



WEB 2013

The First International Conference on Building and Exploring Web Based
Environments

ISBN: 978-1-61208-248-6

January 27 - February 1, 2013

Seville, Spain

WEB 2013 Editors

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Petre Dini, Concordia University, Canada / China Space Agency Center, China

WEB 2013

Foreword

The First International Conference on Building and Exploring Web Based Environments [WEB 2013], held between January 27th- February 1st, 2013 in Seville, Spain, was the inaugural conference on web-related theoretical and practical aspects, focusing on identifying challenges for building web-based useful services and applications, and for 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.

We take here the opportunity to warmly thank all the members of the WEB 2013 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 who dedicated much of their time and efforts to contribute to WEB 2013. 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 are grateful to the members of the WEB 2013 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that WEB 2013 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Web-based environments.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the charm of Seville, Spain.

WEB Chairs:

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

WEB 2013

Committee

WEB Advisory Chairs

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

WEB 2013 Technical Program Committee

Rajendra Akerkar, Western Norway Research Institute, Norway

Patrick Albert, IBM CAS France - Paris, France

Remi Arnaud, AMD, France

Sofia Athenikos, Amazon, Greece

Karim Baïna, ENSIAS / Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes - Rabat, Morocco

Henri E. Bal, Vrije Universiteit, The Netherland

Khalid Belhajjame, University of Manchester, UK

Carlos Bobed, University of Zaragoza, Spain

Coral Calero Muñoz, University of Castilla-La Mancha, Spain

Jacques Calmet, Karlsruhe Institute of Technology (KIT), Germany

Delroy Cameron, Wright State University, USA

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, University of São Paulo, Brazil

Joseph Corneli, KMI, Open University / PlanetMath, UK

Juri Luca De Coi, Université Jean Monnet - Saint-Etienne, France

Cláudio de Souza Baptista, University of Campina Grande, Brazil

Siamak Faridani, Berkeley University, USA

Kenneth K. Fletcher, Missouri University of Science and Technology - Rolla, USA

Raffaella Folgieri, Università degli Studi di Milano, Italy

Olivier Gendreau, Polytechnique Montréal, Canada

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

Tudor Groza, The University of Queensland, Australia

João Guerreiro, INESC-ID - Lisbon and Technical University of Lisbon / Instituto Superior Técnico - Lisbon, Portugal

Allel Hadjali, IRISA/ENSSAT, Université Rennes 1 - Lannion, France

Tzung-Pei Hong 洪宗貝, National University of Kaohsiung, Taiwan

Benoit Hudzia, SAP Research, UK

Mirjana Ivanovic, University of Novi Sad, Serbia

Rajaraman Kanagasabai, Institute for Infocomm Research, Singapore

Roula Karam, Politecnico di Milano, 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
Ivan Koychev, University of Sofia, Bulgaria
Wolfram Laaser, WWEDU World Wide Education GmbH, Germany
Steffen Lamparter, Siemens AG - München, Deutschland
Sergio Luján Mora, University of Alicante, Spain
Xiangfeng Luo, Key Lab of Grid Technology - Shanghai, China
Maristella Matera, Politecnico di Milano, Italy
Abdul-Rahman Mawlood-Yunis, Canada Revenue Agency - Ottawa, Canada
Marios Meimaris, National Technical University of Athens, Greece
Michele Melchiori, Università degli Studi di Brescia, Italy
Christoph Lange, University of Birmingham, UK
Grzegorz J. Nalepa, AGH University of Science and Technology, Poland
Viorel Negru, West University of Timisoara, Romania
Bo Ning, Dalian Maritime University, China
Talal Noor, The University of Adelaide, Australia
Tope Omitola, University of Southampton, UK
Mourad Ouziri, Université Sorbonne Paris Cité, France
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
Laura Po, Università di Modena e Reggio Emilia, Italy
Andre Ponce de Leon F. de Carvalho, University of Sao Paulo at Sao Carlos, Brazil
Hemant Purohit, Wright State University, USA
Isidro Ramos, Polytechnic University of Valencia, Spain
Tarmo Robal, Tallinn University of Technology, Estonia
Christophe Roche, Université de Savoie - Le Bourget du Lac, France
Gustavo Rossi, UNLP, Argentina
Soror Sahri, LIPADE / Université Sorbonne Paris Cité, France
Carmen Santoro, ISTI-CNR-Pisa, Italy
Monica Scannapieco, National Institute for Statistics - Rome, Italy
Clemens Schefels, Goethe University Frankfurt, Germany
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
Yehia Taher, Tilburg University, The Netherlands
Peter Thiessen, eBudd B.V. - Amsterdam, The Netherlands
Thanassis Tiropanis, University of Southampton, UK
Genoveffa Tortora, Università degli Studi di Salerno - Fisciano, Italy
Michalis Vafopoulos, University of the Aegean - Mytilini, Greece

Costas Vassilakis, University of Peloponnese, Greece
Krzysztof Walczak, Poznan University of Economics, Poland
Zhe Wu, Oracle, USA
Lai Xu, Bournemouth University - Dorset, UK
Sule Yildirim Yayilgan, Gjøvik University College, Norway
Fouad Zablith, American University of Beirut, Lebanon
Jie Zhang, Nanyang Technological University, Singapore

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 to 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 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

WebODRA - A Web Framework for the Object-Oriented DBMS ODRA <i>Mariusz Trzaska</i>	1
Contributing to Mathematics Lessons Authoring System (MLAS) A Web-based Application Programming Interface <i>Samer F. Khasawneh and Abdulelah A. Algozaibi</i>	7
Restraining Technical Debt when Developing Large-Scale Ajax Applications <i>Yoav Rubin, Shmuel Kallner, Nili Guy, and Gal Shachor</i>	13
Distributed OSGi Service Platform through Apache CXF and Web Services <i>Irina Astrova and Arne Koschel</i>	19
Towards Online Engagement via the Social Web <i>Ioannis Stavrakantonakis, Andreea-Elena Gagi, Ioan Toma, and Dieter Fensel</i>	26
Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web <i>Wondu Yalew Mallede, Farhi Marir, and Vassil Vassilev</i>	32
A Framework for the Coordination of the Invocations of Web Services <i>Mohammed Alodib</i>	42
Effects of Web Technologies on Tourism Industry in Some Southern European Countries <i>Sara Ficarelli, Sandra Sendra, Laura Ferrando, and Jaime Lloret</i>	49
Algorithm for Automatic Web API Composition <i>Yong-Ju Lee</i>	57
Describing Semantics of 3D Web Content with RDFa <i>Jakub Flotynski and Krzysztof Walczak</i>	63
The Rise of the Web for Agents <i>Ruben Verborgh, Erik Mannens, and Rik Van de Walle</i>	69
Semantic Annotation of unstructured Wiki Knowledge according to Ontological Models <i>Roberto Boselli, Mirko Cesarini, Fabio Mercurio, and Mario Mezzanzanica</i>	75
Translating Natural Language Competency Questions into SPARQLQueries: A Case Study <i>Leila Zemmouchi-Ghomari and Abdessamed Reda Ghomari</i>	81

Extending Web Modeling Language to Exploit Stigmergy: Intentionally Recording Unintentional Trails <i>Aiden Dipple, Kerry Raymond, and Michael Docherty</i>	87
eRaUI: An Adaptive Web Interface for e-Research Tools <i>Farhi Marir, Sahithi Siva, and Yanguo Jing</i>	93
Using Semantic Indexing to Improve Searching Performance in Web Archives <i>Arshad Khan, David Martin, and Thanassis Tiropanis</i>	101
User Profiles in Information Web Portals <i>Carmen Moraga, Maria Angeles Moraga, Angelica Caro, Coral Calero, and Rodrigo Romo Munoz</i>	105
A Graph Model of Events Focusing on Granularity and Relations Towards Organization of Collective Intelligence on History <i>Minoru Naito, Yasuhito Asano, and Masatoshi Yoshikawa</i>	111

WebODRA - A Web Framework for the Object-Oriented DBMS ODRA

Mariusz Trzaska

Chair of Software Engineering
Polish-Japanese Institute of Information Technology
Warsaw, Poland
mtrzaska@pjwstk.edu.pl

Abstract—The modern Web requires new ways for creating applications. We present our approach combining a web framework with a modern object-oriented database. It makes it easier to develop web applications by rising the level of abstraction. In contrast to many existing solutions, where the business logic is developed in an object-oriented programming language and data is stored and processed in a relational system, our proposal employs a single programming and query language. Such a solution, together with flexible routing rules, creates a coherent ecosystem and, as an additional benefit, reduces the impedance mismatch. Our research is supported by a working prototype of the web framework for ODRA, a powerful object-oriented database management system. Furthermore, a simple web application (a forum) has been created to prove usefulness of the approach and the framework.

Keywords-Web frameworks; Web tools; Web applications; Object-Oriented Databases.

I. INTRODUCTION

Modern web applications are usually developed using the three-tier architecture: a presentation layer, business logic (a middle tier) and a data tier. Each of them can be developed through a different technology and can utilize incompatible data models.

Typically, the middle tier is developed using an object-oriented programming language like Java, MS C#, Ruby, etc. However, the object-orientedness is a bit blurry. There is no single, well-accepted, specific definition or set of properties which determines features of an object-oriented programming language. Java and C# are pretty close to each other in that area, but for instance Ruby is based on different concepts, in particular, duck typing [1].

Contrary to implementation of the business logic, the data is usually stored using a relational database system. This causes a negative phenomenon known as impedance mismatch. During the years, numerous approaches have been formulated to solve or minimize the problem. Following Trzaska [2], the solution could use a single model both for the business logic and for the data. In this paper, we would like to employ the idea for a tool aiming at creating web applications. We propose a paradigm which uses the same high level language for working with data and implementing a business logic. In fact, those two utilizations are indistinguishable.

On the software level, our tool is implemented as a prototype system, called WebODRA, which integrates two independent components:

- The object-oriented DBMS ODRA with SBQL, a powerful programming and query language,
- A web server.

This approach increases significantly the level of abstraction, which reduces implementation time, decreases the number of errors and of course, completely eliminates the impedance mismatch. The programmers are able to focus on website's creation using a single, coherent technology.

The main contribution of the paper are the following:

- A new coherent paradigm of creating web application using the same high level programming and query language;
- A working prototype implementation of the approach containing object-oriented database, web server and all the necessary components.

The rest of the paper is organized as follows. To fully understand our motivation and approach, some related solutions are presented in Section 2. Section 3 briefly discusses key concepts of the utilized database and programming/query language. Section 4 presents the prototype implementation of the proposed web framework. Section 5 is devoted to a sample utilization of the prototype. Section 6 concludes.

II. RELATED SOLUTIONS

There are a lot of different web frameworks using many approaches. Just to name the most popular ones (by platform):

- Java: Apache Struts, Java Server Faces, JBoss Seam, Spring, Grails (Groovy);
- MS C#: ASP.Net, ASP.NET MVC, Kentico;
- PHP: CakePHP, Symfony, Zend;
- Smalltalk: Seaside [3];
- Ruby: Ruby on Rails, Sinatra.

They differ in some details but unfortunately share the same problems related to inconsistent models for programming languages and data. Even when an object-relational mapper is utilized the problems decrease not vanish. For instance, the Ruby's Active Record requires some additional information from a programmer to specify some non-mappable objects like arrays [4].

However, it is also possible to find solutions, where a website is developed using a single model. The next paragraphs contain description of such frameworks.

CouchApp [5] is a technology which allows to create applications delivered to the browser from CouchDB [6]. The applications are implemented using JavaScript and HTML5. The general idea is quite similar to our approach because CouchDB is a database management system. However, on contrary to our framework, the DBMS follows the NoSQL philosophy and allows to store documents in the JSON [7] format. There is also no query language similar to SQL or our SBQL (see section 3). All database queries are performed using dedicated API and JavaScript. The result is also returned as a JSON data.

Of course, every web application, needs a GUI. In case of CouchApp a GUI is created as a transformation of returned JSON data into some other format. For instance there are functions, which together with dedicated views, are able to convert the data into HTML, XML, CVS, etc.

Another approach to create a web application might employ the Model Driven Architecture (MDA) paradigm. The idea is to define a model (or models) and, through some transformations, receive a working application. There is a lot of such systems [8, 9, 10]. However, they are not widely utilized. One of the reason could be the amount and type of work which has to be done to get a working website. For instance [10], which is quite common for all MDA solutions, needs the following models and information to be precisely defined:

- UWA requirements,
- Information model,
- Navigation model,
- Transaction & operation model,
- Publishing model,
- Customization model,
- Logical models (UML diagrams): class, sequence.

Of course, the above information is not only required by MDA tools. Furthermore, they have to be provided by all websites' developers. However, it seems that the way of defining them, makes the difference in popularity.

The last described solution is not exactly a framework for programmers. Oracle Application Express [11] is more like a tool for a rapid web application development for the Oracle database. It is available, under different names, since 2000. The application requires a dedicated server and provides easy-to-use programming environment accessible via a web browser.

Most of its functionalities is available via dedicated graphical user interfaces, various wizards and helpers. But, still there are possibilities for using a programming language, namely PL/SQL. SQL, despite of thirty-years existence, and big popularity is the subject of heavy criticism. The SQL's flaws like: inconsistencies, incompatibilities between vendors and shortcomings of the relational model, decrease a value of the solution. Furthermore, application generators have some inherent shortcomings which make them less flexible (in terms of usability, functionality, GUI) than application developed by programmers. We believe that using a more powerful programming and query language together with an object-oriented model can formulate a much better approach.

III. THE ODRA DATABASE

As mentioned previously, our proposal for creating websites is based on utilization an object-oriented database together with a powerful query and programming language. DBMS could be used as a source for data and could be utilized to implement a business logic. For the purpose of the first requirement we need a database query language. However, because of the second necessity, we might need something more flexible and powerful: a fully-fledged programming language with imperative constructs. Both criteria are met by our prototype DBMS called ODRA.

ODRA (Object Database for Rapid Application development) is a prototype object-oriented database management system [12, 13, 14, 15] based on SBA (Stack-Based Architecture) [16]. The ODRA project started to develop new paradigms of database application development. This goal is going to be reached mainly by increasing the level of abstraction at which the programmer works. ODRA introduces a new universal declarative query and programming language SBQL (Stack-Based Query Language) [12], together with distributed, database-oriented and object-oriented execution environment. Such an approach provides functionality common to the variety of popular technologies (such as relational/object databases, several types of middleware, general purpose programming languages and their execution environments) in a single universal, easy to learn, interoperable and effective to use application programming environment.

ODRA consists of three closely integrated components:

- Object Database Management System (ODMS),
- Compiler and interpreter for object-oriented query programming language SBQL,
- Middleware with distributed communication facilities based on the distributed databases technologies.

The system is additionally equipped with a set of tools for integrating heterogeneous legacy data sources. The continuously extended toolset includes importers (filters) and/or wrappers to XML, RDF, relational data, web services, etc.

ODRA has all chances to achieve high availability and high scalability because it is a main memory database system with memory mapping files and makes no limitations concerning the number of servers working in parallel. In ODRA many advanced optimization methods that improve the overall performance without compromising universality and genericity of programming interfaces have been implemented.

The next subsections contain a short discussion of the ODRA main features including its query and programming language SBQL.

A. ODRA Object-Oriented Data Model

The ODRA data model is similar to the UML object model. Because in general UML is designed for modeling rather than for programming several changes have been made to the UML object model that do not undermine seamless transition from a UML class diagram to an ODRA

database schema. The ODRAs object model covers also the relational model as a particular case; this feature is essential for making wrappers to external sources stored in relational databases. Below, we present a short description of the main data model elements:

- **Objects.** The basic concept of the ODRAs database model is object. It is an encapsulated data structure storing some consistent bulk of information that can be manipulated as a whole. A database designer and programmers can create database and programming objects according to their own needs and concepts. Objects can be organized as hierarchical data structures, with attributes, sub-attributes, etc.; the number of object hierarchy levels is unlimited. Any component of an object is considered an object too.
- **Collections.** Objects within a collection have the same name; the name is the only indicator that they belong to the same collection. Usually objects from a collection have the same type, but this requirement is relaxed for some kinds of heterogeneous collections. Collections can be nested within objects with no limits (e.g., in this way it is possible to represent repeating attributes).
- **Links.** Objects can be connected by pointer links. Pointer links represent the notion that is known from UML as association. Pointer links support only binary associations; associations with higher arity and/or with association classes are to be represented as objects and some set of binary associations. This is a minor limitation in comparison to UML class diagrams, introduced to simplify the programming interface. Pointer links can be organized into bidirectional pointers enabling navigation in both directions.
- **Modules.** In ODRAs the basic unit of database organization is a module. As in popular object-oriented languages, a module is a separate system component. An ODRAs module groups a set of database objects and compiled programs and can be a base for reuse and separation of programmers' workspaces. From the technical point of view and of the assumed object relativism principle, modules can be perceived as special purpose complex objects that store data and metadata.
- **Types, classes and schemata.** A class is a programming abstraction that stores invariant properties of objects, in particular, its type, some behavior (methods, operations) and (optionally) an object name. A class has some number of member objects. During processing of a member object the programmer can use all properties stored within its class. The model introduces atomic types (integer, real, string, date, boolean) that are known from other programming languages. Further atomic types are considered. The programmer can also define his/her own complex types. Collection types are specified by cardinality numbers, for instance, [0..*], [1..*], [0..1], etc.

- **Inheritance and polymorphism.** As in the UML object model, classes inherit properties of their superclasses. Multiple inheritance is allowed, but name conflicts are not automatically resolved. The methods from a class hierarchy can be overridden. An abstract method can be instantiated differently in different specialized classes (due to late binding); this feature is known as polymorphism.
- **Persistence and object-oriented principles.** The model follows the orthogonal persistence principle, i.e. a member of any class can be persistent or volatile. Shared server objects are considered persistent, however, non-shared objects of a particular applications can be persistent too. The model follows the classical compositionality, substitutability and open-close principles assumed by majority of object-oriented programming languages.

Distinction between proper data and metadata (ontology) is not the property of the ODRAs database model. The distinction can be important on the business model level, but from the point of view of ODRAs both kinds of resources are treated uniformly.

B. Query and Programming Language SBQL

SBQL (Stack-Based Query Language) is a powerful query and programming language addressing the object model described above. SBQL is precise with respect to the specification of semantics. SBQL has also been carefully designed from the pragmatic (practical) point of view. The pragmatic quality of SBQL is achieved by orthogonality of introduced data/object constructors, orthogonality of all the language constructs, object relativism, orthogonal persistence, typing safety, introducing all the classical and some new programming abstractions (procedures, functions, modules, types, classes, methods, views, etc.) and following commonly accepted programming languages' and software engineering principles.

SBQL queries can be embedded within statements that can change the database or program state. We follow the state-of-the-art known from majority of programming languages. Typical imperative constructs are creating a new object, deleting an object, assigning new value to an object (updating) and inserting an object into another object. We also introduce typical control and loop statements such as if...then...else..., while loops, for and for each iterators, and others. Some peculiarities are implied by queries that may return collections; thus there are possibilities to generalize imperative constructs according to this new feature.

SBQL in ODRAs project introduces also procedures, functions and methods. All procedural abstractions of SBQL can be invoked from any procedural abstractions with no limitations and can be recursive. SBQL programming abstractions deal with parameters being any queries; thus, corresponding parameter passing methods are generalized to take collections into account.

