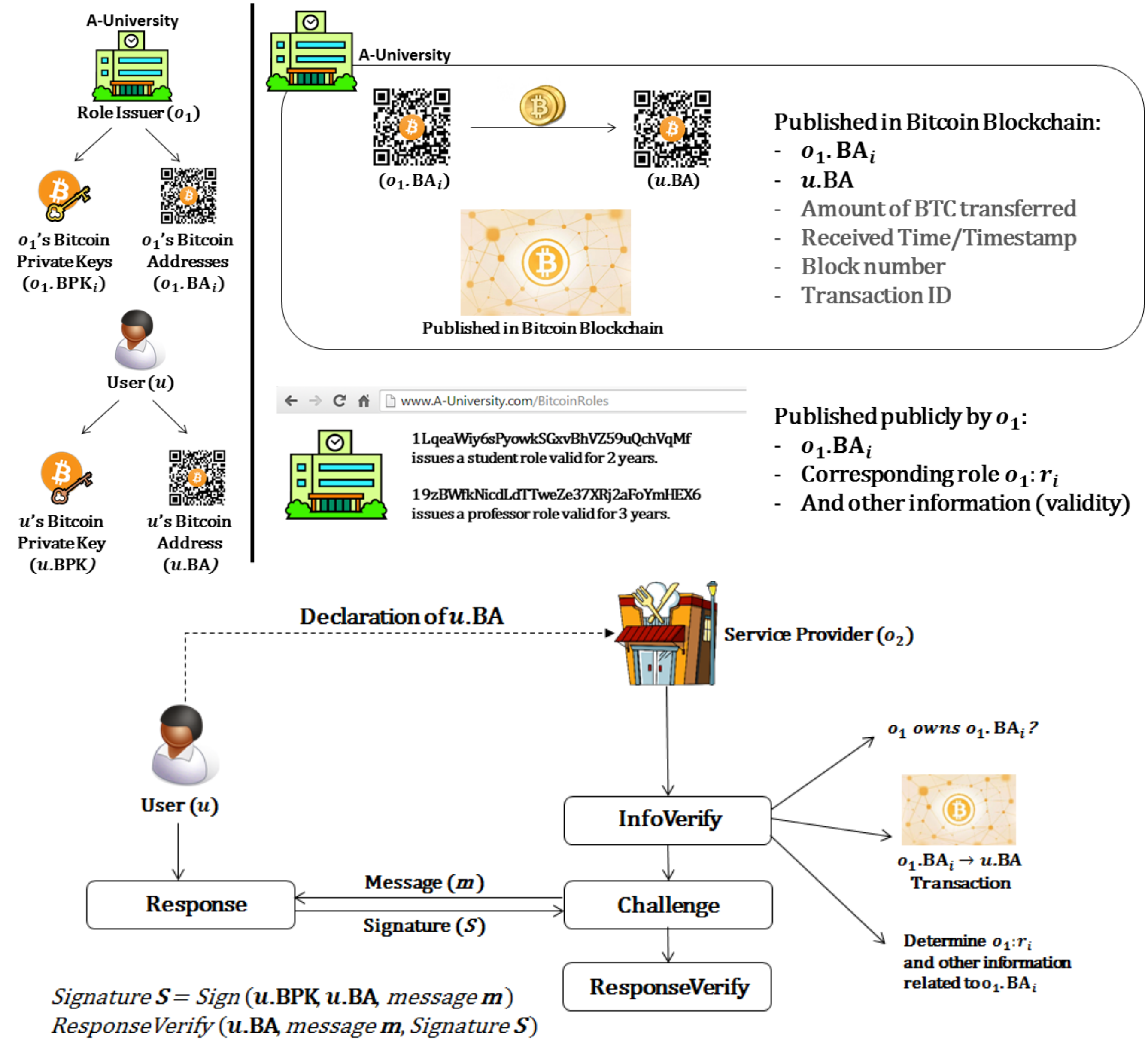
Note that the restaurant does not have to know anything about the role beforehand, and does not have to make any contract or inquiry to A-university that has assigned the role to the student because the details needed by the restaurant are published publicly and/or possessed by the user (details below). In the proposed system, there is no essential difference between users and role-issuing organizations because they both can be the sender and receiver in the Bitcoin transactions (but for simplicity and explanatory purposes, the role-issuing organizations will be differentiated from the users).

B. *Procedures*

Figure 2 shows the overall structure of the proposed system. In this model, we assume that the role-issuing organizations are Bitcoin users while the users and service-providing organizations may or may not be Bitcoin users. Bitcoin user means that the entity owns a Bitcoin wallet and performs Bitcoin transactions.

1) *Pre-requisites*

An organization (o_1) generates n Bitcoin addresses, where n is the number of roles that o_1 wants to manage. The creation of these Bitcoin addresses (and the corresponding private keys) can be accomplished using several options, including Bitcoin wallets and online/offline Bitcoin address generators. After generating the n keypairs, o_1 keeps the private keys secret and secure, and publishes the list of pairs of Bitcoin addresses and corresponding roles using chosen media (e.g., Website, database, etc.) to make it available to the public. We write $o_1.\mathrm{BPK}_i$, $o_1.\mathrm{BA}_i$, and $o_1{:}r_i$ for the private key, Bitcoin address, and the role that is associated with the address $o_1.\mathrm{BA}_i$, respectively, where $1 \leq i \leq n$.


A-University
Role Issuer (o_1)
o_1's Bitcoin Private Keys (o_1.BPK$_i$)
o_1's Bitcoin Addresses (o_1.BA$_i$)
User (u)
u's Bitcoin Private Key (u.BPK)
u's Bitcoin Address (u.BA)
A-University
(o_1.BA$_i$)
(u.BA)
Published in Bitcoin Blockchain
Published in Bitcoin Blockchain:
- o_1.BA$_i$
- u.BA
- Amount of BTC transferred
- Received Time/Timestamp
- Block number
- Transaction ID
www.A-University.com/BitcoinRoles
Published publicly by o_1:
- o_1.BA$_i$
- Corresponding role o_1:r_i
- And other information (validity)
Declaration of u.BA
Service Provider (o_2)
User (u)
o_1 owns o_1.BA$_i$?
InfoVerify
o_1.BA$_i$ → u.BA Transaction
Response
Message (m)
Signature (S)
Challenge
ResponseVerify
Determine o_1:r_i and other information related to o_1.BA$_i$

1LqeaWiy6sPyowkSGxvBhVZ59uQchVqMf issues a student role valid for 2 years.

19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6 issues a professor role valid for 3 years.

Signature S = *Sign* (u.BPK, u.BA, *message* m)
ResponseVerify (u.BA, *message* m, *Signature* S)

Figure 2. Overview of the proposed structure.

The publication of these Bitcoin addresses will serve as proof that o_1 owns and manages the addresses (it should be noted that o_1 will not gain any benefit from publishing Bitcoin addresses that it does not own, and thus, will only publish Bitcoin addresses it owns).

Similarly, a user (u) generates a pair of a private key u.BPK and a Bitcoin address u.BA. Alternatively, o_1 can generate the (u.BPK, u.BA) keypair and send it to u through a secure communication channel. Note however, that it is recommended by the Bitcoin community that only the one who created the keypair should be in possession of the keypair because the private key is used for accessing the Bitcoins stored in the corresponding address.

2) Creating the role-issuer and user connection

The organization o_1 creates a simple Bitcoin transaction using o_1.BA$_i$ as input address and u.BA as output address. In this transaction, o_1 sends an arbitrary amount of Bitcoins to u; currently the minimum amount that can be used for a transaction to be considered valid is 0.00010001 BTC = 0.03 USD (1 satoshi = 0.00000001 BTC plus 0.0001 BTC required transaction fee). Optionally, o_1 can include a higher transaction fee or miners' fee if it wants its transaction to be prioritized in the current round of solving for the block (but for our purposes, if the time of confirmation is not vital, the minimum transaction fee is sufficient). After confirming the details of the transaction, o_1 sends the transaction to the Bitcoin network awaiting for confirmations from miners that it is permanently included in a block in the blockchain. Once included in the blockchain, certain details will be publicly available, including $o_1.\mathrm{BA}_i$, u.BA, the amount of BTC transferred, transaction ID, block number, received time, and the time it was included in the block.

3) Verifying a user-role assignment

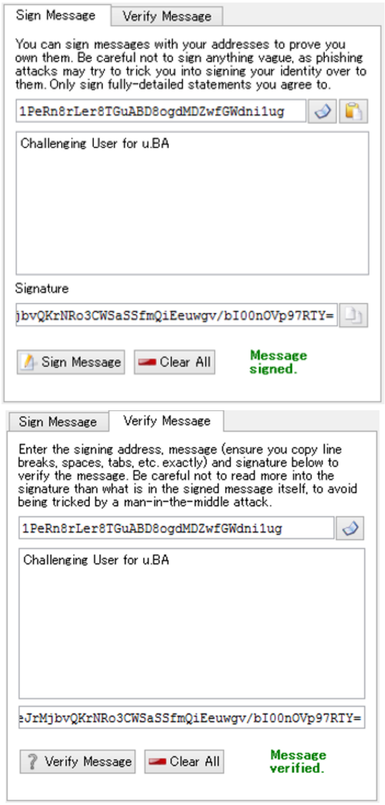
Assume that user u visits a service-providing organization o_2 and asserts that he/she has the role of $o_1{:}r_i$ that was issued by o_1.

The organization o_2 inquires u for the Bitcoin address, say u.BA, that was granted the asserted role of $o_1{:}r_i$ from o_1. Then o_2 will (i) determine the Bitcoin address $o_1.\mathrm{BA}_i$ that is associated with the role $o_1{:}r_i$, (ii) confirm the existence of the transaction from $o_1.\mathrm{BA}_i$ to u.BA, and (iii) verify if u is the genuine owner of u.BA. The Bitcoin address $o_1.\mathrm{BA}_i$ can be found through the medium where o_1 published the Bitcoin addresses it owns. The confirmation of the transaction can be done by using a blockchain browser or a program similar to that. Steps (i) and (ii) assure o_2 that the role $o_1{:}r_i$ and other related information associated with $o_1.\mathrm{BA}_i$ are assigned by o_1 to the owner of u.BA. The ownership of u.BA is verified by a challenge-response protocol where ECDSA keys that are associated with the Bitcoin address u.BA are utilized.

4) Challenge-Response Protocol

The organization o_2 chooses an arbitrary data m and requests u to sign it, together with u.BA, using the private key u.BPK. The signature is defined by S = *Sign*(u.BPK, u.BA, m), and thus a correct S will only be created if u has u.BPK. User u then sends the signature back to o_2 and o_2 will verify using the function *ResponseVerify*(u.BA, m, S). Examples of signing/verifying a message to prove ownership of a Bitcoin address are shown in Figure 3.

Remark that o_2 can confirm if u has access to the role $o_1{:}r_i$ without querying o_1, and that u has little chance to disguise his/her role.