SBQL is a strongly typed language. Each database and program entity has to be associated with a type. However, types do not constraint semi-structured nature of the data. In

particular, types allow for optional elements (similar to null values known from relational systems, but with different semantics) and collections with arbitrary cardinality constraints. Strong typing of SBQL is a prerequisite for developing powerful query optimization methods based on query rewriting and on indices.

C. Virtual Updatable Views

Another interesting and quite unique ODRA property are updatable views. Classical SQL views do the mapping from stored data into virtual data. However, some applications may require updating of virtual data; hence there is a need for a reverse mapping: updates of virtual data are to be mapped into updates of stored data. This leads to the well-known view updating problem: updates of virtual data can be accomplished by updating of stored data on many ways, but the system cannot decide which of them is to be chosen. In typical solutions these updates are made by side effects of view invocations. Due to the view updating problem, many kinds of view updates are limited or forbidden.

In the ODRA project (basing on previous research) another point of view has been introduced. In general, the method is based on overloading generic updating operations (create, delete, update, insert, etc.) acting on virtual objects by invocation of procedures that are written by the view definer. The procedures are an inherent part of the view definition. The procedures have full algorithmic power, thus there are no limitations concerning the mapping of view updates into updates of stored data. SBQL updatable views allow one to achieve full transparency of virtual objects: they cannot be distinguished from stored objects by any programming option. This feature is very important for distributed and heterogeneous databases.

IV. OUR PROPOSAL

Basically, every web application, no matter how it is developed, requires the following set of logical components:

- A graphical user interface,
- A routing system,
- A business logic,
- Data to work with.

The above components could be implemented using various approaches. In some cases a programmer has to manually define them whereas other solutions use generators to create some of them automatically. Additionally, real world websites also require some static files: html templates, css, jpeg, etc.

We have decided to use pure programmatic approach which means that all necessary definitions are provided by a programmer. It may look like a lot of work, but thanks to the high level of abstraction, the amount of information is significantly reduced.

Another feature which simplifies development is MVC (Model – View - Controller) architecture which has been also utilized in many previously mentioned frameworks. Comparing to the other frameworks, our approach uses the

same object-oriented model both for a business logic (Controller) and data (Model). This method not only removes the impedance mismatch but also allows using a powerful query and programming language for developing a business logic (behavior of the application). Furthermore, it is known that query languages operate on higher level of abstraction, effectively reducing the amount of code which needs to be written to achieve the same goals. For instance, a few tenths lines of Java code could be equivalent to a literally few lines of SBQL (or SQL). Not to mention performance and various optimizations, which are much more advanced in query languages.

Another very important area of a web framework is a graphical user interface. There are different methods to deal with the topic, some of them follows the MVC pattern. One of the most popular is using a server-side templating engine. A template contains an HTML code mixed with special tags, usually provided by the framework. In most cases, the tags allow to embed parts of a programming language (e.g., Java), mainly to insert some data (e.g., a list of products or customers). However, some programmers use them to implement additional functionality which duplicates the controller's responsibility. Of course it is an incorrect application of the tags affecting maintainability of the code. At the end, tags are processed by an engine, a final HTML page is generated and sent to a web browser.

Figure 1 contains a simplified logical architecture of our prototype framework for developing web application called WebODRA. The framework consist of two principal parts:

- A web server. It is responsible for responding to incoming requests from a web browser. The implementation of the server is based on open source tool called Jetty [17];
- ODRA Database Management System. This is a standard instance of the ODRA server introduced in Section 3.

The following subsections describe each of the components (from Figure 1) in details.

A. Routing Module

In the center of WebODRA is a routing module which is responsible for a correct processing of incoming web requests. The module is driven by rules defined by a programmer. Each definition, written in SBQL (as an object with specific properties), contains the following information:

- Url. A regular expression which will be applied to the incoming request's url. If there is a match, then the rule will be executed;
- Weight. It affects an order of the processing;
- Name. Human-readable name of the rule. It is especially useful during logging;
- Additional Data. The utilization of the additional data depends on rule's kind;
- Rule's Kind. The kind of the rule which affects processing:

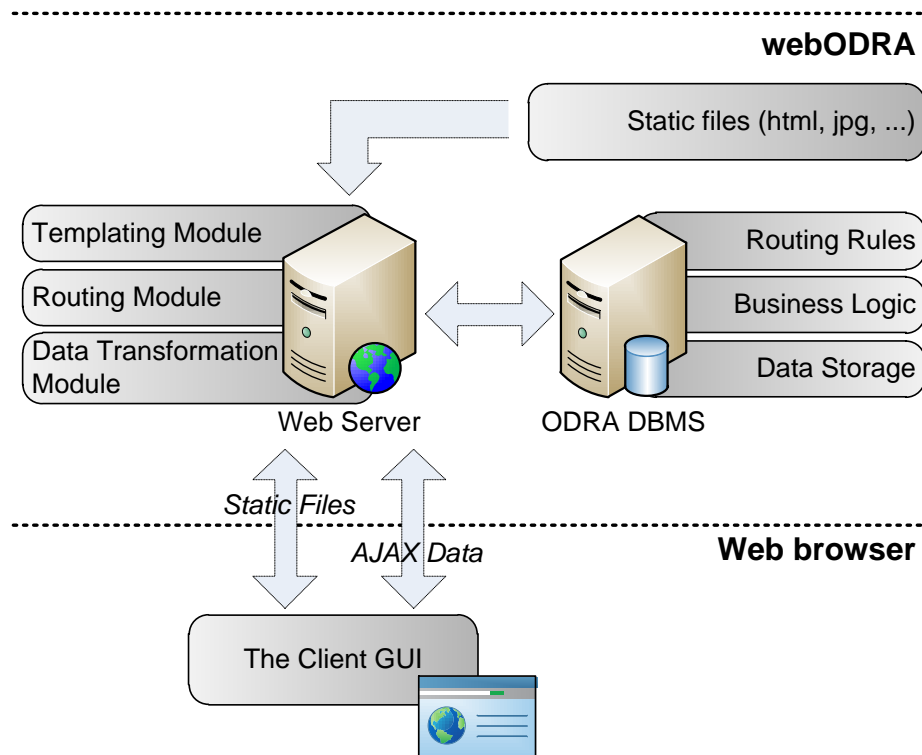


Figure 1. Logical architecture of WebODRA

- Passthrough. The web framework ignores those rules and they are processed by the Jetty server. They serve static files like: pictures, css, Java script, etc.;
- Data route. They contain a SBQL method 's name to execute. The method will get all HTML form parameters entered by a user which makes possible processing them by a SBQL code. The result of the method is transformed (see further) and returned to the browser;
- Page route. An HTML page which is post-processed by our simple templating engine (see further).

B. The Client GUI

As mentioned previously, typical server-side web templating engines, may lead to overuse tags by implementing some business functionality. To prevent this we have decided to use a client-side GUI framework. The idea is based on embedding in a web page some (meta) information which will be used to present business data. We have chosen a framework called Knockout [18] which utilizes new HTML5 *data-* attributes. They allow to create custom attributes and store any information. The process of showing a web page contains two steps. First, a HTML page is downloaded from a server, containing the markers. Then the library sends an AJAX request to asynchronously retrieve necessary data which are then “injected” into the page.

The user data submission is performed on a similar rules. An asynchronous request is send to the server, triggering a Data Rule which process the provided data.

Standard website navigation is performed using a regular hyperlinks (“outside” the framework).

C. Templating Module

The templating module is responsible for a coherent look and fill of the entire website. It operates on a single master page which has a dynamic area fulfilled with some functional pages, i.e. a document repository, a forum, news, etc. For instance, the master page can contain a header, a navigation panel and a footer.

The process is triggered by Page Route rule. When a particular page is requested by a browser, the master page is applied, or which is more correct, the requested page is embedded in the master page and then returned to the browser.

D. Data Transformation Module

When a Data Route rule executes a given SBQL method, the result could be any SBQL data type, i.e. a collection, a single object, a text. It needs to be processed to the format recognized by the Client GUI. The Data Transformation Module recursively converts the result into JSON [7] string, sends it back to the web browser where it is further processed.

V. EXAMPLE UTILIZATION OF THE FRAMEWORK

To verify usefulness of our approach, and the implemented library, we have decided to create a sample

portal with a forum functionality (Figure 2). All business logic has been defined in SBQL language and the data is stored in the ODR database. The prototype supports:

- Logon / logout with simple security model,
- Storing forums with topics and posts,
- Adding posts and topics,
- Responsive layout thanks to the Twitter Bootstrap.

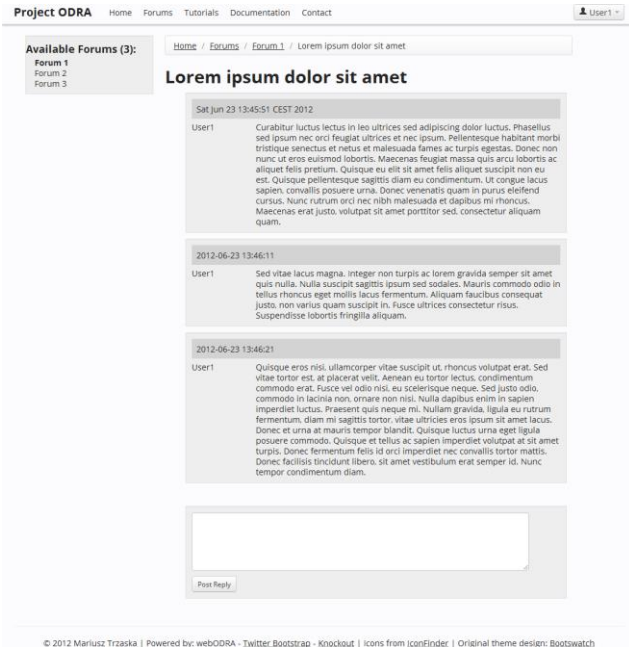


Figure 2. A sample forum developed using WebODRA framework

The example is also included in the library release available from our website.

VI. CONCLUSION AND FUTURE WORK

We have presented our approach to creating web applications using a single, coherent model utilized both for data and business logic. Thanks to the powerful query and programming language SBQL, a programmer stays on the same high level of abstraction, saving time and making less errors.

Our approach is supported by a working prototype framework called WebODRA [19]. Furthermore, we have created a sample portal with a forum functionality, proving that the idea is useful.

The contribution of this paper is based on quite new method for creating websites. To our best effort, we were not able to find a similar solution, directly employing power of a modern database to developing web portals.

We believe that this kind of solutions could be a valuable alternative to existing tools for creating data intensive web applications. Thus we would like to continue our research in that field, improving our framework to make them production-ready.

REFERENCES

- [1] Duck Typing. http://rubylearning.com/satishtalim/duck_typing.html. Last accessed: 2012-08-18
- [2] Trzaska, M.: The Smart Persistence Layer. ICSEA 2011: The Sixth International Conference on Software Engineering Advances, October 23-29, 2011 - Barcelona, Spain. ISBN: 978-1-61208-165-6. pp. 206-212.
- [3] Perscheid M., Tibbe D., Beck M., Berger S., Osburg P., Eastman J., Haupt M., Hirschfeld R.: An Introduction to Seaside, Software Architecture Group (Hasso-Plattner-Institut), ISBN: 978-3-00-023645-7 (2008)
- [4] ActiveRecord: <http://ar.rubyonrails.org/classes/ActiveRecord/Base.html>. Last accessed: 2012-08-20.
- [5] CouchApp: <http://couchapp.org/page/index>. Last accessed: 2012-08-21.
- [6] Anderson Ch., Lehnardt J., Slater N.: CouchDB: The Definitive Guide. O'Reilly Media, ISBN-13: 978-1449379681 (2010)
- [7] JSON (JavaScript Object Notation): <http://www.json.org/>. Last accessed: 2012-08-19.
- [8] Arraes Nunes, D., Schwabe, D.: Rapid Prototyping of Web Applications combining Do-main Specific Languages and Model Driven Design. Proceedings of the 6th International Conference on Web Engineering (ICWE'06; July 11-14, 2006, Palo Alto, California, USA).
- [9] Ceri, S., Fraternali, P. and Matera, M. Conceptual Modeling of Data-Intensive Web Applications, IEEE Internet Computing 6(4), July/August 2002.
- [10] Distante D., Pedone P., Rossi G. and Canfora G.: Model-Driven Development of Web Applications with UWA, MVC and JavaServer Faces. Web Engineering Lecture Notes in Computer Science, 2007, Volume 4607/2007, 457-472, DOI: 10.1007/978-3-540-73597-7_38
- [11] Williamson, J.: Oracle Application Express: Fast Track to Modern Web Applications (1st ed.), McGraw-Hill Osborne Media, ISBN 0-07-166344-4 (2012)
- [12] Subieta K.: Stack-based Query Language. Encyclopedia of Database Systems 2009. Springer US 2009, ISBN 978-0-387-35544-3, 978-0-387-39940-9, pp. 2771-2772
- [13] Adamus R., Habela P., Kaczmarek K., Kowalski T., Lentner M., Pieciukiewicz T., Stencel K., Subieta K., Trzaska M., Wislicki J.: Overview of the Project ODRA. Proceedings of First International Conference on Object Databases (ICOODB) 2008, pp. 179-198
- [14] Subieta K.: Stack-Based Architecture (SBA) and Stack-Based Query Language (SBQL). <http://www.sqpl.pl/>. Last accessed: 2012-06-15.
- [15] ODBA (Object Database for Rapid Application development): Description and programmer manual. http://www.sqpl.pl/various/ODRA/ODRA_manual.html. Last accessed: 2012-06-15.
- [16] Subieta K., Beerl C., Matthes F., Schmidt J.: A Stack-Based Approach to Query Languages. Proc. 2nd East-West Database Workshop, 1994, Springer Workshops in Computing, 1995, pp. 159-180.
- [17] Jetty - Web Server: <http://jetty.codehaus.org/jetty/>. Last accessed: 2012-08-18.
- [18] Knockout Framework: <http://knockoutjs.com/>. Last accessed: 2012-08-19.
- [19] The WebODRA framework: <http://www.mtrzaska.com/webodra>. Last accessed: 2012-11-08.

Contributing to Mathematics Lessons Authoring System (MLAS)

A Web-based Application Programming Interface

Samer F. Khasawneh

Department of Mathematics and Computer Science
The College of Wooster
Wooster, Ohio, USA
skhasawneh@wooster.edu

Abdulelah A. Algosaihi

Department of Computer Science
Kent State University
Kent, Ohio, USA
aalgosai@kent.edu

Abstract—In this paper, we give a broad overview of a Web-based system, MLAS, that enables teachers, who do not necessarily know how to program, to dynamically author and deploy mathematical lessons on the Web. Our primary focus, however, is on a feature inside MLAS; an on-Web Application Programming Interface (API). It is possible through this API to embed educational objects by any developer with XHTML/JavaScript expertise. With an intuitive and easy-to-use Graphical User Interface (GUI), programmers can deploy their own code and add new materials to be used by anybody using MLAS. The API utilizes the simple, yet powerful, site architecture which guarantees structured content placement and retrieval between the application and the back-end MySQL database.

Keywords- API; GUI; Web Technologies.

I. INTRODUCTION

Web-based Mathematics Education (WME) [1, 2], which started in 2003, is a mathematics education system that uses the Web to promote the quality of education. It aims to deliver classroom ready, dynamic, and hands-on lessons and modules to teachers and students. In WME, mathematical lessons are offered as collection of Web pages using cutting-edge Web standards such as PHP [9], JavaScript [7], Document Object Model (DOM) [8], and MySQL [12]. These web pages are connected through a giant architecture that allows easy interoperability and sharing across different schools participating in the WME project. While those WME lessons have proved to be helpful resources and students found them fun to use, from a programming perspective, the process of creating a lesson is considered to be long and challenging. Each WME lesson needs to be hand-coded and should comply with a number of requirements and follow certain protocols in order to work as intended.

The on-Web MLAS [3, 14] is an independent work under the big WME project. MLAS features a well-organized architecture that abstracts all aspects of manipulating the contents. The process of creating a lesson in MLAS is automated and content-rich lessons can be created with few mouse clicks without any programming know-how. Editing lessons enjoys the same simplicity and assumes no advanced computer skills.

Because MLAS can be thought of as a content management system for mathematics education, it might be relevant to explain some terminologies that will be used in this paper. *Manipulatives*, in general, are any objects, such as

coins, tiles, and even a paper that is cut or folded, used to help students understand abstract math concepts such as fractions and percentages in an active, hands-on approach. With the advent of the Web and based on its potential effect in enhancing the quality of education in general, and the math subject in particular, a new term has come into existence, “*virtual manipulatives*”. This term refers to those manipulatives that cannot be “touched” but rather can be “seen” on a computer screen, allowing students to explore them using computer hardware, such as a mouse and keyboard [4].

Existing Web-based systems, including MLAS and WME, assume that a mathematical lesson is a collection of virtual manipulatives. MLAS features a library of customizable virtual manipulatives. When authoring a lesson, a teacher may include one or more interactive virtual manipulatives. The manipulatives can be customized and can interact with one another or questions and comments in the lesson page. MLAS offers a growing library of virtual manipulatives that are fully customizable, editable, and reusable.

Due to the nature of the MLAS project and the need to have dedicated people adding new manipulatives, we think that it is necessary to let others contribute to expand those manipulatives of MLAS.

An *Application Programming Interface* (API) is a specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables [5]. Through the Web-based API MLAS offers, the MLAS library is easily expandable by adding new manipulatives contributed by developers and other experts.

MLAS supports two views: teacher view and student view. Obviously, a teacher is the one who controls the form in which a lesson would look to students. A lesson in the authoring stage where customization is possible is the teacher view, while the view of the “final product” in the lesson page where no customization is allowed is the student view.

This paper is organized as follows. Section II presents the related work in the field. Section III gives detailed overview of our API including its features and capabilities. Section IV shows case study on how to embed an external work to be as if it was natively supported by MLAS. We conclude this

work by presenting our views on possible development and enhancements.

II. RELATED WORK

In this field, it been somehow difficult to find a project that implement this kind of API. The reason behind it, as we researched most of the work [15][16][17][18], is that consider as project who software company take care of its maintenance (adding, deleting or editing a feature). Such softwares suffers allowing users themselves to contribute on systems. In this work, it is essential that teacher has the ability to author hands-on, fully customizable virtual manipulatives e.g. copy old web lesson and paste it at API. We built MLAS project with API feature in order to support ability to be expanded or shrinked in terms of future virtual manipulatives or removing unnecessary one. All that without necessarily programming skills. The research done on [19] aimed to solve the the cost of time and financial issues in the way of developing reusable personalized e-Learning content with appropriate metadata. In here, we are considering API helps in the reusability of manipulatives. In this context, as learning style, the idea of building reusable pedagogical components that transferable to other learning style have been introduced in [20]. Our work is different in the way of solving mentioned issues. Technically, the API architecture built in the way increase the level of the degree customization and reduce implementation overhead.

III. API OVERVIEW AND SPECIFICATION

It is possible through MLAS to embed any number of manipulatives by any developer with XHTML [11] and JavaScript expertise. With an intuitive and easy-to-use GUI, programmers can deploy their own code and add manipulatives. Such manipulatives would appear in the manipulative library so any user can use them

There are basically little to no limitations at all through the self-guided interface. XHTML, JavaScript, and CSS [13] contents can be uploaded to the server and their contents will be stored in our MySQL database. Alternatively, these contents can be written directly into designated text boxes. Because a manipulative often engages the use of dynamic resources, the interface also allows our users to upload any type of resource they wish (.swf, .class, .jpg, etc.). In short, this is a very intuitive work that strengthens MLAS and expands its usability.

It is worth to mention that throughout our literature search, we were unsuccessful trying to find a system that provides such noble feature MLAS offers.

As appears in Fig. 1, the user interface has three separate code segments for the user to fill-in with XHTML and/or JavaScript. The organization and order of these boxes were selected and arranged in a way that we think is very convenient and easy to follow. Therefore we have separated the JavaScript-only box from other boxes. We also gave the users the opportunity to directly type or even copy-and-paste contents in designated areas.

Fig. 2 shows the requirements that have to be taken into account once a user wishes to submit a manipulative. All

these requirements appear underneath the form on the same interface.

Requirements:

1. In Box 1: You need to enter only HTML inside it. Javascript code can be entered inside HTML events like onclick.
2. In Box 2: You must make explicit call to the Javascript function inside the HTML events.
 - * Parameter 1: String of HTML elements that will be saved in the DB and which will be seen by the students. This may also contain Javascript inside the HTML events.
 - * Parameter 2: String that tells the type of your manipulative.
 - * Parameter 3: The string name of your manipulative.
3. In Box 3: Define your needed Javascript functions.
4. Give your functions unique name to avoid conflicts.
5. Resources can be applets, flash files, etc. They get uploaded to "uploaded_files" directory.

Figure 2. List of requirements.

The first box is to be filled with HTML only and may have some JavaScript inside any HTML event such as *onmouseover* or *onclick*. This aids in manipulative flexibility and the multi-form property of all manipulatives. However, this HTML forms the teacher view of a manipulative and hence it is not yet ready to be previewed on the lesson page. Technically, a manipulative content has to be saved in the database in order for the script to be able to pull up its content and display it. At this point, the HTML here can only be appended to a parent element in the DOM tree and as a result would disappear with every page refresh

The second box should have JavaScript code embedded to capture specific behavior from the HTML in the first box. For instance, if the HTML from the box above had two HTML checkboxes and one is initially checked, such as:

```
<input checked="checked" type="checkbox" id="cb1" />First
<br/>
<input type="checkbox" id="cb2"/>Second <br />
```

The jQuery [6] statement below can be used in the second box of the GUI and should pop-up the ID of the selected checkbox.

```
if($('#cb1').attr('checked')==true){
    alert('cb1!');
}
else {
    alert('cb2!');}
```

To make the manipulative available for display, a programmer has to make an explicit call to the JavaScript function "save", which is defined by MLAS. This allows for a manipulative to be saved in the appropriate database table through some PHP and Asynchronous JavaScript And Xml (AJAX) [10] implementations. The *save* function takes three string parameters: The first parameter is the manipulative HTML content to be saved in the database and thus to be previewed in the lesson page. It could be related to the HTML from the first box, but they are not necessarily the

same. The second and third string parameters do not affect the functionality and behavior of a manipulative, but rather are needed to describe its overall meaning. The third parameter, in particular, can be any text and will appear in the “*progress menu*” that documents all user activities.

The third box allows the user to define the JavaScript functions needed. Very often, HTML events will make explicit calls to JavaScript functions which can be defined in this box. Users can include all the functions their manipulatives need in this box. To avoid name conflicts, we ask our users to choose unique names for their functions. Since the user might not always need to define his/her functions, this box can be left unfilled.

Once the user finishes filling out the necessary items in the HTML form and submits it, everything else gets taken care of by MLAS. The resources get saved in a dedicated directory on the server and the user’s code gets saved in a secure database. Then through PHP, we extract that code from the database and encapsulate it, along with other pieces of data, under a couple of JavaScript functions – One function will be triggered once the manipulative is called from the manipulative library (the *show* function), while the other will be called once the teacher is happy with the manipulative and decides to have it included in the lesson (the *submit* function). Fig. 3 explains the process in details.

From Fig. 3, we can see that we use PHP to write two JavaScript functions to encapsulate the user inputs which is already saved into the database. These files can then be included in the lesson page and both functions will be ready to be called once the manipulative is in use.

A typical user cannot distinguish between a user-added and an admin-added manipulative. A user-chosen manipulative image and name would appear inside the manipulative library as if they were natively supported by MLAS. MLAS also includes all the necessary code needed to make user-added manipulatives appear and interact seamlessly in their enclosing pages.

Since user’s HTML input is allowed, this implies that interactive contents can be inserted to MLAS and will be supported as well. For example Java applets and flash files can be embedded to the system with the use of the appropriate HTML tags. Below is a simple example of how an applet can be embedded to MLAS. In the applet context, only the compiled version of the Java program (*.class* extension) needs to be uploaded to the server so the browser can display the applet (assuming the browser has the Java plug-in installed).

IV. CASE STUDY

In this section we present a simple case to show how smoothly embedding an manipulative applet in MLAS. The applet that we will be showing is very simple. It is a single button labeled as “This button doesn’t do anything.” For this case, the only resources we need are the applet image and the applet *.class* file. We begin by giving our manipulative a name (say, Applet Example), and then we upload the manipulative image and the *.class* file (Assume named, *ExampleApplet.class*) through the user interface.

To embed an applet in a Web page, the HTML `<applet>` tag needs to be used, with the *codebase* attribute indicating the directory on the server where the *.class* file exists, and

the *code* attribute to denote the name of the *.class* file itself. Other attributes, such as *width* and *height*, might be used to control the size of the applet. Therefore, the first HTML box can be filled with the following HTML segment.

```
<applet codebase = "uploaded_files/"
      code=" ExampleApplet.class"
      width = "400" height = "50" > </applet>
```

The GUI in Fig. 2 clearly indicates that all user uploads go inside “*uploaded_files*” directory. Therefore that directory is referenced in the *codebase* attribute above.

In order to properly display the applet in the lesson page, we need to have the above applet tag to be the first parameter of the *save* function. Since, in this case, the teacher and student views are similar, nothing else needs to be added to the second box. That can be something like:

```
var applet = '<appletcodebase = "uploaded_files/" ';
applet += 'code = "ExampleApplet.class"';
applet += 'width="400" height="50"></applet>';
```

```
save (applet, "Applet", "Applet added!");
```

All other fields of the HTML form can be left blank, and the user can now proceed. The applet image will now be available in the manipulative library together and a click on that image will show the applet in the lesson (Fig. 4)

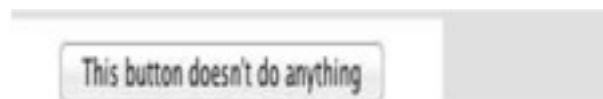


Figure 4. The applet appears in the lesson.

To test this API, we have also embedded a numerous lessons and manipulatives. The time to add such one was fast with no problem encouraged. Requirements for the API ensure that MLAS work smoothly without affecting of removing or adding manipulative. We were effortlessly able to import any desired content and have them work perfectly under MLAS’s framework. Table 1 below shows the time taken on seconds to Add/Remove a manipulative with respect to expert or Non-Expert. As Non-expert user, our sampling included teacher with modest to no expertise. The other way, adding/removing without API, to deal with manipulative is to go to the row php page write necessary piece code and test it. In compare that with API usage, the difference noticeable in term of time and effort. The time ratio in the table shows the deference ratio and how much time this technique save. This ratio represent how this API saved time to write necessary piece of code e.g. PHP/MySQL and test its result and presentation. The reader is referred to [14] for more examples.

TABLE 1. shows the cost in time for applying API feature to MLAS framework.

	Add with API	Add without API	D e f ratio	Rem. with API	Rem. without API	D e f ratio
Expert	31	54	42.6%	14	24	41.6%
Non-Expert	84	2167	96.12%	35	808	95.6%

V. CONCLUSION AND FUTURE WORK

In this work, we briefly introduced an on-Web lesson authoring system, MLAS, for mathematics education. Our system permits teachers and experts to access and author dynamic mathematical lessons without any programming know-how. To achieve this, MLAS offers a growing library of virtual manipulatives that can be extended through a well-organized API. The architecture of the system and its collaborative constituents make it easy to share and exchange content both within and beyond MLAS.

We are constantly trying to reduce the requirements of our API to make it even better. As a possible future work, it might be useful to add a feature that allows the automatic extraction of the HTML/JavaScript code of any external manipulative to be inserted into the API. Further, we are in the process of empowering the entire MLAS with the upcoming HTML5 standards. With HTML5, MLAS would have more dynamic features and be more up-to-date with the Web standards.

REFERENCES

- [1] P. Wang, M. Mikusa, S. Al-shomrani, D. Chiu, X. Lai, and X. Zou. Features and advantages of WME: a Web-based mathematics education system. In Proceedings of the IEEE Southeast Conference. Florida, USA, 2008. pages 621-629.
- [2] P. Wang, M. Mikusa, S. Al-Shomrani, X. Lai, X. Zou, and Zeller. "WME: a Web-based Mathematics Education System for Teaching and Learning." ICME 11 – TSG 22 Theme 3 the 11th International Congress on Mathematical Education. Mexico, July 2008.
- [3] S. Khasawneh and P. Wang. "Overview of Mathematics Lessons Authoring System (MLAS)". Proceedings of CSEDU 2012, Porto, Portugal, pp. 48-54, April 2012.
- [4] CITED Research Center, Learning Mathematics with virtual manipulatives, <http://www.cited.org/index.aspx>. Retrieved: Jan, 2013.
- [5] Application Programming Interface. http://en.wikipedia.org/wiki/Application_programming_interface. Retrieved: Jan, 2013.
- [6] jQuery. <http://www.jquery.com> Retrieved: Jan, 2013.
- [7] JavaScript. <http://en.wikipedia.org/wiki/JavaScript>. Retrieved: Jan, 2013.
- [8] Document Object Model (DOM). Technical report, <http://www.w3.org/DOM/>. Retrieved: Jan, 2013.
- [9] PHP: Hypertext preprocessor. Technical report, <http://www.php.net/> Retrieved: Jan, 2013.
- [10] Asynchronous JavaScript and Xml (AJAX). Technical report, <http://developer.mozilla.org/en/docs/AJAX>.
- [11] XHTML. <http://en.wikipedia.org/wiki/XHTML> Retrieved: Jan, 2013.
- [12] MySQL. <http://en.wikipedia.org/wiki/MySQL>. Retrieved: Jan, 2013.
- [13] Cascading Style Sheet (CSS) to style HTML elements . http://en.wikipedia.org/wiki/Cascading_Style_sheet. Retrieved: Jan, 2013.
- [14] S. Khasawneh. "A Web-based Lessons Authoring System for Mathematics Education". PhD dissertation. 2012
- [15] <http://www.articulate.com/> Retrieved: Jan, 2013.
- [16] <http://www.courselab.com/>. Retrieved: Jan, 2013.
- [17] <http://www.elicitus.com/>. Retrieved: Jan, 2013.
- [18] <http://www.ispringsolutions.com/>. Retrieved: Jan, 2013.
- [19] O. Conlan, D. Dagger, and V. Wade. "Towards a standards-based approach to e-Learning personalization using reusable learning objects". In: Driscoll, M. and Reeves, T.C. (eds.) Proceedings of World Conference on E-Learning AACE. Montreal, Canada, October 15-19, 2002. PP. 210-217.
- [20] C. Bruen and O. Conlan. "Dynamic Adaptive ICT Support for learning Styles – A Development Framework for re-useable learning resources for different learning styles & requirements". Proceedings of the ITTE 2002, Annual Conference of the Association of Information Technology for Teacher Education. 2002 pp. 1238-1241. Chesapeake, VA

```

// Assume the manipulatives table is fetched from the database and stored in $manip variable
//Now, also define a file variable and have a file ready (manipulatives.js) to accept content write

$file = fopen("manipulatives.js", "w");

//Go through the manipulatives table one record by another and place each manipulative in a JavaScript function
foreach($manip as $m) {

//Define a PHP variable that contains some JavaScript code,
// which later will be appended to the page. $m['name'] here refers to the name of the manipulative
    $function = "function show_". str_replace(' ', '_', $m['name']) . " () {\n";

//Assuming user directly placed code in box ($m['html'] is user's first box content). Then, perform some code cleaning
    $m['html'] = str_replace('\n', '\n', $m['html']);
    $m['html'] = str_replace(array("\r\n", "\r", "\n"), ' ', $m['html']);

//Add the buttons to allow work saving or cancellation
    $submit = '<input type="button" value="Proceed" onclick="submit_'. str_replace(' ', '_', $m['name']) . '().";';
    $submit .= 'class="sbmt"><input onclick="cancel()" type="button" value="Cancel">';

//Finally allow user's code to append to the page and end of the show function

    $function .= '$("#page_element").append(\<div>'. $m['html'] . ' ' . $submit . '</div>\n)';
    $function .= "}\n";

//Write content to file
    fwrite($file, $function);

// Define the submit_ $m['name'] which will contain the contents of box 2 including
// the call to save function that will save code in the database
    $function = "function submit_". str_replace(' ', '_', $m['name']) . " () {\n"

//Similarly, do code cleaning as we are assuming user has put code directly in box 2
    $m['js'] = str_replace(array("\r\n", "\r", "\n"), ' ', $m['js']);

    $function .= $m['js'] . "\n";

    $function .= "}\n"; //end of the submit function

//Write content to file and then close it
    fwrite($file, $function);
    fclose($file);

} //end of foreach loop. Finally, the file has to be included in the page for this to work
echo '<script type="text/javascript" src="manipulatives.js"></script>';

```

Figure 1. The API.

Add new manipulative	
* indicates optional field	
Manipulative Name:	<input type="text"/>
Manipulative image: *(if one is not uploaded, default one will be created)	<input type="text"/> <input type="button" value="Browse..."/>
CSS file: *	<input type="text"/> <input type="button" value="Browse..."/>
Source one: *	<input type="text"/> <input type="button" value="Browse..."/>
Source two: *	<input type="text"/> <input type="button" value="Browse..."/>
Source three: *	<input type="text"/> <input type="button" value="Browse..."/>
Box 1: HTML Code >> [Teacher's view]	
<div style="border: 1px solid #ccc; height: 100px;"></div>	Or upload your file: <input type="text"/> <input type="button" value="Browse..."/>
Box 2: Javascript Code >> Need to make explicit call to the Javascript "save" function:	
<div style="border: 1px solid #ccc; height: 100px;"></div>	Or upload your file: <input type="text"/> <input type="button" value="Browse..."/>
Box 3: Javascript Code >> Javascript functions needed: *	
<div style="border: 1px solid #ccc; height: 100px;"></div>	Or upload your file: <input type="text"/> <input type="button" value="Browse..."/>

Figure 3. PHP handling user's input.

Restraining technical debt when developing large-scale Ajax applications

Yoav Rubin, Shmuel Kallner, Nili Guy, Gal Shachor

IBM Research - Haifa
Haifa University Campus
Haifa, Israel

{yoav, kallner, ifergan, shachor}@il.ibm.com

Abstract - Addressing technical debt during the software development process relies heavily on a refactoring phase, in which automatic code transformations are used as a crucial mechanism to reduce a system's technical debt. However, automatic refactoring is not an option when developing Ajax applications. Therefore, an approach that restrains the accumulation of a system's technical debt is needed. In this paper, we present and evaluate such an approach and its reification as a framework. We conclude that our proposed framework enables restraining technical debt in a large-scale Ajax application without the need for automatic code refactoring tools.

Keywords: *software engineering; dynamic languages; code reuse; technical debt; Ajax*

I. INTRODUCTION

Software development is an engineering discipline, and as such, it is composed of an ongoing process of decision making on the one hand and acting upon these decisions on the other. A key aspect in a project's decision-making process is handling technical debt [1]—the toll that suboptimal decisions or actions impose on the future welfare of that project. Technical debt is resolved using technical means and resources.

The impact of suboptimal decisions on a project resembles the impact of financial debt. In some cases, incurring small debts can result in large future rewards. Yet, debt usually comes with interest, which if not paid on time can inflict severe consequences, including a complete halt of the related activity [1].

Technical debt is considered one of the causes of hatching a catastrophe [2] and may affect the eventual success of a software project.

One of the most common technical debt payback strategies, which endeavors to decrease, manage, and control technical debt [3], is code refactoring [4]. This is a code modification process, that can be done either manually or using automatic tools. The essence of this process is to apply behavior-preserving transformations to the code in a way that the resulting code provides better reusability, compatibility among different components, and simplicity of the iterative software design process [5].

A. Refactoring dynamic languages codebase

Ever since the refactoring browser [6] was introduced, targeting the Smalltalk-80 [7] programming language, several attempts were made to create refactoring tools for dynamic languages [8, 9]. These tools aimed at performing

automatic refactoring transformations on code written in a dynamic language such as Ruby [10] or JavaScript [11]. Each such tool tried to overcome the lack of type information, which is essential for correct refactoring transformations [5], by using other sources of information. In the refactoring of Smalltalk codebase, the automatic tool used a combination of test-cases, results of dynamic analysis, and method wrappers [6]. Another technique is static pointer analysis, which was the vehicle that drove automatic refactoring in JavaScript codebases [9]. Another strategy was to rely not only on the analysis of a project's codebase, but rather on additional information provided by the developers, as was done in a Ruby codebase refactoring mechanism [8].

However, automatic refactoring that is based on the techniques described above does not always result in behavior-preserving transformations. Basically, these tools breach the complete correctness requirement that is assumed by developers when using automatic tools. This partial correctness is unavoidable. It can be attributed to the fact that these tools rely on the existence of non-compulsory information, such as a test suite with complete coverage [6], or assume the absence of dynamic behavior [9].

The semi-automatic approach that relies on user input also has its downsides, as it may lead to user errors and suffers from occasional false-negative effects [8].

B. Constraints of Ajax development

The frontend development of web applications is a special case of using a dynamic programming language.

In this domain, the software development is usually done using a collection of technologies termed Asynchronous JavaScript and XML—Ajax [12]. Ajax builds a complete stack of technologies, from document structuring in HTML [13] through its internal representation using Document Object Model (DOM). APIs [14] and visual aspects are modified using Cascading Style Sheets (CSS) [15]. Communication is usually done using the XMLHttpRequest API [16], while interaction among all of the above technologies (and many more) is done using the JavaScript programming language [17].

All modern web browsers implement the stack of Ajax technologies, though the implementations are not identical. Therefore, in addition to understanding each of these technologies, Ajax developers face the cross-browser compatibility problem [18]. Each technology must be

executed within different browsers that might have slightly different semantic interpretations of syntactic elements or might simply have implementation bugs [19].

To reduce the efforts involved with using several technologies and to address the problem of several implementations for each technology, two main approaches are used when developing Ajax-based software [20]:

- The first strategy is to write code in a non-Ajax technology that compiles into Ajax code. One example for this approach is using GWT [21], which is based on Java technologies. Another example is using the CoffeeScript [22] language, which provides syntactic sugaring on top of JavaScript.
- The second strategy is to use a web development library that buffers the different incompatibilities among browsers. Examples for such libraries are YUI [23] and Dojo [24].

Combining these two approaches is possible by using a development platform that is not based on Ajax code but rather compiles down to JavaScript code, which runs on top of a JavaScript library, e.g., ClojureScript, [25] which is compiled to run on top of Closure [26].

The main drawback of the first approach results from the fact that another level of indirection has been added, possibly making it difficult to trace problems in runtime back to the appropriate location in the source code. The main drawback of the second approach results from the fact that libraries also define their own coding idioms, which are different than those of pure JavaScript. Specifically, libraries tend to provide their users mechanisms of object-oriented programming (OOP). This is done in each library by providing unique definitions of a metaobject and of metaobject protocols [27]. This can be thought of as an additional, ad-hoc programming language layer that is specific for the defining library.