You can sign messages with your addresses to prove you own them. Be careful not to sign anything vague, as phishing attacks may try to trick you into signing your identity over to them. Only sign fully-detailed statements you agree to.

1PeRn8rLer8TGuABD8ogdMDZwfGWdni1ug

Challenging User for u.BA

Signature

jbvQKrNRo3CWSaSSfmQiEeuwgv/bI00nOVp97RTY=

Message signed.

Enter the signing address, message (ensure you copy line breaks, spaces, tabs, etc. exactly) and signature below to verify the message. Be careful not to read more into the signature than what is in the signed message itself, to avoid being tricked by a man-in-the-middle attack.

1PeRn8rLer8TGuABD8ogdMDZwfGWdni1ug

Challenging User for u.BA

eJrMjbvQKrNRo3CWSaSSfmQiEeuwgv/bI00nOVp97RTY=

Message verified.

Figure 3. The signing (top) and verifying (bottom) of a message features of a Bitcoin wallet.

V. Role Management

In the proposed framework, the relation between users and roles is represented by the users' possession of the private keys that prove they own the corresponding Bitcoin addresses. This approach involves a possible security risk; the leakage and loss of keys.

A. Personalization of roles

If a user leaks his/her private key, then the people who happen to know the key can also prove ownership of the corresponding Bitcoin address (which in turn can be used to prove that a role associated with the address was assigned to them). Note that the intended user can still prove ownership of the Bitcoin address even after leakage, although such an inappropriate usage of keys can obstruct fair and reliable access control. To deter such irresponsible behavior of users, the proposed system offers three possible measures:

1. Given that the proposed system uses Bitcoin technology, it inherently has a "traitor tracing" capability because the Bitcoin addresses are unique (thus, they can be possibly mapped to users) and the transactions are published publicly. Thus, if a user receives a leaked key and maliciously uses the role associated to the corresponding Bitcoin address, a consequent investigation can possibly lead to the original user associated with the Bitcoin address.
2. Role-issuing organizations can "personalize" roles by including some unique identifiers (which can be encrypted as well) to the data it will publish publicly. For example, A-university can publish "19zBWfkNicdLdTTweZe37XRj2aFoYmHEX6 issues a 'student' role to student #123 with 6 months validity."
3. Role-issuing organizations can "personalize" roles by making use of a public note that is included in the Bitcoin transaction. It should be noted that this public note is a relatively new feature offered by an online wallet (blockchain.info) [19] and is not part of the Bitcoin protocol.

With these measures, a student will be more conscious of leaking/losing his/her key to another person because he/she will have the risk of being identified and subsequently punished for irresponsible behavior. The theft/loss of keys remains a risk, but such risk also exists for ID-cards used in the real world.

B. *Role Re-issuance*

If the private keys are lost or forgotten, or if access to the digital wallet is lost or forgotten, then control over the corresponding Bitcoin addresses is also lost. The ownership of the Bitcoin addresses cannot be verified or proven without the corresponding private keys.

In this case, the role-issuing organizations can easily re-issue the roles by creating another Bitcoin transaction to the new Bitcoin address of the compromised user. The overhead of role re-issuance is relatively low for both the role-issuing organizations and the user.

Moreover, to make sure that the compromised Bitcoin addresses will not be used maliciously, a role-issuing organization can create a revocation list containing these addresses in the media where it publishes the Bitcoin addresses it owns.

C. *Additional Security Measures*

Wallets are the most common target of attacks, but of course, security measures have been implemented and are recommended to minimize such cases. In the proposed system, the purpose of the Bitcoin transaction is to connect the user to an organization and to a role. If the user is a Bitcoin user, he/she is recommended to use other wallets or other addresses to store Bitcoins. Ultimately, the user only needs to store the private key safely, and even keep it offline. The challenge-response protocol can be performed offline. Moreover, an attacker with no prior knowledge of the proposed system and the role associated with the address will have no motives or incentives to steal the private key (even if an attacker can steal the private keys, the addresses will not contain large amounts of Bitcoins to steal).

D. *Endorsement*

The Bitcoin network provides a natural connection between Bitcoin addresses published in the blockchain. This function can be used to realize some personal activities that are not considered in the conventional RBAC approach. One possible example is the endorsement of another person. In the real world, an endorsement among individuals sometimes plays an important role. Semi-closed organizations, such as academic societies and golf clubs, have the tradition or policy that a newcomer must be endorsed or referred by a current member. This mechanism can be realized by extending the proposed system.

Based on the system shown in Figure 2, consider for example that Alice (u) is an authorized member of XYZ golf club (o_1). This relationship is realized by the Bitcoin transaction from o_1.BA to u.BA. If Alice would like to endorse Bob (u_2) to o_1, then she can similarly create a Bitcoin transaction from u.BA to u_2.BA, linking their addresses. Then, Bob can go to o_1 and declare u_2.BA. The club can look up the blockchain and check that u_2.BA was endorsed by u.BA, which was originally endorsed by the club, as represented by o_1.BA. Once the connection is established, o_1 can verify if u_2 is the owner of u_2.BA by using the challenge-response protocol. By querying the blockchain and through the challenge-response authentication, the club does not have to inquire Alice for the verification of the endorsement.

E. *Trusted Timestamping as Proof of Validity or Expiration Dates*

Trusted timestamping is the process of securely creating a proof, i.e., timestamp, of the creation or modification time of a document. It is used for proving that certain information or document existed at some point in time and has not been tampered or modified since.

Traditional timestamping processes follow the RFC 3161 standard, wherein the timestamp is issued by a trusted third party acting as a Time Stamping Authority (TSA) [20].

The Bitcoin network features a timestamp server used in the blockchain to link the blocks together in a chronological manner. This timestamp server has been used, outside the main purpose of Bitcoin, as a trusted timestamping mechanism for digital documents given that it is secure (extremely difficult to attack and modify), robust (DoS resistant), and a trustworthy source of time (the time a transaction is included in the blockchain) [21][22]. Put simply, a hash of the data that a user wants to timestamp is converted into a Bitcoin address. The timestamping service (or the user his/herself) then creates a Bitcoin transaction and makes a small payment to the converted Bitcoin address. This transaction is then stored in the public blockchain. Anyone who wants to verify the point in time a data (the hash of it) existed can be connected to the time the transaction that includes the corresponding converted Bitcoin address was included in the blockchain.

This timestamping scheme is innovative and provides additional features as compared to the traditional trusted timestamps issued by TSAs, which are prone to data corruption and tampering. This timestamping scheme is decentralized and is not controlled by a single authority; it can easily be accessed over the Internet and is always available; it is anonymous (one's identity, the data being given a timestamp, or even the fact that one wants something to be given a timestamp are kept secret); it is relatively low cost; the timestamp is accurate; and it is secure.

The timestamp server of Bitcoin provides a natural solution to the inclusion of expiration dates or validity of the roles in the proposed system. The role-issuing organization includes the expiration dates or validity of the roles it manages in the information it publishes publicly. In this way, the service-providing organization can verify the validity of a role simply by investigating the timestamp of the block where the transaction was included in the blockchain and comparing it with the details published by the role-issuing organization.

VI. CONCLUSION AND FUTURE WORK

The Bitcoin protocol was utilized as an infrastructure to realize a trans-organizational RBAC and represent the trans-organizational usage of roles. The proposed system provides a secure mechanism for verifying the user-role assignments of organizations. Compared to other similar approaches, the proposed scheme provides more flexibility and autonomy while maintaining security. This mechanism allows the realization of many collaborative right managements that are common in physical communication but are difficult to implement over computer networks. Finally, the timestamping mechanism provided in the Bitcoin protocol provides a natural solution to the inclusion of expiration dates or validities in the created roles. Future research will focus on the realization and quantitative analysis of a more comprehensive hierarchical, multi-authority, and attribute-based system, which can offer additional role management features, such as role transfer and access policies in terms of Boolean formulas, and on the development of a prototype for easier use and interoperability.

REFERENCES

[1] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” IEEE Computer, vol. 29, number 2, 1996, pp. 38–47.

[2] S. Farrell and R. Housley, “An internet attribute certificate profile or authorization,” RFC 3281, 2002.

[3] Internet2, “The Shibboleth System,” accessible online: http://shibboleth.internet2.edu/ [accessed: 2015-03-05].

[4] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008, URL: https://bitcoin.org/bitcoin.pdf [accessed: 2015-03-05].

[5] Blockchain Info, “Currency Stats,” URL: https://blockchain.info/stats [accessed: 2015-03-05].

[6] Bitcoin.org, “Bitcon for Individuals,” URL: https://bitcoin.org/en/bitcoin-for-individuals [accessed: 2015-03-05].

[7] C. M. Ellison et al., “SPKI certificate theory,” RFC 2693, 1999.

[8] P. Gutmann, “Simplifying public key management,” IEEE Computer, vol. 37, number 2, 2004, pp. 101–103.

[9] J. P. Cruz and Y. Kaji, “Trans-Organizational Role-Based Access Control in Android,” INFOCOMP 2014, pp. 114–119.

[10] A. B. Lewko and B. Waters, “Decentralizing Attribute-Based Encryption,” EUROCRYPT, 2011, pp. 568–588.

[11] T. Okamoto and K. Takashima, “Decentralized Attribute-Based Signatures,” Public-Key Cryptography-PKC, 2013, pp. 125–142.

[12] Bitcoin Wiki, “Protocol Specification,” URL: https://en.bitcoin.it/wiki/Protocol_specification [accessed: 2015-03-05].

[13] Bitcoin.org, “Bitcon Developer Guide,” URL: https://bitcoin.org/en/developer-guide#block-chain [accessed: 2015-03-05].

[14] Blockexplorer, URL: http://blockexplorer.com/q/probability [accessed: 2015-03-05].

[15] M. Rosenfeld, “Analysis of Bitcoin Pooled Mining Reward Systems,” 2011, URL: https://bitcoil.co.il/pool_analysis.pdf [accessed: 2015-03-05].

[16] RHorning, “Mining cartel attack,” Bitcoin Forum, 2010, URL: https://bitcointalk.org/index.php?topic=2227.0 [accessed: 2015-03-05].

[17] I. Eyal and E. G. Sirer, “Majority is not Enough: Bitcoin Mining is Vulnerable,” Financial Cryptography and Data Security, 2014.

[18] N. T. Courtois and L. Bahack, “On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency,” CoRR, 2014, URL: http://arxiv.org/pdf/1402.1718v5.pdf [accessed: 2015-03-05].

[19] Blockchain Info, “Blockchain Website FAQ,” URL: https://blockchain.info/wallet/website-faq [accessed: 2015-03-05] .

[20] SSL4NET, “Trusted Timestamping,” URL: http://www.ssl4net.com/technology/trusted-timestamping [accessed: 2015-03-05].

[21] J. Klusácek, “Trusted Timestamp in Cryptocurrency Block Chain,” Student EEICT, 2014.

[22] BTProof, “Trusted timestamping on the Bitcoin blockchain,” URL: https://www.btproof.com/ [accessed: 2015-03-05].

Semantics-based Expansion of Search Queries Enforcing Lateral Thinking

Fabrizio Smith and Francesco Taglino

Institute for Systems Analysis and Informatics
National Research Council
Rome, Italy
Email: fabrizio.smith@iasi.cnr.it
francesco.taglino@iasi.cnr.it

Oscar Eijsermans

Department of Informatics and Automation
University Roma Tre
Rome, Italy
Email: oscareijsermans@hotmail.com

***Abstract*—Web search engines are equipped with query expansion facilities to reformulate a seed query and improve retrieval performance. However, such techniques are usually used in accordance with traditional Information Retrieval approaches, which do not distinguish between creative and conventional uses of languages, or between literal and non-literal meanings. But to support a more creative search, with the ultimate objective of being surprised or inspired by the results, non-literal relationships between queries and the texts that they match should be facilitated. This paper presents a query expansion method with a lateral thinking approach, by suggesting, starting from a *seed term* given by the user, a set of lists of terms representing conceptual paths, each of which starts from the seed term. Each term in the path is reached by traversing pre-identified relationships in a given semantic network, while the selection of a specific term is driven by the assessment of a distance metrics between terms. The paper also presents a software implementation of the method, which can be accessed as a mobile web app.**

Keywords–Lateral Thinking; Computational Creativity; Search Queries; Semantics.

I. Introduction

Lateral thinking [1], the term was coined by the physician Edward De Bono, is an attitude for addressing problems through an indirect and creative approach. Lateral thinking leverage reasoning that is not immediately obvious, involving ideas that may not be obtainable by using only traditional step-by-step logic. When we search for something, we are used to follow traditional pattern-based approaches. But in any patterning system, how argued by De Bono, there is an absolute, and even logic, need for something like lateral thinking, in order to yearn for something new that can further trigger creative and innovative behaviors [2].

The most common engines for searching resources over the Web evolved a lot. Currently, they do not just search for resources that exactly match keywords representing users' criteria. Indeed, most of them are equipped with query expansion facilities [3], whose aim is to reformulate a seed query to improve retrieval performance. Common query expansion techniques involve: finding synonyms of words, finding all the various morphological forms of words by stemming each word in the search query; fixing spelling errors and automatically searching for the corrected form or suggesting it in the results; weighting the terms in the original query.

In [4], a classification of automated query expansion techniques is presented. Such a classification is organizaed into: (*i*) linguistic analysis, which leverages global language properties such as morphological, lexical, syntactic and semantic word relationships; (*ii*) corpus-specific (global) techniques, which analyze the contents of a full database to identify features used in similar ways; (*iii*) query-specific (local) techniques, which take advantage of the local context provided by the query; (*iv*) search log analysis techniques, which are based on the idea of mining query associations that have been implicitly suggested by previous users; (*v*) web data techniques, which are based on anchor texts that are often succinct descriptions of the destination page and as such, can be very similar to search queries. However, such techniques are usually used in accordance with traditional information retrieval approaches, which do not distinguish between creative and conventional uses of languages, or between literal and non-literal meanings [5]. But to support a more creative search, with the ultimate objective of being surprised or inspired by the results, non-literal relationships between queries and the texts that they match should be exploited.

This paper presents a query expansion method enforcing a lateral thinking approach, by suggesting, starting from a *seed term* given by the user, a set of lists of terms representing conceptual paths, each of which starts from the seed term. Each term in the path is reached by traversing pre-identified relationships in a given semantic network, while the selection of a specific term is driven by the evaluation of a distance metrics between terms.

The method does not stick to any specific implementation constraint or to any specific knowledge resource needed for implementing the method itself. However, the paper proposes an implementation of the method based on the use of Wordnik[6] as the knowledge base the conceptual paths belong to, and the Lin similarity method [7], applied to WordNet[8], for evaluating the distance between terms. The described implementation has been also released as a mobile web application named LaSearch [9].

The rest of the paper is organized as follows. Section II reports about some computational creativity works for query expansion and engines for inspiring creativity. Section III presents the proposed query expansion method. Section IV describes the architecture of the query expansion engine and its current implementation. Finally, Section V presents the conclusions and future works.

ISBN: 978-1-61208-407-7

II. Related Works

Considering the classification elaborated in [4], and briefly presented in the Introduction, the proposed method belongs to the family of linguistic analysis techniques, which are typically based on the exploitation of dictionaries, thesauri, or other similar knowledge representation sources. Some related initiatives are here reported.

Concept creation is based on the ability to hypothesize and create new concepts to suit a given situation [10]. [11] presents a model of exploratory creativity that uses the WordNet ontology [12] as a basis for inducing the concepts that WordNet appears to lack and which should be profitably added.

Analogical expressions use terms from one domain of discourse to allude to terms in another systematically parallel domain of discourse. Analogy is thus useful when one knows of, or suspects, the existence of a given concept but does not know how it is lexicalized (e.g., *the bible of Islam* to mean the *Koran*). [13] uses a lexical ontology like WordNet to resolve analogies to be used for query expansion. [14] envisages a creative process that can take any given target description and using a pre-stored collection of domain descriptions, identifies potentially creative source domains with which to re-interpret the given problem.

Concept combination is a family of conceptual mechanisms whereby a concept is used as a referential proxy for another strongly associated one, e.g., *lexical metaphors* or *polysemy*. For instance, metaphor is a highly-generative conceptual phenomenon that can be used to create a wide range of linguistic expressions that refer to the same concept [15]. The metaphor process could be directly applied if we wish to retrieve documents that allude to a search concept figuratively rather than literally, in order to expand the search query with plausible figurative lexicalizations of this concept. With this respect, [16] introduces a computational model based on creative metaphors to address non-literal meanings of terms in information retrieval systems.

Bisociation [17] seeks to combine elements from two or more incompatible concepts or domains to generate creative solutions and insight. [18] provides a formal definition of bisociation in order to facilitate bisociation by connecting the knowledge bases of an intelligent agent in the context of a concrete problem, situation or event.

Finally, recent proposals of *inspiration and creative search engines* are worth mentioning. Seenapse[19] is an inspiration engine that helps people to come up with creative ideas, by allowing users to browse mental associations made by people all over the world, in order to expand their own possibility of having a Eureka moment. YossarianLives[20] is a metaphorical search engine, designed to spark creativity by returning disparate but conceptually related terms. ELISE (Evolutionary Learning Interactive Search Engine) aims to decrease search results precision in order to keep some diversity in the retrieved documents. ELISE is based on the cost-efficient Parisian approach and interactive evolutionary algorithms to specialise existing search engines with evolved user profiles [21].

III. Overall Approach

The proposed method is based on the exploitation of a semantic network aiming at suggesting a set of query expansions in the form of paths of terms all starting from a seed term specified by the user as the initial search criterion. A conceptual path is a list of terms that belong to the semantic network and that can be reached by traversing a given relationship in the semantic network itself, starting from the seed term. The length of the path, which depends on the number of iterations of the algorithm at the core of the proposed method, is decided by the user, while the method defines the criteria for traversing the semantic network and applied to build the conceptual paths.

At each iteration, the method first identifies the most ”promising” current paths to expand, and then, for each of them, builds all the possible expansions. For each path, all the terms that in the semantic network can be reached from the tail of the path (the tail of a path is the last term added to the path), by traversing a given relationship, are used to generate the same number of expansions. Specifically, if the tail of a path is related to *n* terms, not yet belonging to that path, *n* expansions of the path will be produced where only one term will contribute to each path. The most ”promising” paths are identified by ranking the tails of the paths with respect to the defined *expansion divergence* function, which is computed by applying a distance metrics between terms. In particular, the *expansion divergence* function, which represents the lateral thinking component, privileges those terms that are closer to the mean distance of all the possible candidates with respect to both its generator (i.e., its preceding term in the path) and the seed term. The mean distances are taken as references, because the method wants to add a new term to a path by expressing a lateral thinking attitude in a ”prudent” way, that is, avoiding that the distances of an added term from both its generator and the seed term increase too much and too fast (i.e., in very few iterations). In addition, the method requires that a term can be added to a path if its distance from the seed term is higher than the distance from its generator term (*Expansion Constraint*).

As a convention, in the rest of the paper, any lower case letter represents a term, and in particular, *t* represents the seed term. The description of the algorithm is supported by the following definitions.

A. Definitions

Definition 1 (Relatedness): the Relatedness relationship *rel* between terms is a linguistic or semantic relation in a given semantic network *SN*, i.e., $(x, y) \in$ *rel* if *y* is directly linked to *x* in *SN*.

Definition 2 (Conceptual Path): A conceptual path (*cp*) starting from the seed term t_1, and having lenght equal to *n*, is a list of *n* terms belonging to *SN* that can be reached, starting from t_1, by traversing the Relatedness relationship.

$$cp = [t_1, t_2, ..., t_n],$$
$$where \; \forall \; t_k \; (t_k, t_{k+1}) \in rel, k = 1...n-1. \qquad (1)$$

Definition 3 (Possible Expansion): the Possible Expansion *PExp*(x) of a generic term *x* is the set of terms *y*, such that $(x, y) \in$ *rel*, where $y \neq x$ and $y \neq t$.

$$PExp(x) = \{y|(x, y) \in rel, y\} \neq xandy \neq t. \qquad (2)$$

Definition 4 (Term Distance): the Term Distance *dist*(x, y) between the terms *x* and *y*, is the distance between *x* and *y*, in accordance with a given metrics (e.g, a linguistic or semantic distance metrics).