The inconsistency among the coding idioms of the various JavaScript libraries results in the inability to create automatic refactoring tools that are library-agnostic, as each such library requires its own code analysis and refactoring mechanisms.

Due to the dynamic nature of JavaScript alongside the differences among the coding idioms of different Ajax libraries, automatic refactoring tools that target Ajax code base do not exist.

The absence of these tools results in a situation in Ajax codebase in which resolving technical debt by using refactoring is done manually. Thus, this process is demanding, error-prone, and difficult to perform, especially when large transformations are needed.

Based on the previously described constraints, along with the innate nature of dynamic languages, using our financial analogy, we can describe technical debt in an Ajax application as a loan shark debt. This is due to the cost of falling back on payments—using dynamic languages means that many errors are detectable only at runtime. This type of debt also leads to the almost impossibility of paying it off

once it starts to accumulate (no automatic refactoring tools exist). Naturally, preventing such debt and restraining it once it starts to accumulate should be a high priority.

The purpose of this work is to describe how a small development team was able to use our framework to deliver a large-scale web application in a relatively short time, while facing the issue of technical debt in an Ajax application. Our approach does not rely on automatic tools, but rather proactively uses abstractions and patterns [28] and especially adheres to the idea of lists as the skeleton of software components [29]. Our approach resulted in a software project that incorporated mechanisms to prevent and restrain technical debt so as to enable the successful delivery of a high-quality product.

The remainder of the paper is structured as follows: Section 2 describes related work. Section 3 introduces the project; Section 4 discusses the abstraction and the way that it was reified. Section 5 presents the evaluations performed with regard to the use of the abstraction. We present the results of our evaluations in Section 6 and we explain them in Section 7. Finally, Section 8 concludes the paper and outlines possible future directions.

II. RELATED WORK

The idea of addressing technical debt as part of the development process, though initially presented decades ago [1], has just begun to resurface and has gained significant interest in the software development community in recent years [31]. As such, not much academic research has been published on this issue to date. Moreover, most of the existing work revolves around the management of technical debt, with an "after the fact" approach, namely by employing various code refactoring methods [4]. This is accompanied with a decision-making process to optimize debt reduction while facing the costs of the code refactoring [32].

Amongst the work that was performed to date on technical debt, we are not aware of any work that is focused on approaches to restrain such debt in a "factor instead of refactor" fashion. An iterative approach, which can be thought of as a compromise between a "post-debt" and "pre-debt" approach, was discussed by Nanette Brown, et al. [33]. They suggested methods to assess the resulting technical debt in an iterative architectural project planning process by using dependency analysis. Such measurements can help in making the right architectural decisions and thus decrease the accumulating debt.

Handling technical debt in a large-scale web application project was discussed by Israel Gat and John D. Heintz [34]. Their paper presents how the Cutter's technical debt assessment tool, which employs both static and dynamic code analysis methodologies, was used to define a technical debt reduction project—one that included a complete rewrite of the frontend component in JavaScript. Reassessment of the new frontend implementation showed that the amount of code duplication remained significantly high (40%).

III. USE CASE

A team needed to develop a web frontend component of a case management [30] product using Ajax technologies, particularly the Dojo toolkit [24]. The project was assigned to a team composed of a lead developer experienced with web application development and a few developers lacking this expertise. The project itself had a tight schedule and needed to be released as part of a larger product with strict deadlines. It had to be developed using an agile methodology, as most of the user interface and user experience requirements were to be defined in an iterative fashion. From day one of the project, the team could clearly see that due to the time constraints, development friction resulting from technical debt could cause the entire project to fail and would have a severe impact on the entire product. Technical debt cannot be overlooked and must be avoided. A solution that prevents the future accumulation of technical debt had to be devised before any other aspect of the component could be developed.

IV. SOLUTION

A. Code and abstraction reuse

We designed a development strategy in light of the experience gained by the team's technical leader in previous web application development projects. Our strategy was to base the software components on a single abstract idea, whose essence is that an application's frontend is composed of various manageable lists of repetitive items, each consisting of another element. Within each list, the items are identical in their list management behavior (adding, changing location, removing), yet they may vary in presentation as well as in the elements that each list item contains.

To allow maximal reusability of this abstraction, we needed to develop an implementation that was as flexible as possible. As such, we developed an implementation that could handle all the list management related aspects and the entire lifecycle of the nested elements, all while remaining presentation-agnostic.

A hidden design agenda of the manageable list abstraction was to force its users to provide code that adapts the abstraction's core functionality alongside the presentation rules, within each use. This would result in constructing a mental model of the abstraction's capabilities from the beginning. Our intention was to verse the developers in using the abstraction for all types of needs, thus enabling them to compose much of an application's frontend from building blocks that are extensions of this idea.

B. The Wrapper/WrapperContainer framework

We turned the reification and implementation of the managed list abstraction into a framework composed of three classes. Two classes correspond to a list—one for a general list and one that supports a drag and drop operation

among the list items. The third class corresponds to a list item. We implemented the following responsibilities into the framework:

- List management
- Event handler with callback hooks
- Lifecycle management of the list, items, and the nested elements

The list item abstraction was implemented in a class called Wrapper, as it acts as a general wrap for any kind of element. The list abstraction was implemented in the class WrapperContainer, as it acts as a container for wrappers, and DndWrapperContainer, which stands for a WrapperContainer that supports drag and drop operations.

The entire framework was implemented in six hundred and forty lines of code (LOC), all in JavaScript and using the Dojo toolkit APIs. The hooking up of the callbacks as well as the possibility to manage the lifecycle of the wrapped element was based on the dynamic nature of JavaScript alongside its idiomatic usage of runtime time inspection.

V. EVALUATION

A. Methodology

To understand the impact of using the abstraction and framework we described above on the project, and especially to determine if it stood up to its target of technical debt restraint, we designed and performed two different evaluations. The first is based on lines of code analysis and the second on a review by a group of experts. This combination of methodologies was picked so it would provide a clear view as to whether the framework was used appropriately, and if so, the extent of its usage.

B. Analysis of the project's codebase

We completed the first evaluation by performing a static analysis of the project's code to measure the portion of reuse that can be attributed to the framework in an attempt to reveal the cost effectiveness of investing in designing, implementing, and using it. Our results pointed out the extent of the framework's use, and hence its significance in the overall codebase.

To perform this analysis, we divided the codebase into three distinct components:

- Framework: the code that was used to develop the Wrapper/WrapperContainer framework
- Extensions: the code that was used to develop the widgets that extend the Wrapper/WrapperContainer framework (a widget is a class, or other software component, which also has a visual representation)
- Other: all the project's code that is not part of the framework or extending it

In our analysis, we concentrated only on the portion of the Ajax codebase that was written in JavaScript, as HTML code is almost always tailored for a specific use. Also, most of the CSS code was part of a library that was used

throughout the organization—one that was not developed as part of the project. Another part of the project's codebase that we ignored in this measurement was a small JavaScript library that was developed in another project and was used "as-is".

C. Review by experts

The second evaluation was done by having five software engineers versed in the domain of large-scale web application development perform reviews of the project.

These engineers were qualified as experts based on the following "expert's threshold" criterion:

- More than 10 years of professional experience as a software engineer
- Of which, at least 5 years working as a front-end engineer
- Of which, at least 3 years working as part of a team that develops large scale Ajax-based web application

We educated the evaluating engineers about the Wrapper/WrapperContainer framework and asked them to use the application and inform us of any place in the application where they see fit for using our framework. Their answers were later compared to their actual use of the framework.

The analysis of the overlap between the reviewers' answers to their actual use was performed to gain an understanding on the use coverage of the framework, i.e., whether the team had used it as much as possible, thus efficiently restraining the project's technical debt. This is especially important in lieu of the hidden agenda behind the design of the abstraction. Moreover, from the reviewers' answers, we could see whether the abstraction indeed fits the domain.

On top of that, as a side-effect of this measurement, we can detect whether technical debt still exists in the system due to not using the framework where it could have been used.

VI. RESULTS

A. Analysis of the project's codebase

We present the results of our first evaluation in Table I, showing that the Wrapper/WrapperContainer framework was extended twenty times in the project, as each file corresponds to a class. The forty files marked as extensions are basically twenty pairs, with each such pair composed of a class that extends Wrapper and a class that extends either WrapperContainer or DndWrapperContainer, depending on its need to provide a drag and drop behavior to the user.

TABLE I. PROJECT'S CODE BY COMPONENT

	Framework	Extensions	Other	Total
Number of LOC	640	9054	13725	23419
Percentage of LOC of entire project	2.73%	38.66%	58.61%	100%
Number of files	3	40	70	113
Percentage of files of entire project	2.65%	35.39%	61.96%	100%

In light of the large number of reusing classes, especially when considering the fact that more than a third of the project classes are extensions of the framework, we can easily accept that the designed abstraction does play a central role in the project.

Also of note is the fact that the portion of the code that extends the framework attributes for more than 38% of the application's code. When we look at the framework alongside its extenders, we see that the list abstraction covers more than 40% of the application code.

From these numbers, not using this abstraction and solving each of the twenty usage scenarios differently would clearly have enforced a large allocation of resources—such as adding more time by delaying the project deadlines or adding more developers. Needless to say, these solutions were unacceptable.

Moreover, in cases in which technical debt is created as part of a specific extension of the framework, it would be secluded from other parts of the application. This results in reduced code cohesiveness and minimal effect of each scenario on the overall technical debt of the system.

B. Review by experts

The results of the second evaluation are presented in Table II, which summarizes several review sessions that were held with three experts. It is important to state that we believe that the reviewers' high expertise and deep knowledge in the domain of web applications development more than compensates for the small number of reviews.

Table II shows how many locations in the application each reviewer thought were applicable for using the framework (the Found column). Such locations were marked either as a location where the development team had indeed used the framework (the In use column) or marked as a location where the team did not use the framework (the Not in use column).

Table II clearly shows that the abstraction that was reified by the framework is indeed a natural fit for the project. The table also hints that it can be used in other frontend projects, as the five reviewers found a high number of places to use it within the discussed application. This is due to the reviewers' familiarity with the usage of the managed list abstraction in such applications. Moreover, although not presented in this table, all of the twenty places where the team used the framework were detected by the experts (when we superpose their reviews).

TABLE II. REVIEWS SUMMARY

	Found	In use	Not in use
Reviewer-1	18	18	0
Reviewer-2	17	17	0
Reviewer-3	16	16	0
Reviewer-4	19	19	0
Reviewer-5	19	18	1

One usage location that was pointed out by Reviewer-5 was marked as Not in use. The framework was not used there since it required a modification to the framework that would change its semantics, a task that the team preferred not to do at the time that that specific location in the application was implemented.

VII. DISCUSSION

In addition to the results presented in the previous section, it is important to state that the project was delivered on time, with high quality, and was praised by its stakeholders and clients. Thus, in light of the aforementioned statements and based on the presented results, we would like to address the idea of using the managed list abstraction alongside its logical reification by the Wrapper/WrapperContainer framework from several perspectives.

A. Software design perspective

From a software design perspective, the Wrapper/WrapperContainer framework, as reification of the managed list abstraction, was a natural fit to act as a main component in a large-scale Ajax project. This component had proved itself as flexible enough to be used in numerous contexts while preserving and reusing its core functionality.

The decision to provide a non-holistic component that incorporates a complete logic implementation with callback hooks and that lacks visual representation resulted in a highly usable and flexible component as the apparent choice in the trade-off between adaptiveness and rigorousness.

B. Developer's cognitive load perspective

We discovered that the use of a single abstraction as the main workhorse of the frontend code had a twofold benefit:

- A lower learning curve: The project's novice developers had to learn only one main abstraction and quickly become proficient in using it and its reification. They were able to do so after an almost insignificant portion of time with respect to the entire project's duration. The reviewing experts understood it after a thirty minute educational session.
- The proficiency of the developers in using the abstraction and the framework: Interpreting that the large number of uses of the framework within the project was a result of the assimilation of the abstraction and the framework into the developers' mental arsenal is sensible. Thus, the developers used it in all appropriate situations. Even during times when

the deadline pressure increased, the developers still thought of using the abstraction and the described framework as the path of least resistance.

These two gains resulted in a lower mental burden on the developers, which allowed them to free mental resources to make better decisions and find better solutions in the overall development process.

C. Debt accumulation perspective

As a result of the limitations imposed by the project's technological domain, the development process of "first code then refactor" was thought of as inadequate. The lack of automatic refactoring tools forced developers to think ahead about their solutions and code to come up with a development flow that did not assume the existence of automatic refactoring tools. This approach was nicknamed "factor instead of refactor". The absence of such an efficient debt payment mechanism resulted in the emergence of a paradigm that minimizes the accumulation of technical debt. This paradigm achieved its goal by focusing on a highly useful abstraction with flexible implementation. This kind of focus had an effect on the project's technical debt similar to the effect of a highly rewarding investment. Basically, since the framework was reused twenty times throughout the project, we can say that the "factor instead of refactor" approach was nineteen times more efficient than the "code then refactor" approach.

We can conclude that using the managed list abstraction and the Wrapper/WrapperContainer framework as a mechanism to control and restrain technical debt in a large-scale Ajax project has proven itself beyond any expectation of the development team. Its contribution to the successful delivery of the project, on time, and with high quality, is highly significant.

VIII. CONCLUSIONS AND FUTURE WORK

Large-scale web applications are becoming abundant for various reasons. An Ajax-based frontend is a crucial component in cloud-based applications as well as in non-native mobile applications. Moreover, users are expecting to have the ability to access their once desktop-only applications via web interfaces. However, the domain of web application development is relatively young, and due to its special constraints, traditional software development processes and tools are rarely sufficient.

Mapping knowledge and ideas that are applicable for static languages to be used in dynamic languages is in dire need. Tools that rely on information that is extracted from a programming language type system (such as automatic refactoring tools) have a key role in the development process of modern software. Since such information is not found in Ajax-based applications, these tools are not available for software developers, and thus standard development processes become less effective up to the point that a project's success can be jeopardized.

Methodologies and paradigms that handle problems that occur in large software projects need to be adapted to web application projects. One such problem, addressed in this work, is how to restrain technical debt. In this paper, we presented one solution—the abstraction of a managed list and its implementing framework. However, other abstractions may fit other types of projects. These abstractions target not only JavaScript components but other technologies as well, such as CSS or HTML.

Finding ways to track technical debt that originated from components implemented using different technologies and multiple programming languages, as part of a single software project, is also necessary. Moreover, we also must address the innate technical debt that is found due to the cross browsers compatibility problem. As such, finding ways to mitigate it into a debt-tracking system that does not yet exist is also a worthy research direction.

ACKNOWLEDGMENT

The authors would like to thank Jayasimha S Kanakatte, Kalaivanan Saravanan, Ravi Ray, Seema Meena, and Susheel Ahuja for all the time they invested on the project. We would also like to thank Maya Barnea, Yossi Mesika, and Andrei Kirshin for the fruitful discussions.

REFERENCES

- [1] W. Cunningham, "The WyCash Portfolio Management System" in Addendum to the proceedings on Object Oriented Programming Systems, Languages, and Applications, pp. 29-30, 1992.
- [2] F. P. Brooks Jr, "The Mythical Man-Month" (anniversary ed.). Addison-Wesley Longman Publishing. 1995, pp. 66-69.
- [3] A. Nugroho, J. Visser, and T. Kuipers, "An Empirical Model of Technical Debt and Interest". In Proc. of the 2nd International Workshop on Managing Technical Debt (MTD 2011), 2011.
- [4] M. Fowler. "Refactoring: Improving the Design of Existing Code". Addison-Wesley Longman Publishing. 1999.
- [5] W. F. Opdyke, "Refactoring Object Oriented Frameworks". PhD thesis, University of Illinois at Urbana-Champaign, 1992.
- [6] D. Roberts, J. Brant, and R. Johnson, "A Refactoring Tool for Smalltalk". Theory and Practice of Object Systems. Vol 3, Issue 4, 1997. pp. 253-263.
- [7] A. Goldberg and D. Robson, "Smalltalk-80: The Language and its Implementation". Addison-Wesley Longman Publishing. 1983.
- [8] T. Corbat, L. Felber, M. Stocker, and P. Sommerlad, "Ruby Refactoring Plug-in for Eclipse. In proceedings of Object Oriented Programming, Systems, Languages, and Applications. 2007. pp. 779-780."
- [9] A. Feldthaus, T. Millstein, A. Moller, M. Schafer, and F. Tip, "Tool-supported Refactoring for JavaScript". In Proceedings of the ACM International conference on object oriented programming systems languages and applications. 2011. pp 119-138
- [10] D. Flanagan and Y. Matsumoto, "The Ruby Programming Language, first Edition". O'Reilly Media. 2008.
- [11] ECMA. ECMAScript Language Specification, 5th edition, 2009. ECMA-262 - accessed Aug. 13th, 2012.
- [12] J.J Garret, "Ajax: a New Approach to Web Applications, <http://adaptivepath.com/ideas/ajax-new-approach-web-applications>. - accessed Aug. 13th, 2012.
- [13] The World Wide Web Consortium (W3C), "HTML 4.01 Specification", <http://www.w3.org/TR/REC-html40/> - accessed Aug. 13th, 2012
- [14] The World Wide Web Consortium (W3C), "Document Object Model (DOM) Level 2 Core Specification", <http://www.w3.org/DOM/> - accessed Aug. 13th, 2012
- [15] The World Wide Web Consortium (W3C), "Cascading Style Sheets", <http://www.w3.org/Style/CSS/> - accessed Aug. 13th, 2012.
- [16] The World Wide Web Consortium (W3C), "XMLHttpRequest", <http://www.w3.org/TR/XMLHttpRequest/> - accessed Aug. 13th, 2012
- [17] A. T. Holdener III. "Ajax: The Definitive Guide". O'Reilly Media. 2008.
- [18] A. Mesbah and M. R. Prasad, "Automated Corss-Borwser Compatibility Testing". Proceedings of the 33rd International Conference on Software Engineering (ICSE 2011).
- [19] A. Taivalsaari, T. Mikkonen, D. Ingalls, and K. Palacz, "Web Browser as an Application Platform: The Lively Kernel Experience". Sun Microsystems Laboratories Technical Report TR-2008-175, January 2008.
- [20] T. Mikkonen and A. Taivalsaari. "The Mashable Challenge: Briding the Gap Between Web Development and Software Engineering". In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10). 2010.
- [21] Google, Inc., "Google Web Toolkit Overview". <http://code.google.com/webtoolkit/overview.html> - accessed Aug. 13th, 2012.
- [22] <http://coffeescript.org/> - accessed Aug. 13th, 2012.
- [23] Yahoo! Developer Network, "YUI Library". <http://developer.yahoo.com/yui/> - accessed Aug. 13th, 2012.
- [24] The Dojo Foundation, <http://dojotoolkit.org/> - accessed Aug. 13th, 2012.
- [25] S. Sierra, "Introducing ClojureScript". <http://clojure.com/blog/2011/07/22/introducing-clojurescript.html> - accessed Aug. 13th, 2012
- [26] Google, Inc. "Closure Tools". <http://code.google.com/closure/> - accessed Aug. 13th, 2012.
- [27] G. Kiczales, J. des Rivieres, and D. G. Bobrow. "The Art of the Metaobject Protocol". Cambridge, MA: The MIT Press, 1991.
- [28] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley, 1994.
- [29] J. McCarthy, "Recursive Functions of Symbolic Expressions and their Computation by Machine, Part I". Communications of the ACM, vol. 3 Issue 4, pp. 184-195, April 1960.
- [30] M. Zisman, "Representation, Specification, and Automation of Office Procedures". PhD thesis. Wharton Business School, University of Pennsylvania, 1977.
- [31] N. Brown, Y. Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R.L. Nord, I. Ozkaya, R. Sangwan, C. Seaman, K. Sullivan, and N. Zazworka. Managing technical debt in software-reliant systems. In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10). 2010.
- [32] N. Zazworka, C. Seaman, and F. Shull. "Prioritizing Design Debt Investment Opportunities". In Proc. of the 2nd International Workshop on Managing Technical Debt (MTD 2011), 2011.
- [33] N. Brown, R.L. Nord, and I. Ozkaya, M. Pais. "Analysis and Management of Architectural Dependencies in Iterative Release Planning". In Proceedings of the Ninth Working IEEE/IFIP Conference on Software Architecture (WICSA). 2011.
- [34] I Gat, J. D. Heintz. "From Assessment to Reduction: How Cutter Consortium Helps Rein in Millions of Dollars in Technical Debt". In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10). 2010

Distributed OSGi through Apache CXF and Web Services

Irina Astrova
 Institute of Cybernetics
 Tallinn University of Technology
 Tallinn, Estonia
 irina@cs.ioc.ee

Arne Koschel
 Faculty IV, Department for Computer Science
 University of Applied Sciences and Arts Hannover
 Hannover, Germany
 akoschel@acm.org

Abstract—The OSGi Service Platform supports rudimentary distribution through Universal Plug and Play (UPnP) specification, which facilitates interaction with UPnP-enabled consumer devices. The main goal of UPnP is to allow simple and seamless connection between devices and sharing of those devices. Although UPnP can be seen as a distributed system, the range of its use is very limited. Yet the way how OSGi services interact to each other is constrained to a single Java Virtual Machine (JVM) where they run. This prevents to provide OSGi services in a distributed manner. Therefore, the main goal of this paper is to add distribution capability to OSGi, without having to change OSGi itself. The contribution of this paper is twofold: (1) it supplies implementation details to show how OSGi can be extended with distribution; and (2) it implements a flight information system to show how this extension can be applied to business applications.

Keywords—OSGi Service Platform; Web services; Apache CXF; distribution; flight information system.

I. INTRODUCTION

The OSGi Service Platform [1] is an emerging successful Java-based standard for developing component-based software. As its core, OSGi is about bundles and services. Bundles provide modularization and encapsulation for components. A bundle is also a deployable unit, which can be installed and removed at runtime. Bundles can register services, which can be looked up in the service registry and then used by other bundles. OSGi can be deployed on a wide range of devices from sensor nodes, home appliances, vehicles to high-end servers.

One of the weaknesses of OSGi (addressed in this paper) is that it defines how services “talk” to each other from within a single Java Virtual Machine (JVM). Thus, the way how the services interact to each other is constrained to an OSGi container where they run (local communication). In today’s IT world, distributed systems have been used rapidly in business applications. Thus, a lack of support of distribution is a severe hindrance for further use of OSGi in business applications because it does not allow external systems to access OSGi services remotely [7]. This holds true especially for enterprise systems, as nowadays OSGi grows more and more from its original roots (viz., embedded systems) into a Java platform for enterprise system. Simple evidence of this fact is that OSGi’s Embedded Systems Expert Group was discontinued, while its counterpart – OSGi’s Enterprise Systems Expert Group – is still “alive and kicking”. Therefore, our main goal was to add distribution

capability to OSGi in order for OSGi to be more applicable to enterprise systems. Toward this goal, in our previous paper [13] we proposed an approach, where distribution is enabled by exposing OSGi services as Web services, which is done by creating a “middleware” bundle that adapts OSGi services to become Web services. Figure 1 gives an overview of our approach.

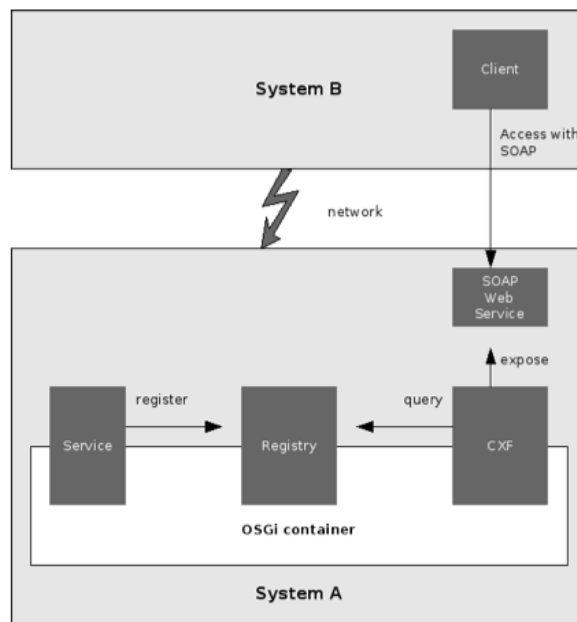


Figure 1. Architecture of our approach [13]. (System B is an external system.)

As middleware, we selected Apache CXF [2]. CXF is an enterprise service bus (ESB) that helps to develop services using different frontend programming APIs on different protocols, which in their turn use Web services defined by WSDL contract with SOAP bindings over HTTP. CXF is divided into multiple units called endpoints. We selected CXF for providing OSGi with distribution capability because being open source, CXF has no license fee. Furthermore, CXF itself can be represented as a set of bundles.

The rest of the paper is organized as follows. In Section II we provide an overview of the related work. In Sections III – VII we supply details on our approach. Given the background from the previous sections, in Section VIII we give an example of how to use our approach. In Section IX

we present the implementation of our approach. In Section X we make conclusions and outline the future work.

II. RELATED WORK

A number of extensions – e.g., OpenSOA, Redistributable OSGi (R-OSGi) [6], Distributed OSGi (D-OSGi), IBM Lotus Expeditor, Eclipse Communication Framework and Newton Framework – were done to allow services to “talk” with each other across multiple JVMs. The goal of all these extensions was to add distribution capability to OSGi, thus enabling a service running in one “local” OSGi container to invoke a service running in another, potentially remote, OSGi container. While meanwhile distribution has become part of D-OSGi, it lacks features like asynchronous messaging. This is, however, possible with our approach by utilizing both the appropriate CXF features and the extension that we implemented.

There also exists J2ME Web Service Specification [8], which extends OSGi with Web services functionality. This specification is widely used in embedded systems. Embedded systems are losing their original meaning, which referred to small computational isolated (stand-alone) systems that give functional support for devices that do not fit to the definition of a computer [9]. Today we can define an embedded system as a micro-processed device, thus programmable, which uses its computing power for a specific purpose [10]. However, the scope of the specification includes only how to expose remote services. It is not specified how external systems can access the services from embedded systems.

This paper extends our previous paper [13] in the following ways:

- It supplies more details on our approach (see Sections III, IV and VI).
- It shows an example of using our approach (see Section VIII).
- It implements a flight information system to demonstrate how our approach can be used in business applications (see Section IX).

III. REGISTERING OSGi SERVICES

Since a CXF bundle and an OSGi bundle exporting Web services are completely decoupled, we cannot rely on their installation order. In particular, the CXF bundle can be installed both before and after the OSGi bundle. Therefore, there are two cases to consider.

First, when the CXF bundle is installed before the OSGi bundle, the CXF bundle can register a service listener to get notified when the OSGi bundle is installed. This listener can be associated with a filter expression.

Figure 2 shows an example of how to register a service listener. Here `context` is an instance of the `BundleContext` class; it is provided during the CXF bundle activation. As can be seen, the `addServiceListener` method of the `BundleContext` class takes two parameters. The first parameter is an instance of a class implementing the

`ServiceListener` interface; this instance will be used by the CXF bundle as a callback to create an endpoint. The second parameter is a filter expression, which specifies that the CXF bundle gets notified about the service registration if and only if OSGi services have the `expose.service` property set to `true`.

```
BundleContext context = ... ;
context.addServiceListener          (cxflistener,
"(expose.service=true)");
```

Figure 2. Registering a service listener, which listens to Web services.

The service listener can use a `ServiceReference` instance to have more information about the OSGi services. In both situations, the `ServiceReference` instance can be used to acquire the necessary service information, e.g., the name of the service interface used by the service registry and an instance of the class implementing the specified interface. Figure 3 shows how to fetch this information.

```
ServiceReference ref = ... ;
Class iface = (Class) ref.getProperty("expose.interface");
String url = (String) ref.getProperty("expose.url");
Object instance = context.getService ( ref );
```

Figure 3. Accessing service properties.

Second, when the CXF bundle is installed after the OSGi bundle, the CXF bundle can query the service registry for OSGi services. Figure 4 shows an example of such a query. As can be seen, the `getServiceReferences` method of the `BundleContext` class takes two parameters. The second parameter is again the filter expression, whereas the first parameter specifies the service interface. However, since the CXF bundle cannot know the service interface in advance, a value `null` is passed to the method to specify that all (registered) OSGi services should be matched against the filter expression.

```
ServiceReference[] refs = context.getServiceReferences(
null , "(expose.service=true)");
```

Figure 4. Querying for Web services.

IV. EXPOSING OSGi SERVICES AS WEB SERVICES

As said above, since the CXF bundle can be installed both after or before the OSGi bundle as well as removed after a certain time, we have to enforce a loose coupling between the two bundles but still enable communication between them. There are three approaches to this:

- Extender model
- Listener model
- Whiteboard pattern (which we follow).

A. Extender Model

In the extender model, the CXF bundle can register a service using a `BundleContext` instance to get informed when new bundles are installed. The CXF bundle can react

on these events and inspect the content of the new bundles. The same approach can be used by the OSGi bundle to inform the CXF bundle about Web services it wants to export. CXF could define the location of a configuration file. This file would contain the Web services and the necessary service properties like a service interface, a port number and a URL.

Figure 5 shows an example of such a configuration file. The CXF bundle would check if the OSGi bundle contains this file and then use the file during the configuration.

```
<service
name="ExampleService"
interface="com.example.ExampleService"
class="com.example.ExampleServiceImpl"/>
<soap/>
<http port="8080" cont ext="/exampleService" />
</service>
```

Figure 5. Configuration file.

B. Listener Model

In the listener model, the CXF bundle itself can register a service that provides the necessary functionality for exporting Web services. For example, this service could have a method `exportService`; the service information like a service interface, a port number and a URL would be passed to that method. The OSGi bundle would fetch the CXF service from the service registry and call the method with the corresponding parameters.

C. Whiteboard Pattern

With the Whiteboard pattern [3], the OSGi bundle can register itself Web services. In addition to the service registration, the OSGi bundle can provide additional service properties (e.g., `expose.interface`, `expose.port` and `expose.url`), which contain the necessary configuration information. The CXF bundle can use the service registry to fetch the Web services.

D. Evaluation of Approaches

The extender model requires that all configuration information to be specified in a configuration file. It is therefore necessary that all this information is available when the OSGi bundle is created. The OSGi bundle is not able to provide additional information or change existing one during the deployment. Moreover, all (exported) Web services have to be listed in the configuration file. Thus, bundles are not able to inform the CXF bundle about new services after the deployment. Therefore, we rejected this approach.

The Whiteboard pattern has been originally developed as an alternative to the listener model. Both can be used for the configuration of the OSGi and CXF bundles. Both ensure that the bundles are completely decoupled from a specific CXF API. No dependencies on CXF classes or packages exist and the bundles do not need to be linked at compile time. However, the extender model analyzes the content of the bundles and uses the bundle events to enable export of new services, whereas the Whiteboard pattern uses the OSGi

Service Layer to enable communication between the bundles.

Moreover, due to the dynamic nature of OSGi (i.e., due to the fact that bundles can be removed at runtime), bundles cannot assume the continuous existence of the CXF service. Rather, they need to monitor the service registry in order to check for the CXF service.

When using the Whiteboard pattern, the configuration information is embedded into the service properties. The OSGi bundle can change these properties at runtime. That is, the OSGi bundle can change the configuration at deployment time and afterwards when needed. If the OSGi bundle wants to stop some service from being exported, it can just remove that service from the service registry and thereby inform the CXF bundle. The OSGi bundle is therefore able to control the export at runtime. Furthermore, the service properties can be used as filters when the service registry gets browsed by the CXF bundle for exported services. For example, the CXF bundle could query for all OSGi services that need to be exported with SOAP or at a specified port. This querying is, however, not possible when the configuration information is embedded into a configuration file as it is done in the listener model. Therefore, we also rejected the listener model and selected the Whiteboard pattern instead.

V. CONFIGURING WEB SERVICES

As said above, we decided to use the Whiteboard pattern for extending OSGi with distribution capability. When using this pattern, the configuration mechanism relies on embedding the necessary configuration information into the service properties while registering a Web service. Since we did not want to change the code, we decided to provide the configuration information during the service registration. There are three approaches to this:

- Java properties
- Declarative services Specification
- Spring-OSGi (which we follow).

A. Java Properties

Figure 6 shows an example of how to register a Web service using Java properties and associate the Web service with the service properties during this registration. The CXF bundle will listen to OSGi services that have a property `expose.service=true`. The service properties are stored in `java.util.Dictionary`, which is then passed as a parameter during the service registration.

```
ExampleService service = new ExampleServiceImpl();
Dictionary dict = new Hashtable();
dict.put ("expose.service", true);
dict.put ("expose.interface", ExampleService.class);
dict.put ("expose.url", "http://localhost:8080/exampleService");
```

Figure 6. Registering a Web service using Java properties [13].

B. Declarative Services Specification

The intention of Declarative Services Specification (DSS) [11] is to ease the use of the OSGi Service Layer.

Figure 7 shows a code example of how to register a Web service using DSS.

```
<?xml version="1.0" encoding="UTF-8"?>
<component name="ExampleService">
  <implementation
    class="com.example.ExampleServiceImpl"/>
  <property name="expose.service">true</property>
  <property name="expose.interface">
    com.example.ExampleService
  </property>
  <property name="expose.url">
    http://localhost:8080/exampleService
  </property>
</service>
<provide interface="com.example.ExampleService"/>
</service>
</component>
```

Figure 7. Registering a Web service using Declarative services Specification [13].

C. Spring-OSGi

This approach is similar to DSS. The key difference is that Spring-OSGi [12] itself defines a service registry like OSGi does. This registry is managed by `BeanFactory`. In particular, OSGi services can be searched and registered by `BeanFactory`; `BeanFactory` also makes services applicable to dependency injection. Figure 8 shows a code example of how to register a Web service using Spring-OSGi. Here `exampleService` is a normal Spring bean that will act as the instance of the service. An XML element `<osgi:service>` publishes this service in the service registry by referencing it and embeds additional information like the necessary configuration for CXF into the service.

```
<?xml version="1.0" encoding="UTF-8"?>
<bean id="exampleService"
  class="com.example.ExampleServiceImpl"/>
<osgi:service ref="exampleService">
  <osgi:interfaces>
    <value>com.example.ExampleService</value>
  </osgi:interfaces>
  <osgi:service-properties>
    <prop key="expose.service">true</prop>
    <prop key="expose.interface">
      com.example.ExampleService
    </prop>
    <prop key="expose.url">
      http://localhost:8080/exampleService
    </prop>
  </osgi:service-properties>
</osgi:service>
```

Figure 8. Registering a Web service using Spring-OSGi [13].

D. Evaluation of Approaches

The use of Java properties is the simplest approach; it embeds the configuration information directly into Java classes. Another advantage of this approach is that it allows for full control over the service properties and thus, it can be

used if, e. g., some values need to be calculated at runtime. However, because of the dynamic nature of bundles, the state of the services have to be tracked all the time. Therefore, we rejected this approach.

Instead of “hard-coding” the necessary logic for the service registration and then the tracking of the service state, DSS allows us to define this declaratively. Bundles that want to publish or use Web services define their intention in a configuration file that is then processed by DSS. However, DSS can be viewed as a hybrid approach that combines the Whiteboard pattern with the extender model. Since the configuration get supplied through the extender model, DSS inherits all the drawbacks of the extender model. Therefore, we also rejected this approach and selected Spring-OSGi instead.

VI. ANALYZING SERVICE INTERFACES

The information required by CXF is necessary to export a Web service registered in the service registry. This information is either specified as the service properties during the service registration or deduced from the service interface. The frontend that should be used by CXF (either JAX-WS or simple) can be determined by checking if the service interface is annotated. For example, when using the JAX-WS frontend, the service interface is annotated with `WebService`. Figure 9 shows an example of how to determine if this annotation is present.

```
Annotation [] as = iface.getAnnotations ();
for ( Annotation a : as )
{
  if (a.annotationType().equals(WebService.class))
    { // use JAX-WS frontend
    }
}
```

Figure 9. Using annotations.

In addition, CXF supports the use of different data binding frameworks. As with the frontend, the data binding used by CXF (either JAXB or Aegis) can be determined by checking if the classes passed as method parameters are annotated.

VII. ACCESSING WEB SERVICES WITH CLIENT FROM EXTERNAL SYSTEMS

CXF is divided into multiple units called endpoints. The communication over a network often takes place in a heterogeneous environment where some endpoints may share the same set of technologies whereas others may use a different set. When using CXF, the class libraries to use different technologies are located at different machines. Therefore, when services want to change the used technologies, only the libraries on the corresponding machines have to be updated. Other services are not affected.

To create an endpoint, CXF can fetch the configuration information from the service properties or deduce it from the service interface. Figure 10 shows how this information can be passed to CXF. The `ServerFactoryBean` class is

used to configure an endpoint. The service interface, the URL and the service instance are passed to a `ServerFactoryBean` instance. The `getFactory` method of the `ServerFactoryBean` class returns either a `JaxWsServerFactoryBean` or `ServerFactoryBean` instance depending on the frontend (either JAX-WS or simple). Similarly, the `getDataBinding` method returns either a `JAXBDataBinding` or `AegisDatabinding` instance depending on the data binding (either JAXB or Aegis).

```
ServerFactoryBean factory = getFactory(frontend);
factory.setServiceClass (( Class ) iface );
factory.setAddress ( url );
factory.getServiceFactory(). setDataBinding(
getDataBinding(databinding));
factory.setServiceBean (instance);
Server server = factory.create ();
```

Figure 10. Passing configuration information.

After fetching all necessary configuration information (i.e., a service interface, a port number and a URL), CXF becomes responsible for creating an endpoint based on this information using a method `create` of a class `Server`. The endpoint is published at the specified URL and can be accessed using SOAP [4].

In addition to exposing OSGi services to external systems, these systems require to access endpoints. There are two approaches to accessing OSGi services remotely:

- Creating a remote API
- Creating a proxy (which we follow).

A. Creating Remote API

To enable bundles to access (remote) endpoints, CXF could offer a special API that encapsulates the necessary logic. A service that is to be used by the bundles would then be registered in the service registry. Figure 11 shows a code example of how to create a remote API.

```
RemoteEndpoint re=new RemoteEndpoint();
re.setAddress(url);
re.setDataBinding(getDataBinding(databinding));
Object result=re.callMethod("methodname", "parameter");
```

Figure 11. Creating a remote API [13].

B. Creating Proxies

CXF provides a class `ClientProxyFactoryBean` to create a proxy [5]. This proxy will implement the service interface. Thus, it can be casted to a variable declared as an instance of the service interface. The proxy forwards method calls to the (remote) endpoint, waits for the result, and passes the result to the caller. This way we have both an elegant and a flexible way to access Web services remotely.