Definition 5 (Mean Distance): the Mean Distance *md*(*PExp*(*x*), *y*) of the terms in *PExp*(*x*) with respect to a given term *y*, is the arithmetic mean of the Terms Distance between each term *w* in *PExp*(*x*) and *y*.

$$md(PExp(x), y) = \sum_{w \in PExp(x)} \frac{dist(w, y)}{|PExp(x)|} \qquad (3)$$

Definition 6 (Local Mean Divergence): the Local Mean Divergence *lmd*(y, x) of a term *y* in *PExp*(*x*) is the absolute value of the difference between the Mean Distance between *PExp*(*x*) and *x*, and the Terms Distance between *x* and *y*.

$$lmd(y, x) = |md(PExp(x), x) - dist(x, y)|, \\ where\ y \in PExp(x) \qquad (4)$$

Definition 7 (Absolute Mean Divergence): the Absolute Mean Divergence *amd*(y, x) of a term *y* in *PExp*(*x*) is the absolute value of the difference between the Mean Distance between *PExp*(*x*) and *t*, and the Terms Distance between *y* and *t*.

$$amd(y, x) = |md(PExp(x), t) - dist(y, t)|, \\ where\ y \in PExp(x) \qquad (5)$$

Definition 8 (Expansion Divergence): the Expansion Divergence *div*(y, x) of a term *y* in *PExp*(*x*) is the sum of the Local Mean Divergence of *y* in *PExp*(*x*) and the Absolute Mean Divergence of *y* in *PExp*(*x*).

$$div(y, x) = lmd(y, x) + amd(y, x) \qquad (6)$$

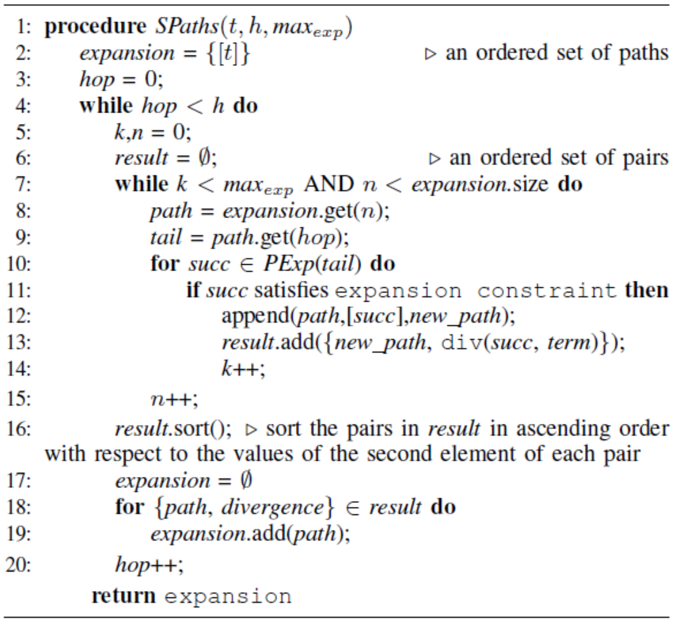
Definition 9 (Suggested Paths): the Suggested Paths *SPaths*(t, h, max_{exp}) of the term *t* in *h* iterations (hops) is the set of conceptual paths, all having *t* as the head, having length equals to *h*+1, and resulting by *h* iterations of the algorithm in Figure 1, in which, at most max_{exp} terms are expanded at each iteration.

Definition 10 (Expansion Constraint): Given a conceptual path $cp = [t, t_1, ..., t_h]$ belonging to *SPaths*(t, h, max_{exp}), and any two adjacent concepts $t_k andt_{k+1}$ in *cp*, the Term Distance between the seed term *t* and t_{k+1}, must be greater than the Term Distance between *t* and t_k.

$$dist(t, t_{k+1}) > dist(t, t_k), \\ where\ t_k\ and\ t_{k+1} \in cp = [t, t_1, ..., t_h], \\ cp \in SPaths(t, h, max_{exp}), k = 1, ..., h - 1. \qquad (7)$$

The rationale behind the Expansion Constraint is to enforce the following two principles:

- *laterality*, that is achieving an incremental growth of the lateral thinking attitude in the expansion of the seed term at each iteration (hop);
- *reliability*, that is guaranteeing that whenever a *boundary* term (i.e., a term presenting the maximum distance value with respect to the seed) is considered in a given hop, it will not be expanded in the subsequent hop; intuitively, a *boundary* term saturates the admissible lateral thinking attitude, and thus, the fitness of the terms belonging to its possible expansions cannot be properly characterized.



```
procedure SPaths($t, h, max_{exp}$)
    expansion = {[t]}                          ▷ an ordered set of paths
    hop = 0;
    while hop < h do
        k,n = 0;
        result = ∅;                            ▷ an ordered set of pairs
        while $k < max_{exp}$ AND $n <$ expansion.size do
            path = expansion.get(n);
            tail = path.get(hop);
            for succ ∈ PExp(tail) do
                if succ satisfies expansion constraint then
                    append(path,[succ],new_path);
                    result.add({new_path, div(succ, term)});
                    k++;
            n++;
        result.sort();  ▷ sort the pairs in result in ascending order
    with respect to the values of the second element of each pair
        expansion = ∅
        for {path, divergence} ∈ result do
            expansion.add(path);
        hop++;
    return expansion
```

Figure 1. The Suggested Paths algorithm

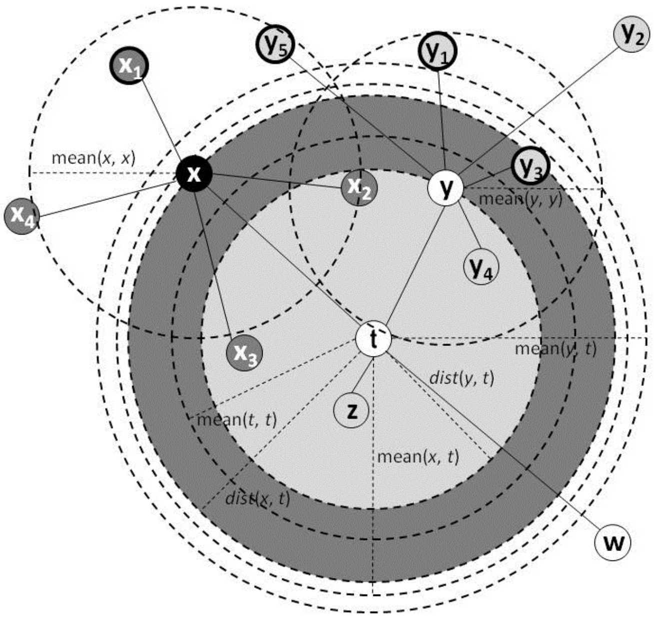


Figure 2. An example of term expansion and terms suggestion

B. Example

Here, an example of application of the presented method. Figure 2 elaborates about *SPaths*$(t, 2, 2)$, that means to search for conceptual paths of length 3 (applying two iterations of the algorithm in Figure 1), by expanding at most two terms at each iteration.

TABLE I. VALUES ABOUT TERMS IN *PExp*(*t*)

term	*dist(term, t)*	*lmd(term, t)*	*amd(term, t)*	*div(term, t)*
y	0.287	0.044	0.044	0.088
x	0.390	0.059	0.059	0.117
w	0.488	0.157	0.157	0.315
z	0.159	0.172	0.172	0.344

TABLE II. Values about terms in *PExp*(y)

y_i	*dist*(y_i, y)	*lmd*(y_i, y)	*dist*(y_i, t)	*amd*(y_i, t)	*div*(y_i, y)
y_1	0.240	0.027	0.461	0.028	0.056
y_2	0.387	0.120	0.607	0.174	0.293
y_3	0.176	0.091	0.375	0.058	0.149
y_4	0.175	0.093	0.236	0.197	0.290
y_5	0.359	0.092	0.486	0.053	0.145

TABLE III. Values about terms in *PExp*(x)

x_i	*dist*(x_i, x)	*lmd*(x_i, x)	*dist*(x_i, t)	*amd*(x_i, t)	*div*(x_i, x)
x_1	0.227	0.054	0.560	0.153	0.208
x_2	0.281	0.000	0.261	0.145	0.145
x_3	0.314	0.033	0.232	0.175	0.208
x_4	0.302	0.021	0.573	0.166	0.187

At the first iteration, the algorithm finds that *PExp*(t) = $\{x$, y, z, $w\}$. Consequently, it builds the following set of paths $\{$[t, x], [t, y], [t, z], [t, w]$\}$, and then orders these paths with respect to the Expansion Divergence of their tails.

At the second and last iteration, the algorithm has to select the most promising two current paths to expand, on the basis of the Expansion Divergence value of their tails, which are shown in Table I. According to such values, the paths [t, y] and [t, x] are further expanded. Then, the Possible Expansions of y and x, that are *PExp*(y) and *PExp*(x), respectively, are computed, and corresponding paths are built.

PExp(y) = $\{y_1$, y_2, y_3, y_4, $y_5\}$, but y_4 does not satisfy the Expansion Constraint, since *dist*(y_4, t) $<$ *dist*(y, t), as can be seen by considering Table I and Table II. At the same time, *PExp*(x) = $\{x_1$, x_2, x_3, $x_4\}$, but x_2 and x_3 do not satisfy the Expansion Constraint since *dist*(x_2, t) $<$ *dist*(x, t) and *dist*(x_3, t) $<$ *dist*(x, t), as can be seen by considering Table I and Table III.

Table IV summarizes by ordering terms in *PExp*(y) and *PExp*(x) with respect to their Expansion Divergence values, plus the information regarding the respect of the Expansion Constraint. According to that, the final Suggested Pahs are provided, such that *SPaths*(t, 2, 2) = $\{$[t, y, y_1], [t, y, y_5], [t, y, y_3], [t, x, x_4], [t, x, x_1], [t, y, y_2]$\}$.

IV. Implementation

In this Section, we introduce the prototypical implementation of the approach presented in the previous section for expanding search queries. For the realization of the algorithm, we adopted: (i) the open linguistic knowledge base Wordnik, as semantic network for computing the possible expansions of terms, and (ii) the Lin metrics [7], computed by exploiting the hyponym hierarchy of the WordNet lexical database, as linguistic similarity criteria. According to these choices, which rely on agreed and robust solutions for natural language processing, we have:

- the *Relatedness* relation *rel* is represented by the *Same Context* relation in the Wordnik thesaurus, which links terms that might be used in a similar context;
- the *Term Distance* is computed as the complement of the Lin similarity value, i.e.:
  $\mathit{dist}(x, y) = 1 - \mathit{sim}(x, y)$, where $0 \leq \mathit{sim}(x, y) \leq 1$ is computed according to the Lin similarity.

In Table V, we show some examples of expansion of the seed term "innovation", obtained according to the current implementation. Each row of the table reports the five best results computed by performing *#hop* iterations of the algorithm, hence achieving paths of different lengths. In the experiment, the constant max_{exp} has been fixed to 10, in order to bound the computational cost of the algorithm. The ranking is then computed according to the least *Expansion Divergence* (see Definition 7).

TABLE IV. Ordered *Expansion Divergence* values.

term$_i$	*div*(*term*$_i$, *term*)	*Laterality*
y_1	0.056	YES
x_2	0.145	NO
y_5	0.145	YES
y_3	0.149	YES
x_4	0.187	YES
x_1	0.208	YES
x_3	0.208	NO
y_4	0.290	NO
y_2	0.293	YES

TABLE V. Results of the expansion of the term 'innovation'

#hop	*result*
1	[innovation, investment]
	[innovation, achievement]
	[innovation, technology]
	[innovation, strategy]
	[innovation, research]
2	[innovation, research, management]
	[innovation, achievement, adventure]
	[innovation, research, education]
	[innovation, investment, risk]
	[innovation, strategy, enterprise]
3	[innovation, achievement, adventure, mystery]
	[innovation, achievement, enterprise, mission]
	[innovation, investment, risk, result]
	[innovation, research, investigation, observation]
	[innovation, achievement, enterprise, ambition]
4	[innovation, achievement, adventure, romance, poetry]
	[innovation, investment, risk, result, change]
	[innovation, research, management, manager, fund]
	[innovation, achievement, adventure, tale, legend]
	[innovation, strategy,enterprise, project, test]

A. LaSearch Architecture

The developed *LaSearch* tool (Lateral Search) is a Web-based application, developed in Java, whose main components are depicted in Figure 3. The client side has been designed as a mobile application, which interacts with a REST Web Server exposing (i) the query expansion procedure (Section III), implemented in the *Query Builder*, and (ii) methods for gathering data from external repositories or search engines, implemented in the *Data Source Library*.