Figure 12 shows a code example of how to create a proxy. The information about the service interface, the URL and the data binding are passed to this proxy.

```
ClientProxyFactoryBean factory=new
ClientProxyFactoryBean();
factory.setServiceClass(iface);
factory.setAddress(url);
factory.getServiceFactory().setDataBinding(
getDataBinding(databinding));
Object proxy = factory.create();
Dictionary props = . . . ;
context.registerService(serviceClass.getName(), proxy,
props);
```

Figure 12. Creating a proxy [13].

After the proxy has been created, it gets registered in the service registry. During this registration, additional service properties can be attached to the proxy. These properties can be used as a filter expression when bundles are querying the service registry for specific services. Figure 13 shows an example of a query that returns all services that represent endpoints located on a server `www.company.com`.

```
ServiceReference[] refs = context.getServiceReferences(null,
"&(service.is_remote=true)
(service.host=www.company.com)");
```

Figure 13. Querying for specific services.

C. Evaluation of Approaches

Creating a remote API is simple. However, this approach introduces several drawbacks. After the bundle has been removed, all the information specified in the service interface is lost. This interface defines which methods are available and which parameters the methods require. Hence, the service interface represents the contract between a server and a client. By embedding the method name and parameters into a method call, we have to ensure that the specified values conform to the service interface. Moreover, depending on the used frontend, the service interface may specify additional semantics. For example, some methods may use asynchronous call semantics and require the registration of a callback method. However, the remote API would have to be aware of all these possibilities. Therefore, we rejected this approach and selected to create a proxy instead.

VIII. EXAMPLE

To demonstrate our approach, let us consider `SimpleService`.

Figure 14 shows an example of the service interface. `SimpleService` will be accessed by a client through the service interface, which is by definition separate from the service implementation. This separation enables changing the service implementation without changing other services.

```
package com.xyz.simple;
public interface SimpleService {
    public String getName();
}
```

Figure 14. SimpleService interface.

Figure 15 shows an example of the service implementation.

```
package com.xyz.simple;
public class SimpleServiceImpl implements SimpleService {
    public String getName() {
        return "My Name is SimpleService";
    }
}
```

Figure 15. SimpleService implementation.

Figure 16 shows an example of an OSGi bundle, which exposes SimpleService as a Web service using the Java properties approach (see Section V).

```
package com.xyz.simple;
import org.osgi.framework.BundleActivator;
public class SimpleBundleActivator implements BundleActivator {
    public void start(BundleContext context) throws Exception {
        System.out.println("Starting SimpleBundle");
        SimpleService service = new SimpleServiceImpl();
        java.util.Properties props = new Properties();
        props.put("expose.service", true);
        props.put("expose.interface", SimpleService.class);
        props.put("expose.url", "http://localhost:8080/simpleservice");
        context.registerService(SimpleService.class.getName(), service, props);
    }
    public void stop(BundleContext context) throws Exception {
        System.out.println("Stopping SimpleBundle");
    }
}
```

Figure 16. OSGi bundle.

Figure 17 shows an example of the client (i.e., a CXF bundle, which accesses SimpleService). The client listens to SimpleService at the endpoint URI: "http://localhost:8080/simpleservice".

```
public class SimpleBundleActivator implements BundleActivator {
    public void start(BundleContext context) throws Exception {
        CxfOsgiUtils.proxyRemoteEndpoint(context, SimpleService.class, "http://localhost:8080/simpleservice");
        ServiceReference ref = context.getServiceReference(SimpleService.class.getName());
        SimpleService service = (SimpleService) context.getService(ref);
    }
}
```

Figure 17. CXF bundle.

IX. FLIGHT INFORMATION SYSTEM

To prove the feasibility of our approach, we implemented a flight information system (FIS) using our approach. The purpose of the FIS is to inform passengers waiting at the airport about flight changes made by airlines. Figure 18 gives an overview of the FIS.

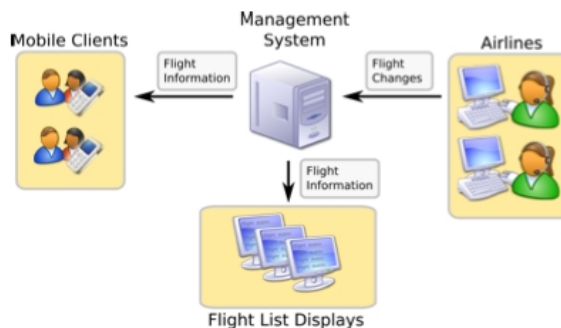


Figure 18. Architecture of flight information system.

The FIS consists of the following components:

- Management system
- Airline interface
- Flight list display system
- Mobile client.

The airline interface lets airlines to change information about scheduled flights. For example, an airline can change the aircraft type to a smaller one if the currently scheduled plane cannot be filled or the airline can change the departure time if the currently scheduled plane is delayed due to weather conditions. All these changes are broadcasted to the flight list display system and the mobile client by the management system. Figure 19 shows GUI of the airline interface.

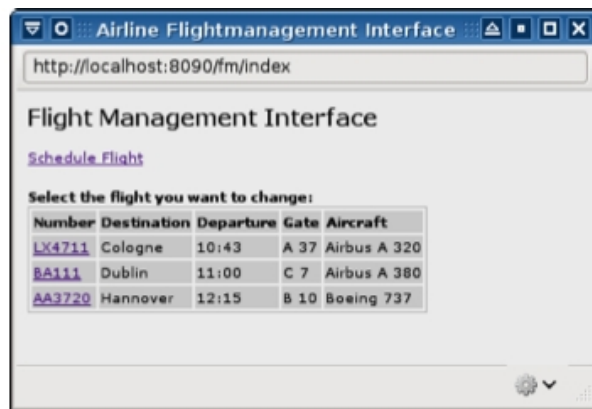
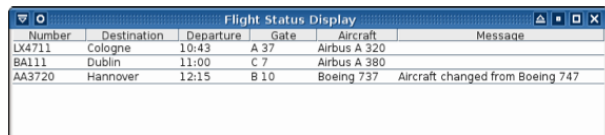


Figure 19. GUI of airline interface.

The flight list display system sends updated flight information (including current departure time, gate, aircraft type and flight status: on-time, delayed or cancelled) to flight list displays, which are spread over the airport. Figure 20 shows GUI of the flight list display system.



Number	Destination	Departure	Gate	Aircraft	Message
LX4711	Cologne	10:43	A 37	Airbus A 320	
BA111	Dublin	11:00	C 7	Airbus A 380	
AA3720	Hannover	12:15	B 10	Boeing 737	Aircraft changed from Boeing 747

Figure 20. GUI of flight list display system.

In addition, the mobile client lets passengers receive updated flight information from the comfort of their mobile phones. Figure 21 shows GUI of the mobile client.



Figure 21. GUI of mobile client.

As can be seen, the central component of the FIS is the management system. This component dispatches flight changes it receives from the airline interface to all interested parties – i.e., the flight list display system and the mobile client. The management system was implemented as a Java Business Integration (JBI) application. All other components around the management system – i.e., the airline interface, the flight list display system and the mobile client – were implemented as OSGi bundles. Therefore, the FIS is served as an example of how OSGi containers can communicate with an external (non-OSGi) system in a distributed manner.

X. CONCLUSION AND FUTURE WORK

We have proposed an approach to extending OSGi with distribution. Our approach does not require making changes to OSGi. Rather, it encourages using the concept of bundles.

OSGi was originally targeted towards embedded systems. Therefore, the main benefit of adding distribution capability to OSGi is to make OSGi more applicable to enterprise systems, which usually require remote communication.

We have added distribution capability to OSGi using CXF. CXF enables external systems to invoke (registered) OSGi services. Bundles can define which of the services can be accessed by external systems. CXF is used to create

(remote) endpoints and helps to utilize different technologies when needed. Our approach ensures a loose coupling between CXF and OSGi bundles. OSGi services do not need to be aware of the distribution or how they are exposed to external systems. The only needed interface is the OSGi Service Layer. Furthermore, bundles can access (remote) endpoints. This access is fully transparent to the bundles. A proxy delegates to the endpoints and gets registered in the service registry in order to be used by the bundles. In addition, we have used our approach to implement the flight information system.

In the future, we are going to make performance evaluation of our approach.

ACKNOWLEDGMENT

Irina Astrova's work was supported by the Estonian Centre of Excellence in Computer Science (EXCS) funded mainly by the European Regional Development Fund (ERDF). Irina Astrova's work was also supported by the Estonian Ministry of Education and Research target-financed research theme no. 0140007s12.

REFERENCES

- [1] OSGi – The Dynamic Module System for Java, <http://www.osgi.org>, last access: September 21, 2012.
- [2] Apache Software Foundation: CXF pages, <http://cxf.apache.org>, last access: September 21, 2012.
- [3] Kriens, P., Hargrave, B.J.: Whiteboard pattern, http://www.osgi.org/documents/osgi_technology/whiteboard.pdf, last access: September 21, 2012.
- [4] World Wide Web Consortium. SOAP Version 1.2. <http://www.w3.org/TR/soap12>, last access: September 21, 2012.
- [5] Gamma et al., Design Patterns, Addison-Wesley, 1995.
- [6] Swiss Federal Institute of Technology (ETH) Zurich: R-OSGi pages, <http://r-osgi.sourceforge.net>, last access: September 21, 2012.
- [7] Astrova, I., Koschel, A., Roelofsen, R., Kalja, A.: Evaluation of the Applicability of the OSGi Service Platform to Future In-Vehicle Embedded Systems. In Proceedings of the 2nd International Conferences on Advanced Service Computing (SERVICE COMPUTATION), IARIA, pp. 202–207, 2010.
- [8] Sun Microsystems. J2ME Web Service Specification. <http://jcp.org/en/jsr/detail?id=172>, last access: September 21, 2012.
- [9] Janecek, J.: Efficient soap processing in embedded systems. In Proceedings of the 11th IEEE International Conference on the Engineering of Computer-Based Systems (ECBS), pp. 128–135, 2004.
- [10] Wolf, W.: Computer as Components: principles of embedded computing system design. Morgan Kaufmann, 2001.
- [11] Cervantes, H., Hall, R.: Automating Service Dependency Management in a Service-Oriented Component Model, http://www.osgi.org/wiki/uploads/Links/AutoServDependencyMgmt_byHall_Cervantes.pdf, last access: September 21, 2012.
- [12] Interface21 Inc: Spring-OSGi, <http://www.springframework.org/osgi>, last access: September 21, 2012.
- [13] Roelofsen, R., Bosschaert, D., Ahlers, V., Koschel, A., Astrova, I.: Think large, act small: An approach to Web Services for embedded systems based on the OSGi framework. In Proceedings of the 1st International Conference (IESS), pp. 239–253, 2010.

Towards Online Engagement via the Social Web

Ioannis Stavrakantonakis, Andreea-Elena Gagi, Ioan Toma and Dieter Fensel

Semantic Technology Institute (STI) - Innsbruck
University of Innsbruck
Innsbruck, Austria
{firstname.lastname}@sti2.at

Abstract—The blossom of Web 2.0 has created new opportunities for the enterprises. Web users are disseminating more information than ever before about their interests, their experiences with products and services, their thoughts and whatever interacts with them in a daily basis. The challenge for the enterprise is to exploit and transform this social media explosion into added value on the business plan. The enterprises in the new Web 3.0 era should have the utilities to disseminate messages in a multi-channel attitude and to listen to the discussions around their products and services. Furthermore, they should be able to take action and interact with these discussions in a productive way for the internal improvement of the organization but also for establishing strong bonds with the users. The web users for an enterprise could be existing customers or potential customers, average users or advocates. This paper intends to discuss the characteristics and the nature of the online engagement with the customers in the scope of business intelligence on the Web.

Keywords-Social Web; engagement; social media; Web 3.0

I. INTRODUCTION

Web 2.0 has radically changed our communication possibilities. More and more communication has been freed from the geographic barriers that formerly limited their speed and expansion. Active participation and interaction of users have created a new platform for people to communicate with each other [4]. This platform relies primarily on the actions and contributions of users to create “a rich, lively, useful, and enjoyable space which draws people back again and again” [7]. Considerable bargaining power has been shifted from the supplier to the consumer. Tourism organizations and enterprises, and particularly travel agencies and hotels, have been seriously challenged by this shift or “consumer revolution” [10], but at the same time enormous opportunities have been opened up. For tourism organizations the internet has become one of the most important marketing communication channels [19].

However, the boom of the internet, dubbed as “the growth of the multi-channel monster” by [15], has also raised questions about marketing, distribution channels, business management and efficient marketing research in the tourism sector [13]. Organizations of all sizes, commercial and not-for-profit, regularly face the challenge of communicating with their stakeholders using a multiplicity of channels, e.g. websites, videos, public relation activities, events, email, forums, online presentations, social media, mobile applications, and recently structured data. The social

media revolution has made this job more complicated since the communication channels grow exponentially, shifting from a mostly unilateral “push” mode (one speaker, many listeners) to an increasingly fully bilateral communication, where individual stakeholders (e.g. customers) expect one-to-one communication with the organization. Moreover, the contents of communication become more and more granular and increasingly dependent on the identity of the receiver and the context of the communication.

Organizations need an integrated solution that provides management and execution of communication goals in a mostly automated fashion, with costs equivalent to mass-media communication, along with the granularity of individual experts, and at the pace of real-time social media. We are aiming to mechanize important aspects of these tasks, allowing scalable, cost-sensitive, and effective communication for small-or-medium sized enterprises (SME) and comparable organizations for which information dissemination is essential but resources are significantly limited. Additionally, it may also help intermediaries such as marketing agencies to extend their business scope by increasing the cost-effectiveness ratio. The current paper presents the concept of online engagement, providing a description of the concept in general, as well as the methodologies of applying it using tools to listen and respond to customers’ needs and demands.

The structure of the paper is the following. Section II defines the online engagement in the market ecosystem between the enterprises and the customers. Section III presents the objectives of the customer engagement for the enterprise. Section IV presents the spiral of online communication where online engagement is a major aspect. Section V focuses on the use cases of online engagement. Section VI discusses the related work in this field, while Section VII provides conclusions and directions for future work.

II. ONLINE ENGAGEMENT FOR SOCIAL ENTERPRISES

Customer engagement is not a novel concept for the marketing community. On the contrary, it is the main objective of any enterprise that has a long-term roadmap in the market and aims in the further development of the organization. It is hard to find a specific definition for engagement as it is dynamically changing through the decades and it has been used to refer to anything from what the consumers feel when they see an ad, the degree of interest, to the way the consumers will respond to

advertising. In some cases, engagement has been considered to represent all of the above, plus many other qualities. One notable definition for engagement has been given in 2006 by the Advertising Research Foundation: “Engagement is turning on a prospect to a brand idea enhanced by the surrounding context”. Similarly, James Speros, the Chief Marketing Officer (CMO) at Ernst & Young defines engagement as “all about making it relevant to the consumer”. Engagement has been associated with a wide range of terms, including “involvement”, “experience”, “connection”, “wantedness”, “resonance”, “relationship”, and “stickiness”. The creation of a “one size fits all”, universal definition for engagement seems unlikely, particularly in the ever emerging environment of Web 3.0. However, the semantics behind the engagement remain the same: *the enterprise and the customer are engaged in a long-term relation, which is beneficial for both sides.*

The current paper aims to discuss the engagement procedure in the new era of business, in which enterprises should be active in the Social Web sphere, understand the opportunities for them in this field and adopt dissemination strategies in the social media and networks [5]. The active enterprises in social media are aware of the social Web 2.0 channels of dissemination and are willing to use them to reach their audience in a more efficient and scalable way. The exploitation of the social media for dissemination purposes is the first and very important step for an organization in the engagement spiral. Furthermore, the enterprise is not the only entity that is disseminating information in the Web; individuals are more talkative than ever before in the Web ecosystem. People are expressing their thoughts and views about anything including products, services and brands. The enterprise should be able to listen to these voices and input these data in the infrastructure of communication with the customers. New customers and new audiences are present in the Web sphere and potentially talking about enterprises’ products and services. The enterprise should not remain silent, but grab the opportunity and interact with these people in a relevant, meaningful and interesting way. The online engagement is leveraging the online communication between the enterprise and the audience in a relation, which will drive the design of the new products and services. The new business era is more client centric than it used to be in the last few years.

III. OBJECTIVES OF ENGAGEMENT

Customer engagement is formed on the premise of listening and responding to customer conversations. On the other hand, online customer engagement is inherently different than offline engagement as the nature of the customers’ interactions with a brand and other customers in the online space is defined by the medium used to converse (i.e. the online platform or service employed). Discussion that takes place on forums or using Facebook cannot be replicated by an offline medium. As such, the methods employed for offline engagement can no longer be applied online. However, two main questions must be posed. First why should an enterprise apply these methods in the online space? Second, what are the benefits of online customer

engagement? The current section aims to shed light on the main benefits an enterprise can achieve in brand management, quality management and transactions increment.

A. Yield Management

The concept of “yield” refers to the financial and economic gains that can result from tourism [16]. Yield (revenue) management is mostly about maximizing the short term gain of an enterprise by combining segment pricing with statistical analysis. Achieving short-term increase of income is a valid target for a business entity; however, it is difficult to realize in a multi-channel world as hotels are confronted with a multitude of option that often come with their own constraints (e.g. price constraints on the offers) and which, in some cases, generate costs without guaranteeing actual income.

Many solutions to yield management are based on complex statistical methods and complex domain assumptions on how variation of the price can influence the number of bookings of a service. However, a multi-directional multi-channel approach must also rely on Swarm intelligence. Observing in real time the reaction of customers and competitors will be the key to achieve online marketing. Adopting your offer and your price dynamically in response to the behavior of your (online visible) environment will become critical to economic success.

B. Brand Management

The successful brand is the one that is connected seamlessly in the minds of the customers with an activity, an idea or a fact. In this respect, we could consider the case of the energy drink Red Bull. Due to the efforts of brand management it has been connected with the need of an instant boost to the energy and the performance of the human body. Connecting the philosophy of the enterprise’s products on something that happens in the real world is the best way to keep the business alive. Thus, it is crucial to be active in the new field in which the reputation of your brand is being shaped and affected in any positive or negative way. Social media has given individuals the opportunity to voice their opinions in ways that have not been available in the past. Spotting out the influencers [8] and the advocates of a brand would help the enterprise to understand the problems and the difficulties that the key customers are facing or what they like most about the products and the services. The brand is not only about the quality of the product; the brand is a step further, which reflects the total customer experience with the products and services.

In consumer marketing, brands provide the primary points of differentiation between competitive offerings [20]. The American Marketing Association proposes a company oriented definition, which describes the brand as a name, term, sign, symbol, or design, or any other feature that identifies the goods or services of one seller or group of sellers and differentiates them from those of competitors. Brands are key organizational assets strategically positioned in the market by offering features desired by consumers that

are distinct from competition [20]. If we embrace the assumption that brands are pivotal resources for generating and sustaining competitive advantage [1][14], proper management of brands value becomes essential for the enterprise's long-term economic success. Using social media a company can encourage the development of loyal and engaged customers while launching a new product, as well as help retain existing customers and create brand advocacy through word-of-mouth.

C. Quality Management

Besides the support and facilitation of the different marketing processes, the engagement approach could be realized as a valuable source of input and feedback from the end users of the products and services regarding the delivered quality. In the new customer-centric era the customer decides what quality is. Furthermore, by taking seriously the complaints of the customers, the product development team has a unique opportunity to improve the final product and the characteristics of it. The quality management in the different types of organizations is materialized and interpreted in different ways, e.g. for a hotel is the quality of the daily hospitality services while for a hardware company could be the quality of the next version of the end-product.

D. Transactions Augmentation

At the top layer of the objectives, as a logical consequence, the ultimate goal is the revenue growth via engaged customers. The enterprise should have a sound plan to monetize the online communication results. Research results obtained by [6] show that fully engaged customers deliver a 23% premium over the average customer in terms of share wallet and profitability growth. For instance, the engaged customers of a hotel would choose the same hotel in a future visit to the same place as they know that the hotelier would treat them in the best way due to their connection. Furthermore, a hotel chain would benefit from engaged customers as they would choose the hotels of the chain for their stay in any place in the world. Consequently, there is a plethora of opportunities to transform these relations into future transactions.

IV. SPIRAL OF ONLINE COMMUNICATION

The proposed online engagement approach is considered to be in the scope of the top layer at the Semantic Web stack, which is the "User Interface & Applications" as presented in [3]. This layer includes applications that are exploiting the languages and the technologies that have been developed in the Semantic Web (Web3.0) era. The vision is to establish the fundamentals for the future online communication paradigm on top of the semantic technologies in order to benefit from their power.

We start to take the view on interaction and communication, i.e., allowing other agents to post to us and us to post to communication pieces of others. Therefore, the concept of customer engagement comes to play. The spiral

of online communication refers to the potential infinite loop between the producer of a message and the receiver of it through one or more communication channels. This communication pattern is not only bi-directional but also it can be triggered by any of the stakeholders. The traditional communication pattern in the past, between an enterprise and the audience of potential customers, used to be a one-to-many relation: as the enterprise was disseminating messages and the customers were passively receiving them. In certain cases, means to reply to a user comment were available; however, means that would have allowed the enterprise to begin a conversation in real time with the customer were not. As mentioned, this particular aspect has been changed by the Social Web. Thus, the enterprise should establish a listening method in parallel to the dissemination of the messages in order to receive input from the customers even before the spread of any message or advertisement.

A. The listening phase

Social Media [11] is a term used mostly for web-based techniques of human-to-human communication that stresses the social, topical, and contextual relations between communicating individuals, allowing real-time interaction with a large, yet specific audience of partners. Social media sites have gained huge popularity in recent years, attracting millions of users, on different platforms, who consume and create content.

In order to assess the massive amount of user generated content produced by social media, specialized monitoring tools have been created. The social media monitoring process is the continuous systematic observation and analysis of social media networks and social communities. In essence, social media monitoring is executed using the following steps: data gathering (achieved through web crawling and direct access to the streams of the social media networks via APIs), data filtering (eliminating "noise", like irrelevant posts, duplicates or spam), data analysis (natural language processing algorithms and sentiment analysis are applied to identify key topics, influencers, detractors, as well as conversations a company should join) and data presentation (presentation of results in a way that is meaningful for the company and leads to actionable insights). The added value of social media monitoring is that it offers access to real customers' opinions, complaints and questions, at real time, in a highly scalable way.

Currently there is an enormous number of available Social Media Monitoring (SMM) tools on the market, thus making an educated choice about which tool to use has become increasingly difficult. Moreover, creating an evaluation framework for such tools has been a challenge for many reviewers and market research enterprises. For instance, Forrester [9] assesses tools based on three criteria: current offering (services and features offered), strategy (how they address enterprise-level needs) and market presence. Both [12] and [18] have tried to create more detailed evaluation frameworks that focus on the basic features of a social media monitoring tool, as well as on the technology and user interface features. According to the [18]

the main features that a tool should provide are the following:

- *Listening Grid*: The listening grid focuses on three main aspects: (1) the channels that are monitored (e.g. blogs and micro-blogs, social networks, video and image websites, etc.); (2) which countries and languages the tools provide support for; and (3) the topics relevant to the enterprise.
- *Analysis*: Having established a listening grid, the next step is to analyse the data and produce actionable reports and insights for the user of the tool.
- *Engagement*: The engagement concept refers to the ability of the tool to support reaction and response to the social media posts.
- *Workflow Management*: Workflow refers to the process of assigning, tracking and responding to social media streams, usually in a team environment in order to prevent double responses and missed opportunities.
- *Near real-time processing*: It is crucial for enterprises to follow up potential customers or customers' complaints, questions and thoughts well in time.
- *API*: The social media monitoring tool should provide an API solution in order to make feasible the integration of the social media monitoring with other tools (e.g. customer relationship management tools).
- *Sentiment Analysis*: Customer sentiments (which may be positive, negative or neutral) are determined using elements of computational linguistics, text analytics, and machine learning elements, such as latent semantic analysis, support vector machines, Natural Language Processing.
- *Historical data*: Access to previously captured data is required in order to compare the current metrics and reports related to the monitored topic with any previous state of it.
- *Dashboard*: The dashboard offers users graphical representation of the raw data in the form of charts, listings, and historical graphing of queries and phrases.
- *Export results*: In order to comply with their customers' needs, social media monitoring tools developers should enable users to download the results of their tool's analysis in different formats such as excel workbook or CSV format.

The aforementioned features were stressed out to be the most important for the social media monitoring tools. At the same time, these factors comprise the prerequisites for the listening phase implementation in the scope of an engagement framework.

B. The engagement phase

The Social Web must not be used only as a means for dissemination, a place to read, but as a place to publish and respond to user generated content in the most effective way. Therefore, four main requirements must be taken into

consideration: (1) we must provide a smooth integration of write and read activities in both respects; (2) we must ensure that we implement the process character of communication that is based on a chain of combined read-write processes to achieve interaction; (3) we must support cooperation based on online communication that allows engagement, transactionality and economic cooperation following successful online interaction; and (4) we should know where, when, who, why and what will communicate in response to the feedback that is being collected from the social media ecosystem.

The **first important requirement** for a proper communication tool support is to provide a smooth integration of write and read activities in both respects. That is, it must offer a) publication means for others in our publication channels as well as b) easy means to publish at external publication places of others.

The **second major requirement** for a proper communication tool support is to implement the process character of communication that is based on a chain of combined read-write processes to implement interaction. This requirement has several sub-features. The tool should be able to *trace* the history and state of a communication. For example, in a CRM system workflow and information sharing facilities must be provided to allow several employees to properly implement the sequential steps of a communication. Furthermore, this kind of tools should be able to support *multi-channel communication*. A communication may start with a tweet, a Facebook post, a private direct message, and further email communication, etc. The communication is not sequential and happens in different ways and mediums. In essence it is a parallelized interaction of various agents taking the role of a publisher or listener. Supporting *multi-agent communication*, where larger numbers of agents orchestrate a multi-directional communication process in parallel is another aspect of this requirement. The original basic Sender-Message-Channel-Receiver (SMCR) model of communication proposed by [17] is unidirectional. A sender sends a message through a channel to a receiver. The direction of the communication and the different roles are fixed. However, the model is inadequate for representing current online communication which is exponentially more complex than the initial telephone communications described by Shannon and Weaver. Agents interact and communicate in parallel, permanently alternating their role in these acts of communication or in Web2.0 terms, users are prosumers, i.e., consumer and producer of information. Therefore, we have adopted the *transactional model* of communication and its underlying premise that individuals are simultaneously engaging in sending and receiving messages [2].

Communication is potentially an infinite process, however, only when driven by an underlying purpose that uses it as a means to an end. The **third major requirement** is support cooperation based on online communication

allowing engagement and support transactionality and economic cooperation following successful online interaction.

Moreover, it is very important to define and specify the different characteristics of the engagement approach regarding the needs of an enterprise. This perspective of the engagement process comprises the **fourth major requirement**. Each single enterprise and organization has a different business plan and the nature of the offered services and products is unique. It is crucial to know where, when, why, what and who will communicate in response to the feedback that is being collected from the social media ecosystem. These parameters define a five-dimensional space in which all the engagement threads of the enterprise and customers can be mapped to. The administrator of the tool should be able to decide which type of interaction and medium (*where*) would be the best in order to fulfill the 3rd major requirement (transactionality); which are the time limits to form the reply (*when*); if it is necessary to take care of the message and initialize a discussion (*why*); the adapted content that should be presented in the context of the reply (*what*); and the appropriate person that could treat efficiently the upcoming discussion (*who*).

On top of these requirements there is an opportunity to automatize different activities with semantics and produce a scalable solution that could facilitate the communication processes of SMEs. The vision of the authors is to empower the SMEs to stay competitive in a market that moves towards the personalization perspective in offered services and products and human-centric marketing. The one-to-one communication between the enterprise and the customer (B2C) is not easily materialized by a SME as it needs a remarkable amount of resources.

V. USE CASE OF ONLINE ENGAGEMENT FOR SMEs

There is a myriad of potential applications in different business sectors and fields for the online-engagement paradigm. The current paper presents a use case of the concept in the SME's ecosystem and especially in the lodging business domain.

The hotelier of a small or medium-sized hotel should be able to exploit the power of the social media instead of being afraid to get involved. The added value of social media for the hotelier lies on four major pillars: a) the multi-channel dissemination; b) the quality management of the hotel's services; c) the development of the business plan regarding the feedback from the customers and d) the increase of transactionality (i.e. bookings) via the dissemination of call-to-action messages.

The plethora of communication channels that are available in the Web sphere should be exploited by the hotelier in the most efficient, productive and automatic way towards the development of the awareness and visibility around the hotel and the business activities of it (e.g. special events, excursions, restaurant, sport tournaments organization etc.). The need of a tool that could be able to

orchestrate the dissemination process of the hotel is more than crucial for a small or medium sized hotel, which is not able to hire a team of experts to take care of the social marketing of the hotel. The hotelier should be able to disseminate any information related to the hotel in an abstract way, decoupled from the communication channels.

The quality management of a hotel is a major aim of the owner as it is one of the crucial prerequisites for the success and perpetuation of the business. The appropriate tool could enable the hotelier to listen to the feedback in the social media (not only in review sites like TripAdvisor) from the customers and internally exploit this information for the improvement of the provided services. This approach guarantees the quality of the existing services and the total experience of the customers regarding their visit. Moreover, listening in real time could help the hotel manage and fix any problems that happen during the customers' stay and ensure that the feedback posted by these customers on social media channels is positive. For example, considering a hotel customer that is expressing on twitter his dissatisfaction for the hotel that he is currently staying in his trip regarding the hygiene of the room, the response of the hotelier should be instant and both online and offline. Thus, the hotelier should force an immediate investigation and reaction of the room service to his room in order to manage and resolve the issue. The response time in this example should be in terms of a few hours, the responsible group of responding and being assigned the issue is the room service and the place of response is offline, at the room of the customer.

The ultimate goal of any business is the increase of the economic transactions and the rise of profit. The engagement concept promotes the establishment of long-term relationships between the hotel and the customers. The engaged customers cannot only be transformed into repetitive customers and visitors of the hotel, but they can also become advocates as they influence positively their social circles. Personalization is the key aspect in this kind of relationships. The hotelier should be able to mechanize in a human-centric way the dissemination of information and the response to the online discussions in order to increase the number of bookings and customers.

VI. RELATED WORK

The proposed paradigm of engagement includes different technologies and concepts that are needed in order to build the described approach. The aim of the current paper is to define the fundamentals and the requirements of a complete and effective online communication framework that could empower SMEs to engage their customers and any potential customer. In this essence, there is not any similar approach available according to our knowledge. However, the various components of the engagement framework are in the scope of existing fields, like social media monitoring and semi-automatic matchmaking.

The SMM tools are soundly covering the listening phase that was previously mentioned in the scope of the online

communication and the engagement concept. A large part of the available SMM tools are described in the different reports and papers that have been already mentioned, like [9], [12] and [18]. The social media monitoring technics and findings are perfectly reusable and extendable under the umbrella of the engagement approach and will be taken in consideration during the design of the framework. However, this kind of tools does not support the engagement concept in the essence that was described in this paper and envisioned by the authors. The main objective of the application is to enable the SMEs to handle the multi-channel interaction with the customers in a semi-automatic way by employing reusable communication workflows, which address various specific patterns of dissemination and reaction.

VII. CONCLUSION AND FUTURE WORK

The engagement concept should be treated by the enterprises as an opportunity to build strong ties with their customers and turn them into advocates that will add positive value to the brand reputation via the online word of mouth. By being authentic, transparent, and operating with integrity, the enterprise could successfully engage their market and build a community of advocates who would spread their message virally. The challenges for bringing the engagement to the full potential are definitely the scalability of the possible solutions and the effectiveness of the approaches. In this paper, we tried to put the bases for the next step in online communication and specify the requirements of an effective engagement framework.

The future steps beyond this preliminary exploration of the online communication space will be the actual design of the framework that could address the described ideas and concepts. The reference architecture of the framework will be based on the aforementioned requirements and professional objectives of the small and medium sized enterprises. The design of the engagement framework will exploit the semantic web technologies and will integrate the existing knowledge in communication methods with the communication patterns that can be used in the context of the Social Web.

REFERENCES

- [1] D. Aaker, "Managing assets and skills: the key to a sustainable competitive advantage.", 1989.
- [2] D. Barnlund, "A transactional model of communication.", *Foundations of communication theory*, 1970, pp. 83-102.
- [3] S. Bratt, "Semantic web, and other technologies to watch.", *World Wide Web Consortium*, January 2007.
- [4] P. Casoto, A. Dattolo, P. Omero, N. Pudota and C. Tasso, "Accessing, Analyzing, and Extracting Information from User Generated Contents.", *Handbook of Research on Web 2.0, 3.0, and X.0*, 2008, pp. 312-328.
- [5] D. Fensel, B. Leiter, S. Thaler, A. Thalhammer and I. Toma, "Effective and efficient on-line communication", 2012.
- [6] J. Fleming, C. Coffman and J. Harter, "Manage your human sigma." *Harvard Business Review*, vol. 83(7), 2005, p. 106.
- [7] J. Freyne, M. Jacovi, I. Guy and W. Geyer, "Increasing engagement through early recommender intervention.", *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 85-92.
- [8] A. Galeotti and S. Goyal, "Influencing the influencers: a theory of strategic diffusion.", *The RAND Journal of Economics*, vol. 40(3), 2009, pp. 509-532.
- [9] Z. Hofer-Shall, "The Forrester Wave™: Listening Platforms, Q3 2010.", *Forrester Research*, 2010.
- [10] L. Huang, C. Yung and E. Yang, "How do travel agencies obtain a competitive advantage?: Through a travel blog marketing channel.", *Journal of Vacation Marketing*, vol. 17(2), 2011, pp. 139-149.
- [11] A. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media.", *Business horizons*, vol. 53(1), 2010, pp. 59-68.
- [12] H. Kasper, M. Dausinger, H. Kett, T. Renner, J. Finzen, M. Kintz and A. Stephan, "Marktstudie Social Media Monitoring Tools.", *Fraunhofer IAO Studie*, 2010.
- [13] S. Liu, "A theoretic discussion of tourism e-commerce.", *Proceedings of the 7th international conference on Electronic commerce*, 2005, pp. 1-5.
- [14] M. Louro and P. Cunha, "Brand management paradigms.", *Journal of Marketing Management*, vol. 17, 2001, pp. 849-875.
- [15] S. Mulpuru, H. Harteveldt and D. Roberge, "Five Retail eCommerce Trends To Watch In 2011.", *Reproduction*, 2011, pp. 1-8.
- [16] J. Northcote and J. Macbeth, "Conceptualizing yield: sustainable tourism management.", *Annals of Tourism Research*, vol. 33, 2006, pp. 199-220.
- [17] C. Shannon, "The mathematical theory of communication. 1963.", *MD computing: computers in medical practice*, vol. 14(4), 1997, p. 306.
- [18] I. Stavrakantonakis, A.E. Gagiou, H. Kasper, I. Toma and A. Thalhammer, "An approach for evaluation of social media monitoring tools.", *Common Value Management*, 2012, p. 52.
- [19] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, B.Y. Zhao, "Serf and turf: Crowdturfing for fun and profit.", *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 679-688.
- [20] L. Wood, "Brands and brand equity: definition and management.", *Management decision*, vol. 38(9), 2000, pp. 662-669.

Algorithms for Mapping RDB Schema to RDF for Facilitating Access to Deep Web

Wondu Y. Mallede, Farhi Marir, Vassil T. Vassilev

Knowledge Management Research Centre,

School of Computing, London Metropolitan University, London, U.K

wym0001@my.londonmet.ac.uk, f.marir@londonmet.ac.uk, v.vassilev@londonmet.ac.uk

Abstract— Semantic Web sets the standard for a universal and interoperable data representation that is not only readable to the naked eye but also to computers. The use of Uniform Resource Identifier (URI) and the capability to use description logic through semantic ontology languages makes semantic web the favoured framework to represent knowledge. Semantic Web standards play a vital role in making the existing relational database, which is locked behind in the “deep web”, available for computer processing. In order to map the relational database in its entirety, the methodology should not only map the data but also the domain specific knowledge. The algorithms and their implementation presented in this paper use the meta-data from the data dictionary to construct the initial semantic repository, while using domain specific knowledge during in-processing stage.

Keywords-Relational Database Mapping; Domain Specific Knowledge; Semantic Web; Data Mapping Algorithm

I. INTRODUCTION

The current web experience gives us a fairly abundant data. Using a few keywords and common search engines, it usually does not fail to return search results as well. With all its openness, the web gives anyone a chance to contribute ideas to be shared by the whole world about any topic. The web often feels like it is a mile wide, but an inch deep. How can we build a more integrated, consistent, deep web experience? [1].

The ANSI-SPARC (American National Standards Institute, Standards Planning And Requirements Committee) architecture for databases dictates the separation of the conceptual level from both external view (users) and physical level (files). The conceptual level consists of all the entities, their relationships and constraints that channel the data back and forth between the external and physical level. Mapping the conceptual level to semantic Resource Description Format (RDF) [12] makes the “deep web” re-surface for semantic interpretation and processing. The Semantic Framework helps narrow the gap between the “deep web” and the “surface” web.

Semantic Web is a framework which allows information to be represented not only structurally using suitable structural definitions (“data schema”), specific instances of them (“data”) and their use (“access rights”), but also semantically using a logical model which allows formal interpretation and sound logical inference about the information (“knowledge”). Relational data models on the other hand lack the capability to represent “knowledge” in

spite of their popularity. Since most KR (Knowledge Representation) mechanisms and the Relational Data Model are based on symbolic languages, the ability to represent and utilize knowledge that is imprecise, uncertain, partially true and approximate is lacking, at least in the base/standard models [9]. This lack of capability to represent and process knowledge while it is still in relational model is one of the challenges in Knowledge Management. This research focuses on the development and evaluation of algorithms to map the Relational Database (RDB) schemas to RDF in Semantic Web to allow access to deep web applications. The evaluation and validation of the developed mapping algorithms has been undertaken on a space project management domain-specific application.

Space program projects involve activities like observation, human space exploration, space launch and navigation, operational maintenance, etc. The range of terminologies, standards, unit measurements, and definitions will all be referring to the space domain ontology. The practical evaluation of the developed mapping algorithms has been undertaken on Oracle database system representing the space database schema which is composed of six major relational concepts- Documents, Risks, Non Conformances, Reviews, Actions, and Projects.