The core component is the *Expander*, which implements the *suggested expansion* procedure, as defined in the algorithm in Figure 1, relying on: (i) the *Thesauri Manager*, which computes at each hop the *possible expansion* of the given terms according to the integrated thesauri, e.g., Wordnik is currently supported; (ii) the *Ranker*, which is responsible for assessing the *expansion divergence* of the terms at each hop; (iii) the *Similarity Reasoner*, for computing the linguistic *term distance*. The latter, in turn, depends on the NLP module, based on WordNet and the Semantic Measures Library (SML)[22], which provides the implementation of the Lin similarity metrics.

Finally, at the current status of the implementation we integrated in the *Data Source Library* three connectors for

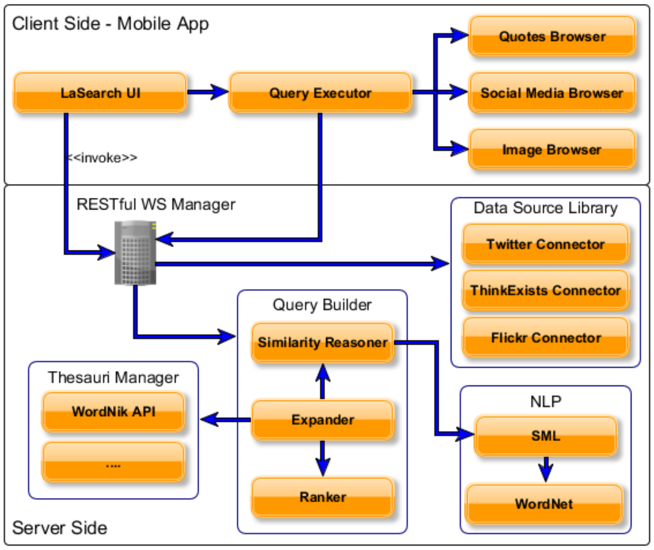


Figure 3. Architecture of the implemented prototype

extracting: (i) *hashtags* from Twitter, through the Rite-Tag API[23]; (ii) images from Flickr, through the exposed APIs[24]; (iii) quotations from ThinkExists, by using the boilerpipe library[25].

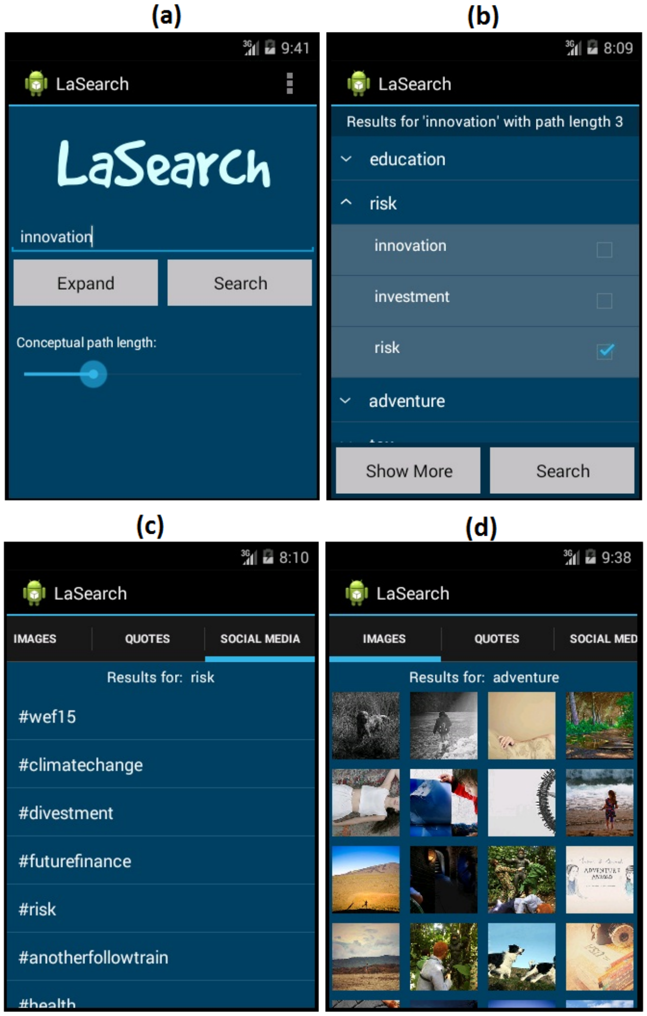
B. Mobile Application

The client side of the current prototype (upper part of Figure 3) is implemented as a mobile application, runnable on Android devices. It allows the user to input a term to be expanded, specifying the length of the *conceptual path*, i.e., the number of hops to be executed by the algorithm (Figure 4.a). The results of the expansion can be browsed (Figure 4.b), in order to select the terms to be used as search request by the *Query Executor*. The latter invokes the *Data Source Library*, which in turn retrieves data from a set of external repositories through the connectors implemented on the server side. In particular, we focused on three categories of data considered of particular interest in this context, namely, quotations, images, and user-generated contents on social media platforms.

The results collected by the *Query Executor* are then aggregated and presented to the user by specific browsers that allow the user to navigate the results. As an example, Figure 4.c shows popular Twitter *hashtags* related to the keyword *risk*, while Figure 4.d shows images retrieved on Flickr using the keyword *adventure*.

V. Conclusions and Future Works

In this paper, we presented a method for the expansion of search queries, based on the use of a semantics-enabled approach, tailored at enforcing lateral thinking mechanisms in order to achieve more creative, inspiring and evocative results. Our initial results have been implemented in a tool, *LaSearch*, that allows the users to expand a *seed* term, obtaining conceptual paths whose terms are reached by traversing pre-identified relationships in a given thesaurus and evaluating linguistic distance metrics. Query expansions are then used to gather results from external sources, such as images and quotation repositories, or social media.



innovation

Results for 'innovation' with path length 3

education

risk

innovation ☐

investment ☐

risk ☑

adventure

Results for: risk

#wef15

#climatechange

#divestment

#futurefinance

#risk

#anotherfollowtrain

#health

Results for: adventure

Figure 4. Screenshots of the user interface

The preliminary results presented in this paper open up several directions for future research. First of all, we plan to push forward the empirical evaluation of our proposal, to assess the quality of the results as perceived by human users. For what concerns the performance point of view, first experiments are encouraging and show that during the interactive usage of the tool, expansion and search tasks can be performed in an acceptable amount of time and memory resources.

Secondly, since the method is sensitive to the used semantic network and the relationship(s) in it (in the current implementation we used Wordnik as the semantic network, and ”same context” as the relationship), we plan to test additional knowledge sources (e.g., dbpedia [26], ConceptNet [27]), and relationships in them, as well as to consider, in the identification of terms in a path, criteria depending on the domain of corresponding concepts, in order to select terms from different contexts, which would be much related to lateral thinking. Furthermore, another initiative could be to integrate heterogeneous creativity-based approaches (e.g., analogies, metaphors [28]) for suggesting terms to the conceptual paths.

Finally, the most challenging initiative is go towards a formal definition of *lateral attitude*, introducing a computable metrics in order to be able to assess its increment at each iteration of the method. Many expansion results shown in Table V, such as [*innovation, research, education*], [*innovation, investment, risk*] or [*innovation, achievement, enterprise, ambition*], are significant examples of the kind of expansion we want to achieve. Indeed, while the terms in the paths incrementally diverge from the seed from a semantic perspective and, intuitively, increase their 'laterality', at the same time they maintain a figurative connection with it. However, translating this intuition into a structured method still remains an open issue. One could argue that precise calculation rules prevent creativity, while introducing random and chaotic behaviors could foster lateral thinking. So, is it finally sufficient to be crazy, free and nonjudgmental to be more creative [29]? We guess that it is not as simple as that.

REFERENCES

[1] E. De Bono, The Use of Lateral Thinking. Penguin, UK, Jan. 1971, ISBN: 978-0140214468.

[2] ”Serious Creativity,” URL: http://www.debonogroup.com/serious_creativity.php [accessed: 2015-03-12].

[3] E. M. Voorhees, “Query expansion using lexical-semantic relations,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval July 03-06, 1994, Dublin, Ireland.* Springer-Verlag New York, 1994, pp. 61–69, Bruce Croft, W., van Rijsbergen, C. J., Ed., ISBN: 0-387-19889-X.

[4] C. Carpineto and G. Romano, “A survey of automatic query expansion in information retrieval,” *ACM Computing Surveys*, vol. 44, pp. 1:1–1:50, 2012.

[5] T. Veale, “Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT) June, 2011, Portland, Oregon, USA.* Association for Computational Linguistics, Jun. 2011, pp. 278–287, ISBN: 978-1-932432-87-9.

[6] ”Wordnik,” URL: https://www.wordnik.com/ [accessed: 2015-03-12].

[7] D. Lin, “An information-theoretic definition of similarity,” in *Proceedings of the 15th International Conference on Machine Learning (ICML), 1998, Madison, Wisconsin, USA.* Morgan Kaufmann Publishers Inc., 1998, pp. 296–304, ISBN: 1-55860-556-8.

[8] ”Wordnet,” URL: http://wordnet.princeton.edu/ [accessed: 2015-03-12].

[9] ”LaSearch,” URL: http://leks.iasi.cnr.it/tools/lasearch [accessed: 2015-04-07].

[10] T. Veale, ”The Challenge of Creative Information Retrieval,” *Computational Linguistics and Intelligent Text Processing, Proceedings of the 5th International Conference (CICLing 2004), February 15–21, 2004, Seoul, Korea.* Springer-Verlag Berlin Heidelberg, 2004, pp. 457–467, Gelbukh, A., Ed., ISBN 3-540-21006-7, ISSN: 0302-9743.

[11] N. Seco, T. Veale, and J. Hayes, “Concept creation in lexical ontologies,” in *Proceedings of 4th Language Resources and Evaluation Conference (LREC) May 26–28, 2004, Lisbon, Portugal.* European Language Resources Association, 2004, pp. 1877–1880, Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., Silva, R., Ed., ISBN 2-9517408-1-6.

[12] G. A. Miller, “Wordnet: A lexical database for english,” *Communication of the ACM*, vol. 38, pp. 39–41, 1995, ISSN: 0001-0782.

[13] T. Veale, “The analogical thesaurus: An emerging application at the juncture of lexical metaphor and information retrieval,” in *Proceedings of the 15th International Conference on Innovative Applications of Artificial Intelligence (IAAI) August 12–14, 2003, Acapulco, Mexico.* AAAI Press, 2003.

[14] “A creative analogy machine: Results and challenges,” in *Proceedings of the 3rd International Conference on Computational Creativity May 30–June 1, 2012, Dublin, Ireland.* pp. 17–24, Maher, M. L., Hammond, K., Pease, A., Prez y Prez, R., Ventura, D., Wiggins, G., Ed., ISBN: 978-1-905254668.

[15] G. Lakoff and M. Johnson, *Metaphors we Live by*. University of Chicago Press, Chicago, 1980, ISBN: 9780226468013.

[16] T. Veale, “Once more, with feeling! using creative affective metaphors to express information needs,” in *Proceedings of the 4th International Conference on Computational Creativity June 12–14, 2013, Sydney, Australia*. pp. 16–23, Maher, M. L., Veale, T., Saunders, R., Bown, O., Ed., ISBN: 978-1-74210-317-4.

[17] A. Koestler, *The act of creation*. The Macmillan Company, 1964, ISBN: 0330731165.

[18] W. Dubitzky, T. Kötter, O. Schmidt, and M. R. Berthold, *Towards Creative Information Exploration Based on Koestler's Concept of Bisociation.* Springer Berlin Heidelberg, 2012, pp. 11–32, in Berthold, M. R., Bisociative Knowledge Discovery - An Introduction to Concept, Algorithms, Tools, and Applications - Part I, ISBN: 978-3-642-31829-0.

[19] ”Seenapse,” URL: https://www.seenapse.it/ [accessed: 2015-03-12].

[20] ”Yossarianlives,” URL: http://www.yossarianlives.com/ [accessed: 2015-03-12].

[21] Y. Landrin-Schweitzer, P. Collet, and E. Lutton, *Genetic Programming and Evolvable Machines*, vol. 7, pp. 9–31, 2006, ISSN: 1389-2576, March.

[22] ”Semantic Measures Library,” URL: http://www.semantic-measures-library.org/sml/ [accessed: 2015-03-12].

[23] ”Rite Tag,” URL: https://ritetag.com/ [accessed: 2015-03-12].

[24] ”Flickr,” URL: https://www.flickr.com/services/api/ [accessed: 2015-03-12].

[25] ”Boilerpipe,” URL: https://code.google.com/p/boilerpipe/ [accessed: 2015-03-12].

[26] ”DBpedia,” URL: http://dbpedia.org/ [accessed: 2015-03-12].

[27] ”ConceptNet5,” URL: http://conceptnet5.media.mit.edu/ [accessed: 2015-03-12].

[28] ”Metaphor Magnet,” URL: http://ngrams.ucd.ie/metaphor-magnet-acl/ [accessed: 2015-03-12].

[29] E. de Bono, “Exploring patterns of thought... serious creativity,” *Journal for Quality and Participation*, vol. 18, pp. 12–18, 1995.

SmartSocialMarket: A Social Commerce Architecture

Luca Salvatori* and Fausto Marcantoni*
*University of Camerino, 62032 Camerino, Italy
Email: {luca.salvatori,fausto.marcantoni}@unicam.it

***Abstract*—The evolution of the approach users have with the World Wide Web - particularly towards social media - has led to the need of e-commerce platforms aimed at user interaction. Offering a service focussing only on online shopping is no longer satisfactory. In order to provide a successful user-seller interaction, further Web 2.0 tools need to be offered. This development of electronic shopping - which provides new business opportunities - is called social commerce. This paper aims at describing the SmartSocialMarket architecture, which is based on e-commerce components and, at an upper level, on social components. Social components allow for the provision of tools that can improve the user interaction within the platform, and can also offer new market opportunities to sellers.**

***Keywords*–Social commerce, Collaborative commerce, Collaborative shopping, Social shopping**

I. Introduction

Social media - also called consumer-generated media - play a key role in Web 2.0. These internet applications can be virtual communities, blogs, wikis, photo and video sharing, networking sites, social bookmarking and other social applications [1]. In particular, the use of social networking websites has recently increased significantly. This tendency seems not to be going to stop any time soon, and in fact, it will improve over time [2]. On a daily basis, users communicate with their friends and share information on social networking platforms such as MySpace, Facebook and YouTube [3][4]. Practically, consumers turn into active content writers starting from being clients with no chance of interaction [5]. Such social activities lead to the creation of stable trust relationships among stakeholders, which, according to a wide number of researchers, could guarantee an increase of the economical power of business transactions [6][7]. This phenomenon, which can be considered relatively new, has been developing over time. Many researchers are focussing on it, as it is able to offer numerous research opportunities, both theoretical and practical.

The name “social commerce” comes from the combination of “social networking” and “e-commerce”. Users can carry out business transactions or simply browse for selling/purchasing opportunities within a highly collaborative and social environment. More in detail, users can generate contents such as reports, ratings, photos and videos to be shared with others [8][9]. Both sellers and purchasers can take advantage of this new business method. Sellers improve their income thanks to the advice and suggestions coming from purchasers, which help companies know their customers’ taste about brands and products of their interest [10]. At the same time, these opinions can attract new potential buyers. Through this shared information, customers are more aware of the quality and features of the products they like. This ensures they have more power within the whole business interaction.

The aim of this study is describing social commerce architecture.

This paper has the following structure: Section 2 introduces the concept of social commerce; Section 3 shows the architecture of social commerce and analyses the modules that build it; Section 4 shows the SmartSocialMarket in practice; Section 5 draws the conclusion of this work.

II. Brief view of the subject

Social commerce can be defined as internet-driven and social media-mediated business. This fosters the word of mouth through Web 2.0 and thus the exchange of information, opinions and ratings between purchasers and sellers.

This definition derives from studies made by many researchers who have provided different other definitions and insights on this phenomenon themselves.

Dennison defines social commerce as word of mouth applied to e-commerce [11].

A significant number of researchers shares a wider definition of social commerce. It is described as a social media-mediated form of trade where a more creative and collaborative approach is used by both sellers and purchasers [12][13]. In other words, social commerce is considered to be the execution of business transactions through social media and Web 2.0 in general [14]. Therefore, social commerce consists in the evolution of e-commerce and in the use of social media as tools to support purchasers during the different stages of the purchasing process.

Other professionals provide a definition of social commerce that comes from the concept of social network. These represent social commerce as the whole of collaboration and shopping activities carried out by customers in an environment which is similar to the social network one [15][16].

Finally, starting from the seller point of view, other researchers describe social commerce as a collaboration among those, which ensures they gain money out of it [17][18].

Despite the different definitions of social commerce provided, researchers agree on defining the main difference between social commerce and e-commerce. Whereas in e-commerce users mainly carry out individual activities that keep them isolated and not connected to the community, social commerce customers interact with it thanks to the use of the services and tools provided. For example, they can express their preferences and opinions, and also get to know the other community members’. As Huang and Benyoucef suggest, implementing such kind of commerce leads to the analysis of many disciplines, such as marketing, computer science, sociology and psychology [19]. With regards to marketing, social commerce represents the tendency to implement marketing tools during the online trading stages. These tools influence the customers’ purchasing decisions [20]. Technically speaking, social commerce is described as the implementation of tools as referral schemes, purchasing groups and social wish lists to support business activities [12][21].

About the social relevance of social commerce and its psychological component, the implementation of this technology encourages users to actively enter the Web community so that they are influenced by relevant contents shared by other users. These entail shop reliability and product quality [22][23].

III. SMARTSOCIALMARKET ARCHITECTURE

A. Architecture description

The SmartSocialMarket architecture is a modular type architecture (Figure 1), whose system is divided into two macro components: the e-commerce management and the social relationship management.

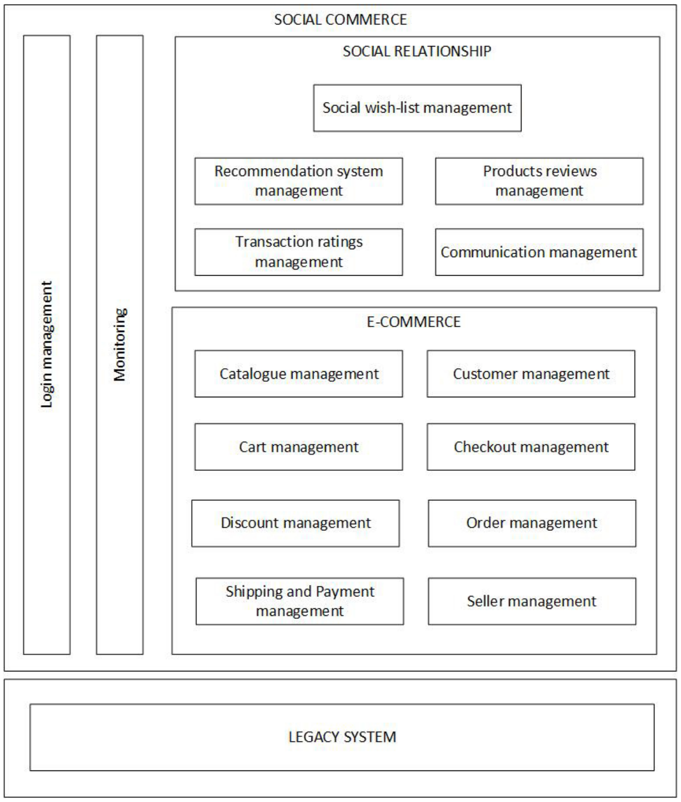


Figure 1. SmartSocialMarket Architecture

The first represents the e-commerce basic and usual components about the system itself, whereas the social management is the advanced part. This enables the management of the customer-customer and customer-seller social interaction.

This second component will be described thoroughly, as it represents an innovative architecture area. It entails all the management components of the social relations among e-commerce customers. Thanks to such functionalities, users can interface with other consumers so to create some sort of community. The exchange of information creates social presence in the community and trust relationships that convince users to buy more. This is because such purchases are believed to be safer in terms of shop and product trustworthiness. Moreover, the recommendation system component allows to provide users with advice tailored to their future purchases via dedicated algorithms. This advice takes into account the user's relationships with other consumers and their link to products. Beside these two macro components, other two core areas are represented by the monitoring and login management, which embrace both the business and social parts of the architecture.

At an inferior layer we find the legacy system, which consist of the already existing company infrastructure and information about products and customers, represents an inferior level of the structure.

More in detail, the two macro components present several parts. The e-commerce management component is composed by the catalogue, cart, discount, customer, seller, checkout, order, shipping and payment management. The social management component consists of social wish-list, recommendation system, transaction ratings, product reviews and communication management.

B. Analysis of the individual components

The most important stakeholders who interact with the platform are the customer and the seller.

The customer takes advantage of all the potential social and purchasing functions that the platform provides, whereas the seller represents an active user who is able to take advantage of the all potential dealer features. This last stakeholder is interested in the sale process in order to increase their profits and visibility; they also need to increase their customers knowledge in order to gather new products. Some architecture components regard the customer activities, others are related to the seller and others to both.

As concerns the e-commerce macro component, it embraces the catalogue, customer, seller, cart, checkout, order, discount and shipping and payment management.

The **catalogue management** component, which applies to both the customer and the seller, has to organise the activities related to the products and categories. It allows the seller to insert and update new categories in the system and relate the products to these features: price, title and description. This component also manages the product stock. With regards to the customer side, the catalogue management component allows them to view the product information and its features. This component is linked to the discount management component - because a discount might be applied to a product or to a category - and to order management components in order to provide the details of other products the customer ordered. Finally, it is worth mentioning that this component is synced with the legacy system so that the company and the social commerce data are aligned.

The **customer management** component has to manage the commerce accounts, the accounts group, the customer data and its shipping addresses. In particular, the commerce accounts consist of the customers e-mail, username and password; the accounts group consist of a list of accounts which have been created for a particular aim, as making the account easier to be found or managing discounts in a quicker way. The customer data are usually their name, last name, fiscal code, phone number and residence address. Finally, about the shipping addresses, this component has to allow the customer to insert, delete and update their address. Another task of this component consists in showing the customer information to the seller who, in this way, gains a total view about them.

The customer management component is linked to the checkout and order management components to which it provides the customer addresse and commerce account information. It also interfaces with the discount management component because a discount can be applied to a customer or to a group of them. Finally, it is connected with the login management component in order to transmit the customer login data.

Another important task is that it syncs the information with the legacy system which consist in the already existing company customer, in order to manage the customer both offline and online.

The **seller management** component organises the accounts of sellers who can work in the system using a seller profile. This component links with the login management component in order to transmit the seller login data.

As regards the **cart management** component, it allows the customer to add products in the basket, update the already ordered quantity and delete an item from the user cart. This component is linked to the catalogue management component. As a matter of fact, it has to know some product information too, for example, their stock level or their base price. It also interfaces with the discount component to calculate that specific product discounted price. Finally, this component has to give the cart information to the checkout component so that this knows what products have been ordered by the customer.

The **checkout management** component organises all the customer order configurations. Some tasks of this component regard the managing of the shipping and payment type, the customer address where the ordered products are to be shipped and the billing address. This component is linked to the customer management component, from which it is possible to retrieve the customer information, and the cart management component, in order to know what the products ordered by the customer are. In addition, it interfaces with the order management component in order to communicate when an order is confirmed. Finally, it connects with the payment and shipment management components to know the payment and shipment type and the price related to the order.

The **order management** component allows sellers to organise the already confirmed orders. Sellers can view and modify the details that concern the order status, the ordered products, the shipment and payment methods, the customer data and the address where to ship the items and the billing address as well. In order to obtain these details, this component has to interact with the catalogue, shipment and payment and customer management components. This component is also linked to the checkout management component because it needs to know when a customer confirms an order to save it in the order list. Finally, it connects to the transaction rating management component because the consumer can express a personal opinion and give a vote about their confirmed order.

The **discount management** component allows sellers to organise the discount that can be applied to a particular product or to a specific product category; on the other hand, it can be associated to a customer or to a set of them. The discount can be applied in percentage or as amount. This component is linked to catalogue management component, because a discount can be applied to a product or to a category, and it interfaces with the customer management component because a discount can be applied to a customer or to a group of them. Finally, it is important that the discount management component can sync its data with the legacy system data in order to guarantee the same discount both in offline and online shopping.

At the end of the process, the **shipping and payment management** component organises all data related to the shipping and payment method. The seller can set the shipping type and price that can vary according to the order destination country, weight, price or size. This component interfaces with the checkout management component to provide it with information on the payment and shipment type and price related to the order. It also interfaces with the order management component to provide the information concerning the order shipping and payment.

With regards to the social commerce macro component, it embraces the social wish-lists, the recommendation system, the product reviews, the transaction ratings, and the communication management.

About the **social wish-list management**, the aim of this component is to organise all customer and seller activities related to the wish-list in the social context. It allows the customer to manage particular products lists called wish-lists. More in detail, the customer can add or remove a wish-list where the desired products can be added or removed. Wish-lists can be used for the customer's own tracking of their desired products, but at the same time, they can be shared with other purchasers, being those also searchable and accessible to any user of the platform. Furthermore, sellers can view the customer social wish-lists and monitor their preferences. This component is linked to the catalogue management component, because it needs the product details in order to add them to the social wish-list, and to the recommendation system component in order to give it the desired products list to influence the creation of product recommendations.

The **recommendation system management** is composed by two activities: data collection and data extraction. The aim of the first activity is to collect the relationship between customer and customer, and between customer and item. Furthermore, the seller can decide what degree of priority to give to a product in order to influence the algorithm that generates the recommendation. Then, during the data extraction phase, this component selects some items in order to meet the customer tastes and suggests them to buy these products. In order to generate advice, the recommendation system uses sentiment analysis algorithms so that it can interpret comments left by consumers and guess if they are satisfied or not with a given product (polarity classification). The recommendation system management interfaces with the social wish-list, products review, chat and order management in order to extract the customer data related to their tastes and preferences. In addition, this component interfaces with the newsletter management component in order to send the product to be suggested.

About the **product review management component**, its aim is to allow customers to write reviews based on a 5-star scale and related to a particular product. In this way, user create a so called "usergenerated content". Another task of this component is to allow customers to read reviews written by other purchasers. The reviews represent an important tool because they are capable of exerting influence on other consumers in their purchasing decisions. This component is linked to the recommendation system component in order to give it information on the user preferences,which are useful to generate product suggestions. It also interfaces with catalogue and customer management components because it needs to know who the customer who created the review is and to what products the review is associated.

The **transaction rating** component allows customers to generate reviews and vote about a shopping transaction. In this way, customers can express their personal views about seller reliability, shipment velocity, payment safety and product quality. Customer reviews are useful to encourage other shoppers to buy on the social commerce platforms. It also allows sellers to inspect customer reviews in order to accept or reject them as purchasers. Moreover, it allows sellers to view the transaction ratings written by consumers in order know what the consumer satisfaction rate towards social commerce is. This component is linked to the order management component because when the customer inserts a personal opinion and votes, they have to relate it to a specific order.

The last social commerce management component is the **communication**. The aim of this component is facilitating information sharing among customers and sellers during the online purchasing activity, making it similar to a joint (offline) shopping trip to a shop or a shopping centre. In this way, customers can communicate regarding product profiles, ask suggestions, exchange opinions, compare experiences with others and find the desired products more efficiently.

In addition it organise all seller and customer activities related to sending marketing emails. Practically, it allows customers to sign in the mailing list and allows sellers to manually insert new customers in it. Thanks to this component sellers can also create a new marketing email and send it to all contacts in their mailing list. In particular, sellers can communicate with customers about new discounts, special sales or the arrival of new products. About chat management, this component links to the customer management component because the consumer needs to know what the other available clients are, in order to use the chat tool. As regards the newsletter, this component is linked to the catalogue management component because it allows sellers to automatically insert the product details and their related Web pages in the email. It also interfaces with the discount management component, in order to know what product are discounted, and with the recommendation management component in order to know what products the email alerts need to suggest.

As mentioned above, beside these two macro components, there are two independent elements: the login management component and the monitoring component.

The **login management** component allows users to authenticate into the system inserting their own username and password. After a successful authentication, they can access as “customer” or “seller” and perform only the operations that have been enabled for their relevant profile. If the authentication fails, the system informs the user with a warning message containing an error description, and allows them to insert their credentials again. This component links to customer and seller management components in order to receive the customer’s login details and the seller’s data. The authentication step is required for the customer to confirm the order, to view the confirmed order list, to use the chat, to submit a transaction rating and a product review and to manage the social wish-list, therefore this component is linked to the components that provide such services. Regarding the seller, the login is a fundamental step in order to perform almost all administrative tasks: managing other sellers, customers, catalogues, discounts, newsletters, shipping and payment methods, orders, chat, transaction ratings, recommendation system, social wish-list and performing monitoring tasks.

About the **monitoring** component, it represents the main component of the infrastructure that is related to the seller stakeholder. It allows the seller to monitor the customer preferences and needs. This activity makes the seller aware of their potential customers and potential needs, predicting the market trends and maintaining a valuable relationship with them. This component is essential for the trust infrastructure, as it ensure transparency and safety. Social commerce and social networking websites are frequently exposed to social engineering attacks. Therefore, activities such as disclosing confidential and sensitive information on publicly accessible or even internally shared workspaces may end up in security breaches and invasion of privacy. This component links to almost all other social commerce and e-commerce components, because it is used to supervise the social commerce activities.

IV. SmartSocialMarket in practice

The graph in Figure 2 represents the activity diagrams explaining the above described architecture. The grey background boxes show the social activities, while the white background ones show activities that are linked to ordinary business operations.

Following the flow of operations conducted by a said user within social commerce is made possible by referring to this graph. Such social commerce has been implemented according to the architectural features of SmartSocialMarket.

At the beginning of their operations, users can view comments and ratings given by other customers who have already purchased goods through social commerce. Understanding how influential the meaning of this information can be on the path customers are walking is a key element. This function is carried out by the transaction rating management component of the architecture. Users can choose to view a category of products managed by the catalogue management component or a list of products supplied by the recommendation system management component without the need to check comments and ratings.

Practically speaking, consumers can browse the social commerce freely, viewing the specific product categories they have selected. Alternatively, they can opt for a guided browsing, resulting from the tips the system provides about the products that are considered to be most appropriate for that user. The user can subsequently view all the features of a product, such as title, price and description. The management is carried out by the catalogue management component.

The client can perform three different operations during this phase.

The first one consists of viewing the comments and ratings other users have left about a given product, so that it is possible to evaluate any potential purchase at best. This activity is managed by the product review management component.

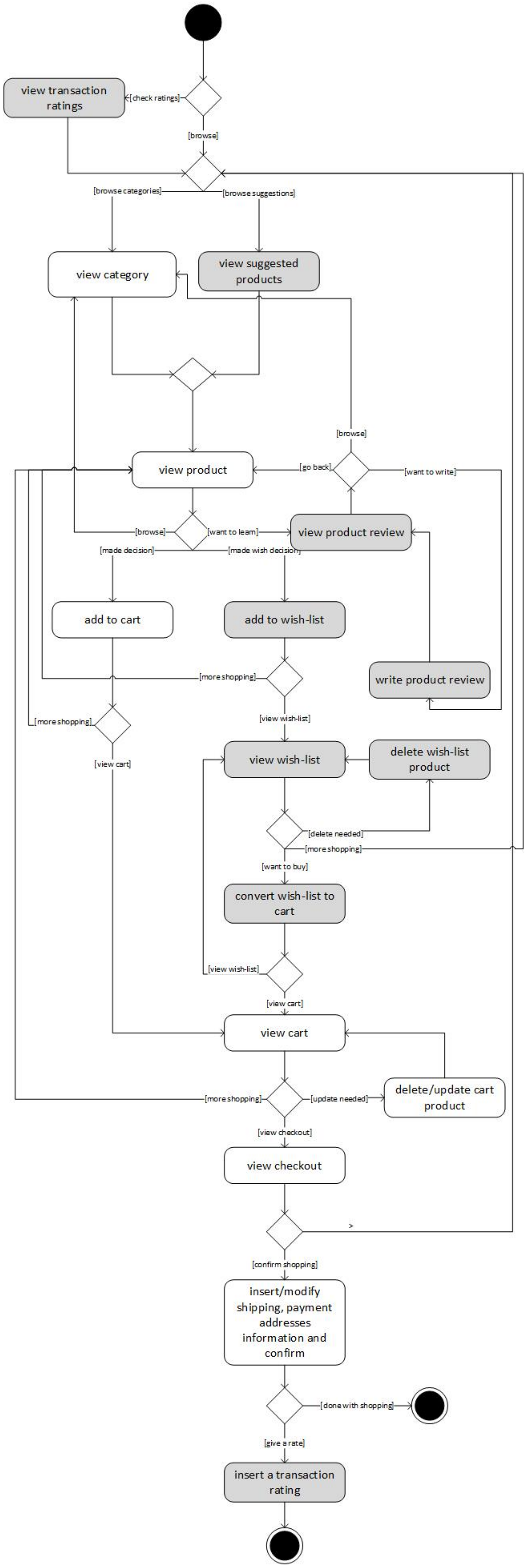


Figure 2. SmartSocialMarket Activity Diagram

During this phase the user can write their own report on the product and go back to view the others', or go back to viewing the product or the product category without leaving any comment or vote.

The second operation a user can carry out is buying a product following a simple and immediate method: adding the product to their cart and staying in the product view page. This activity is managed by the cart management component. In this way they can keep on browsing the social commerce and make further purchases or view their cart (this function is also performed by the cart management component).

The third operation consists in adding the product to their personal wish-list, thanks to the support of the social wish-list management component. Once this operation has been completed, the user can go back to viewing other products so that they can make more purchases or gain information about their wish-list and the products that have been added to it. When viewing the list, users can cancel orders that they have previously placed, viewing details of any products on the wish list or make their wish list turn into a real cart. In this way, clients can buy products in which they have been particularly interested over time and which, therefore, have been added to the wish-list. Once such "conversion" has been done, users can go back to viewing their wish list or their cart. While checking the cart, they can edit the quantity of products they want to order or go back to view them again, so that they can make further purchases. Users can view checkout information, which is managed by the checkout management component, and go back to viewing wish lists or input and edit their purchase data as shipping and billing addresses, and shipping and payment modes. Finally, they can confirm their order.

If they wish, users can rate and comment the transaction they have just completed, so that they can advise other users before their purchases. This is a key activity, as the comments users have made about their transaction can be read by another potential buyer and can condition their purchasing decision about that product.

V. Technology

Table I summarizes all the comparison factors among the major e-commerce platforms available for the CMS Umbraco, implementing an e-commerce managing system, without providing any social service. However, these platforms can be considered as a basis for possible social commerce developments.

The Umbraco platform has been chosen thanks to its features and potential. This CMS has proven to provide more ease of development when compared to other CMSs as Joomla! and Wordpress when it has been examined carefully and after practical applications. It also provides the platform administrator with an excellent management support. The advantages of the use of Umbraco are related to the possibility of extending its administration panel widely and easily. This can be done by using Umbraco's own structures or by writing specific back-end modules. Other advantages are related to the possibility of immediate integration of portions of codes written and defined by the developer (the so called "macros").

To achieve our purpose, the elements regarding the development of a social platform have been taken into account. Specifically, for each platform it has been verified if they use

TABLE I. MAIN E-COMMERCE PLATFORM FOR UMBRACO

	uCommerce	**uWebShop**	**Tea Commerce**	**Merchello**
Use of external database tables	YES	NO	NO	YES
Easy management of many products	YES	NO	NO	YES
Customization level	HIGH	LOW	MEDIUM	HIGH
Stability level on Umbraco 7	YES	NO	YES	NO
Open Source	NO	NO	NO	YES
Price	Free with restriction	from 299	1299 Ex. VAT	Free

Umbraco database tables or external ones. We also verified if the management of several products might create problems about the platform performance and usability for the user. Finally, we focused on the level of platform customization, its stability degree in relation to the latest version of Umbraco, price and license (open source or not).

Between the analyzed e-commerce platforms, Merchello represents a good choice to develop social commerce on Umbraco 7. Although this platform presents low stability, because of its youth, Merchello possesses the highest improvement level.

Alternatively, the uCommerce platform can be considered more complete and more stable than Merchello. However, uCommerce is not open source so it could not result suitable for an extended e-commerce implementation.

VI. Conclusion

The social commerce phenomenon has attracted researchers and business managers' attention more and more. Its implementation - which implies researchers have to steadily face challenges - is nowadays a warranty of business opportunities for sellers and of innovative and more stimulating purchasing modes for consumers.

This paper aims at showing the SmartSocialMarket architecture of a social commerce, describing it both in its complexity and in its single components. In particular, a detailed analysis of the single activities each module carries out has been introduced, together with an explanation about how each of those interfaces with the others. Starting from this description, it was possible to show how this architecture can be executed in practice.

An Activity Diagram has been built and it includes all the activities users can complete in order to place an order in a potential social commerce that is implemented as this architecture describes. The activities that describe the social interaction among consumers have been shown in such diagram, so that they can highlight the added value of a social commerce as opposed to an e-commerce.

This implementation provides many services that are able to allow a strong social interaction among consumers and among these and sellers. An example of this is the possibility to leave comments and ratings about products and completed transactions.

Implementing social commerce is also beneficial to sellers. A deeper knowledge of the consumer tastes and interests guarantees the chance to foresee the market demand and gain monetary growth.

Finally, this article has provided a description of the activity diagram flow, showing how a generic user can create a purchase process within social commerce. Thinking about the future, the aim is to apply this architecture to a real social commerce, so that it is possible to compare the actual benefits that this implementation causes. Consequently, it will be possible to focus on the behaviour of those components that, for their wide variables and implementations challenges, represent a harder research effort. Such research will need to be applied to the solving algorithms.

References

[1] K.-Y. Kwahk and X. Ge, "The effects of social media on e-commerce: A perspective of social impact theory," in System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE, 2012, pp. 1814–1823.

[2] K. K. K. Hari and S. P. Rajan, "A clubbing of e-commerce and social networking sites," in Ubiquitous Computing and Multimedia Applications (UCMA), 2011 International Conference on. IEEE, 2011, pp. 8–9.

[3] G. Swamynathan, C. Wilson, B. Boe, K. Almeroth, and B. Y. Zhao, "Do social networks improve e-commerce?: a study on social marketplaces," in Proceedings of the first workshop on Online social networks. ACM, 2008, pp. 1–6.

[4] S. Li and E. Karahanna, "Peer-based recommendations in online b2c e-commerce: comparing collaborative personalization and social network-based personalization," in System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE, 2012, pp. 733–742.

[5] H. Ickler, S. Schülke, S. Wilfling, and U. Baumöl, "New challenges in e-commerce: How social commerce influences the customer process," in Proceedings of the 5th National Conference on Computing and Information Technology, NCCIT, 2009, pp. 51–57.

[6] S. Kim and H. Park, "Effects of various characteristics of social commerce (s-commerce) on consumers trust and trust performance," International Journal of Information Management, vol. 33, no. 2, 2013, pp. 318–332.

[7] C. S.-P. Ng, "Intention to purchase on social commerce websites across cultures: A cross-regional study," Information & Management, vol. 50, no. 8, 2013, pp. 609–620.

[8] T. Koivumäki, R. Svento, J. Perttunen, and H. Oinas-Kukkonen, "Consumer choice behavior and electronic shopping systems–a theoretical note," Netnomics, vol. 4, no. 2, 2002, pp. 131–144.

[9] B. Choi, C. Lee, H. Lee, and M. Subramani, "Effects of web retail service quality and product categories on consumer behavior: a research model and empirical exploration," in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on. IEEE, 2004, pp. 10–pp.

[10] E. Turban, N. Bolloju, and T.-P. Liang, "Social commerce: An e-commerce perspective," in Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business. ACM, 2010, pp. 33–42.

[11] G. Dennison, S. Bourdage-Braun, and M. Chetuparambil, "Social commerce defined," White paper, vol. 23747, 2009.

[12] C. Wang and P. Zhang, "The evolution of social commerce: The people, management, technology, and information dimensions," Communications of the Association for Information Systems, vol. 31, no. 5, 2012, pp. 1–23.

[13] S. Parise and P. J. Guinan, "Marketing using web 2.0," in Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. IEEE, 2008, pp. 281–281.

[14] R. Gonçalves Curty and P. Zhang, "Website features that gave rise to social commerce: a historical analysis," Electronic Commerce Research and Applications, vol. 12, no. 4, 2013, pp. 260–279.

[15] Y. R. Kang and C. Park, “Acceptance factors of social shopping,” in Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, vol. 3. IEEE, 2009, pp. 2155–2159.

[16] X. Chen and J. Tao, “The impact of users’ participation on ewom on social commerce sites: An empirical analysis based on meilishuo. com,” in Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference on. IEEE, 2012, pp. 810–815.

[17] A. T. Stephen and O. Toubia, “Explaining the power-law degree distribution in a social commerce network,” Social Networks, vol. 31, no. 4, 2009, pp. 262–270.

[18] D. Kim, “Process chain: A new paradigm of collaborative commerce and synchronized supply chain,” Business Horizons, vol. 49, no. 5, 2006, pp. 359–367.

[19] Z. Huang and M. Benyoucef, “From e-commerce to social commerce: A close look at design features,” Electronic Commerce Research and Applications, vol. 12, no. 4, 2013, pp. 246–259.

[20] E. Constantinides and S. J. Fountain, “Web 2.0: Conceptual foundations and marketing issues,” Journal of Direct, Data and Digital Marketing Practice, vol. 9, no. 3, 2008, pp. 231–244.

[21] R. G. Curty and P. Zhang, “Social commerce: Looking back and forward,” Proceedings of the American Society for Information Science and Technology, vol. 48, no. 1, 2011, pp. 1–10.

[22] W. Suraworachet, S. Premsiri, and N. Cooharojananone, “The study on the effect of facebook’s social network features toward intention to buy on f-commerce in thailand,” in Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on. IEEE, 2012, pp. 245–250.

[23] Z. Gan, J. He, Q. Ding, and V. Varadharajan, “Trust relationship modelling in e-commerce-based social network,” in Computational Intelligence and Security, 2009. CIS’09. International Conference on, vol. 1. IEEE, 2009, pp. 206–210.