Semantic Web has a data model as part of its architecture that will be used as a repository to store ‘semantic data’- RDF. RDF is a format to store data in Semantic Web and will use RDF Schema (RDFS) [13] and Web Ontology Language (OWL) [14] to interpret the data. Data in RDF Schema not only has literals but also a semantic meaning attached to it. This meaning has different informal hierarchies and formal taxonomies in its knowledge domain (space-domain). This leads to the introduction of a consensually shared view of concepts called Ontologies. Ontology is a formal, explicit specification of a shared conceptualisation [10]. The specifications use relations, functions, constraints, and axioms to conceptualise the abstract model. 'Formal', in the ontology definition, refers to the fact that the expressions must be machine readable; hence, natural language is excluded [6]. RDF was designed for situations where Web data need to be processed and exchanged by applications, rather than being displayed for people. The ability to exchange data between different applications means that the data may be made available to applications other than those for which they were originally intended [11].

Semantic data models and frameworks help organise the knowledge about specific domain and share amongst systems. The models help to see the “semantic” from

different angles and refine the meaning by working on the ambiguity while reusing its commonality. When two (or more!) viewpoints come together in a web of knowledge, there will typically be overlap, disagreement, and confusion before synergy, cooperation and collaboration [1].

With Relational data on one side and RDF repository data on the other, the mapping algorithms have involved domain specific heuristic formulation to satisfy the accurate implementation of knowledge transfer. Different reasoning logic and rules are part of the mapping algorithms.

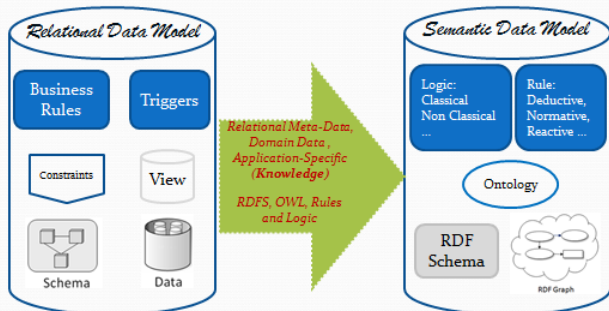


Figure 1. Relational to Semantic Mapping Model

The mapping implementation involves RDF, Web Ontology Languages (RDFS/OWL), RDF query language (SPARQL) and semantic rule languages on Oracle database systems. The research project focuses on how to take advantage of the already existing vast relational data to build a knowledge base in using Semantic Web Framework. In the current Semantic architecture RDF is a favoured data serialization format to represent and manage knowledge. The ultimate goal of the research is a formal heuristics-based methodology for mapping relational database to semantic knowledge base. Section II explores existing approaches on mapping relational databases to RDF and points out what is being consistently missed during mapping. Section III analyses the different levels of 'knowledge' with respect to mapping. Sections IV and V outline the mapping algorithms and their implementation respectively and finally Section VI summarises with conclusion and future work.

II. EXISTING APPROACHES

The World Wide Web Consortium (W3C) RDB2RDF Working Group (WG) has conducted a survey of current approaches for mapping of relational databases to RDF [8]. The report summarises the different mapping implementation, query implementation, application domain, mapping creation, mapping representation and accessibility.

The mapping implementation is either using a static Extract Transform Load (ETL) or a dynamic on demand query-driven implementation. The static data warehousing approach has its own drawback in reflecting the current data. The queries are run in periodic intervals without compromising the current performance using mapping rules. It also gives an opportunity to analyse the data with respect to validation rules. The dynamic approach, on the other hand,

costs a lot of performance time even though the outcome is a current reflection of what is in the relational database.

The query implementation follows two paths. The SPARQL-> RDF or the SPARQL-> SQL-> RDB path. SPARQL treats each RDF graph as a RDB table with three columns, ?subject, ?predicate and ?object. Each row corresponds to one tuple and the query result of SPARQL constitutes a table of RDF nodes [5].

Automatic mapping of RDB table to RDF class node and RDB column to RDF predicate leave behind most of the semantics of the data. Tools like Virtuoso RDF View [3] expanded the above notion to map RDB unique identifier (primary key) to RDF object and column values as RDF subject. Even though these automatic mapping tools could be used as a starting point, there is still a lot to do to analyse, refine and process the Semantic data.

The survey also suggested the use of pre-existing public ontology resources such as the National Centre for Biomedical Ontologies [15] or an automatic domain-specific mapping tool such as D2RQ [2] that also allows custom user mapping rules. This approach also helps to reduce the amount of redundant knowledge. In one of the projects [4] based on the Royal Commission on the Ancient and Historical Monuments of Scotland (RCAHMS), 1.5 million entities of the database are converted into 21 million RDF triples. Using the domain semantics-driven generation the size of the RDF dataset is reduced by 2.8 million.

A further "feature-based comparison" between the major mapping languages (Direct Mapping, eD2R, R₂O, Relational.OWL, Virtuoso, D2RQ, Triplify, R2RML, R3M) based on RDB2RDF WG report [8] also shows the different features of existing mapping languages [6]. The paper compares the mapping languages using four categories: *direct mapping*, *read-only general-purpose mapping*, *read-write general-purpose mapping*, and *special-purpose*.

What is being consistently missing from the existing mapping languages is the lack of "knowledge" consideration, which is not always explicitly represented and the use of "rules" and "logic". This "knowledge" can be *derived* from the explicitly represented relational model. It can also be *checked* using "application-specific predicates" and/or *executed* using "application-specific procedures/functions". The deficiencies of the existing mapping languages can be categorised into the following major levels of "knowledge".

1. Lack of using all the available "Relational Database Area Knowledge" and their variant meta-data combinations.
2. Lack of using "Domain Data Knowledge" like data-patterns (disjointness, symmetry, transitive chain, etc.)
3. Lack of using "Application Specific Knowledge" like "application-specific predicates" and "application-specific procedures/functions".

- Lack of using “rules” and “logic” to elicit different “application-specific predicates” which is the recommendation of the W3C RDB2RDF WG [11].

III. RELATIONAL DATABASE KNOWLEDGE LEVELS

The mapping algorithms for converting relational databases into Semantic Web repositories which we have developed account several different types of knowledge: related to the relational model itself (*relational model knowledge*), related to the data stored in the database (*domain data knowledge*), related to the use of data (*domain users knowledge*) and knowledge about the database application (*application knowledge*). The conversion of the relational database is performed in three subsequent stages: *pre-processing*, during which the semantic repository is created and structured, *in-processing*, which incrementally maps the relational data and *post-processing*, which modifies the generated semantic repository to account additional domain-specific knowledge. After mapping the relational database to semantic RDF repository, different semantic rules are also applied to analyse the domain.

The domain knowledge which is broadly divided as “*relational database area knowledge*”, “*domain data knowledge*”, “*domain users knowledge*” and “*application-specific knowledge*” is used as a resource to facilitate the mapping of relational database to semantic RDF.

- Application Specific Knowledge
- Domain Data Knowledge
- Domain Users Knowledge
- Relational Database Area Knowledge

The use of the existing meta-data and knowledge at different levels as listed above and the formulation of additional knowledge from the existing knowledge contributes to the efficient representation of relational databases in Semantic Web.

A. Relational Database area Knowledge

The relational database area knowledge is used to identify the tables and columns to be considered in the mapping as well as the database constraints and data type restrictions on table columns. The relational database consists of different relational objects that are grouped into relational schemas. The tables and columns contain the data that is going to be mapped. In relational database, constraints are used to keep the integrity of the data. The constraints need to be mapped together with the data to maintain the integrity after mapping.

The database uses data type restrictions to guarantee data integrity during storing, retrieving and processing operations. The standard SQL data types are considered during the database manipulation process. These SQL data types are also mapped using an equivalent semantic RDF data type.

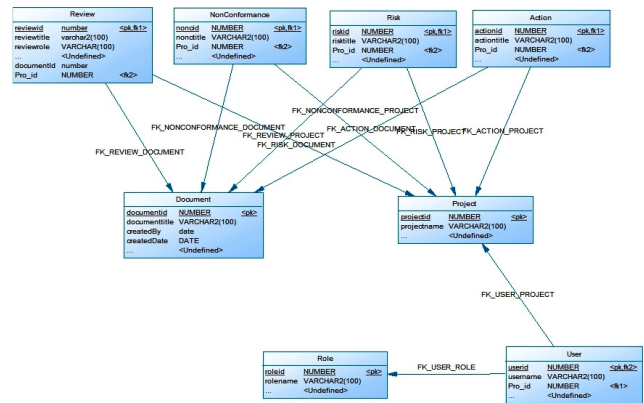


Figure 2. Relational Database Schema Overview

The relational database consists of database schemas (S) as a way of grouping relational objects. The database schema $S(T_1, T_2, \dots, T_n)$, where n is the number of relational tables T is referred as the ‘owner’ of all the database objects under the schema. The relational data is stored in tables $T(A_1, A_2, \dots, A_m)$, where A is the column attribute and m is the number of column attributes in the table. Each table consists of different column attributes A_1, A_2, \dots, A_m , where each column attribute has its own domain $dom(A)$ and range $ran(A)$.

The database constraints considered during mapping are primary key ‘pk’, foreign key ‘fk’, UNIQUE ‘unq’, NOT NULL ‘nn’, and CHECK ‘ck’ which are represented as $pk(T)$, $fk(T)$, $unq(A)$, $nn(A)$ and $ck(A)$, respectively.

Each table T is a set of tuples t_1, t_2, \dots, t_n , where n is the number of tuples in a table. Each tuple t is a set of values $\langle v_1, v_2, \dots, v_n \rangle$, where v_i is the value corresponding to column attribute A_i for current tuple $1 \leq i \leq n$. Individual attribute values in a tuple are represented using the attribute and value pair as $t(A_i, v_i)$.

A relationship in relational databases is a situation that exists between two relational tables indicated by a foreign key constraint. The relationship between two tables is commonly referred as binary relationship. A group of binary relations may form a pattern that involve three (ternary), four (quaternary) or more tables that are commonly referred as N-ary relationships.

The tables involved in a relationship are classified as “strong” or “weak” tables depending on where the foreign key is placed. A “strong” table is indicated by a primary key (pk) database constraint using one or more column attributes while a “weak” table uses a foreign key to refer to the “strong” table.

Binary relationships are represented using a foreign key constraint that involve one or many cardinalities each side to form a one to one, one to many and many to many relationship.

- One-to-one (1:1)
- One-to-many (1:*)
- Many-to-many (*:*)

B. Domain Data Knowledge

Domain Data Knowledge describes how the domain data is used to represent knowledge in the mapped RDF Schema. Domain data is not represented explicitly but rather derived from the existing “relational database area knowledge”.

Data patterns between relationships can be represented as domain knowledge using semantic web knowledge representation languages like OWL [14].

N-ary relationships that involve more than two tables create different patterns of relationships. These patterns are discovered using the primary and foreign key “relational database area knowledge” constraints on one or more columns. Patterns like “chain”, “star”, “triangular” help to identify more “domain data Knowledge” in addition to the “relational database area knowledge”.

- Transitive Chain of Relations
- Disjointness
- Symmetries
- Star Relations, etc.

For instance, if there is a pattern where a database column is used both as a *primary key* ‘pk’ and a *foreign key* ‘fk’, then the referencing table is a “subClass” of the referenced table. This “predicate” uses primary key ‘pk’ and foreign key ‘fk’ “relational database knowledge” to create a “subClass” relationship using a *symmetric*-type pattern on a column that is being used both as a primary key ‘pk’ and foreign key ‘fk’.

Referencing Table T, Referenced Table T’,
Column attribute A, primary key pk(T), foreign key fk(T)

```
Begin
If( (A in (fk(T))) AND (A in (pk(T)))) then .

<owl:Class rdf:ID="T" >
  <rdfs:subClassOf rdf:resource="#T"/>
</owl:Class>

End if .
End .
```

Figure 3. Domain Data Knowledge "predicate"

C. Application Specific Knowledge

The ‘Space Project Management’ (SPP) domain-specific knowledge is used as a database schema source for evaluating the mapping of database application domain knowledge into semantic web. The domain-specific knowledge is used to ‘prune’ the semantic data at different stages of the *pre-processing* phase. The domain-specific knowledge is also a source for pattern discovery and interpretation at different stages of the *in-processing* and *post-processing* phase.

The applications used in SPP utilise different concepts and terminologies in the domain specific knowledge. The concepts that are used in the domain specific knowledge are

summarised as *Documents, Risks, Non Conformances, Reviews, Actions, and Projects*. These concepts have unique as well as common domain specific knowledge. The unique domain specific knowledge is used to identify the concept while the common ones are used to associate and correlate the different domain specific concepts. The different concepts and sub-concepts establish domain specific SPP-Ontology.

Each “concept” has attributes that define and explain the concept. There are also sub-concepts that are related to the concept through relationships. Some attributes like “status type” are used to identify an application specific “status” within SPP-Ontology.

Both “concepts” and “Relationships” may have further sub-concepts and sub-relationships represented by a nested square bracket ([]).

- Attributes- [URI, Definition, Title, {Status}]
- Status Types (sample)- [Register, Acknowledge, Assign Controller, Reduce, Accept, Resolve, Close]
- Sub-Concepts (sample)- [Domain, Scenario, Criticality [Likelihood, Severity], Rank, Trail]
- Relationship Types (sample)- [hasDomain, ofScenario, hasCriticality [hasLikelihood, hasSeverity], hasRank, hasTrail]

The concept URI attribute is a unique representation of the particular concept in SPP-Ontology. The “definition” and “title” attributes have the formal detailed definition and short descriptive title respectively.

The domain specific knowledge relies on attributes like “Relationship Type” to determine the semantic relationship between a “concept” and a “sub-concept” within SPP-Ontology.

For instance, if the application uses a common “Document Repository”, an application-specific “predicate” can be used to check and relate all concepts to point to the document repository. This “predicate” is executed whenever the *primary key* ‘pk’ of “Document” concept is used as a *foreign key* ‘fk’ in the rest of the concepts.

IV. RDB-RDF MAPPING ALGORITHMS

Currently, the first phase of creating semantic RDF Schema ontology is finalised using the mapping procedures below. The procedures use the three layers of the use case knowledge- “relational database area knowledge”, “domain data knowledge” and “application specific knowledge”. After the data mapping, different levels of RDF inferencing will be applied to further explore the knowledge base.

1) *mapDatabase*

The database mapping procedure uses incremental iterative approach that loops through all tables T and their

column attributes A within the schema S. The database mapping starts by mapping the tables- mapTable().

The tables are mapped into classes C. Corresponding to each class an RDF Repository C_RDF object is also created. The structure of the RDF repository is based on a TRIPLE format (subject, predicate, object).

After mapping the tables, mapColumn() algorithm maps the columns in each table into property column “hasP” of the existing class table C. mapColumn() procedure uses mapDatatype() procedure to return RDF equivalent data types for each relational attribute.

The different relational database constraints are also mapped using mapConstraint() procedure. The procedure maps constraints like primary key ‘pk’, foreign key ‘fk’, UNIQUE ‘unq’, NOT NULL ‘nn’, and CHECK ‘ck’.

Finally, the algorithm maps the different relationships among tables and columns. MapRelationship() algorithm uses the foreign key constraint between tables to find out the different types of relationship with respect to degree of relationship, transitive chain of relation, disjointness, etc.

Procedure mapDatabase (S)

Input: Schema S

Begin

mapTable(S) .
 mapColumn(S) .
 mapConstraint(S) .
 mapRelationship(S) .

End .

Figure 4. mapDatabase() Algorithm

2) *mapTable*

The Semantic Web equivalent of the relational algebra relations is a Class. During mapping the same name for the table T is used for the mapped class C. The class name is used to map subsequent relational columns into semantic class properties.

The classes in the Semantic Web repository can be considered repositories of data to hold the relational data after the mapping. Each Class table C represents the relational table in semantic web. To represent the class data in RDF triples (subject, predicate, object), a separate RDF repository C_RDF is created. In C_RDF the class ID (i.e., primary key equivalent of the source table) is used as a ‘subject’ while the rest of the properties are used as ‘predicates’. Each property “value” corresponding to the class ID is the ‘object’ of the RDF triple.

In addition to the class table C and RDF repository C_RDF, an OWL class is created using the class table C as an ID.

Procedure mapTable (S)

Input: Schema S

Output: Class C, RDF Repository C_RDF, OWL Class

Begin

For each table Ti in S loop .

Create Class Table Ci .

Create RDF Repository Ci_RDF
 using Class Table Ci and a TRIPLE type attribute .

<owl:Class rdf:ID="Ci" />

End loop .

End .

Figure 5. mapTable() Algorithm

3) *mapColumn*

The column attributes in the relational database are mapped as properties in semantic classes. A property in a class can describe an entity class or a relationship class. Each property has a set of allowable domain values that could be shared with one or more properties.

The columns are broadly divided as “key columns” that are used to identify an occurrence of a relation and “simple columns” that only describe a relation. During mapping columns into properties, the following column types are used as criteria to choose the appropriate representation in the semantic web.

Column types

- Candidate Key (CK): minimal set of attributes that uniquely identifies each occurrence of an entity type.
- Primary Key (PK): candidate key selected to uniquely identify each occurrence of an entity type.
- Foreign Key (FK): referencing a primary key (PK) in another relation
- Simple Column (SC): a non-candidate key that describes an entity type.

The column attributes of a table is denoted as T(A1, A2, ..., An)

T(A1, A2, ..., An) = {PK, {CK1, CK2,..., CKx}, {FK1, FK2, ..., FKy}, {SC1, SC2, ..., SCz}}

Where x, y, z is a whole number.

When the cardinality of x is greater than 1, candidate keys (CK) are treated as *composite* keys.

The column attributes in the relational database are mapped to corresponding class properties. The constraint type associated with the attribute determines the cardinality. Each column attribute A is mapped to property P prefixed by the word “has” as “hasP”.

Procedure mapColumn (S)

Input: Schema S, Table T, Column attribute A

Output: Property P, OWL:DatatypeProperty

Begin

```

For each table Ti in S loop .
    For each Column Aj in Ti loop .
        get mapped Class Table Ci of Ti.
        set Aj as Property Column hasAj
        .
        get &xsd:type_equivalent (Aj) .

        <owl:DatatypeProperty rdf:ID="hasAj">
        <rdfs:domain rdf:resource="#Ci" />
        <rdfs:range rdf:resource
        ="&xsd:type_equivalent" />
        </owl:DatatypeProperty>

    End loop .
End loop .
End .

```

Figure 6. mapColumn() Algorithm

4) *mapConstraint*

mapConstraint() algorithm maps relational database constraints into their equivalent semantic web representation. The algorithm reads both table level as well as column level constraints. For primary key pk(T) constraints, it creates “InverseFunctionalProperty” and “maxCardinality” OWL properties. For foreign key fk(T) constraints, it creates “ObjectProperty” OWL property. If a foreign key column is also part of the primary key pk(T) constraints, then the referencing table (T) is set as a “subClass” of the referenced table (T’).

For UNIQUE ‘unq(A)’, NOT NULL ‘nn(A)’, and CHECK ‘ck(A)’ database constraints, the algorithm creates equivalent and “InverseFunctionalProperty”, “minCardinality”, and “hasValue” OWL property restrictions respectively.

If the column attribute is a primary key pk(T) of the table, the maximum cardinality of the property car(P) is set to one. If the column attribute has a unique constraint unq(A), the maximum cardinality of the property car(P) is set to one. On the other hand if the property has a NOT NULL constraint nn(A), the minimum cardinality of the property car(hasP) is set to one.

mapConstraint() procedure maps column attribute(s) A using the table T and its constraints (primary key, foreign key, UNIQUE, NOT NULL, CHECK) to a semantic property P and OWL cardinality properties.

```

Procedure mapConstraint (S)
Input: Schema S, Table T, Referenced Table T’, Column
attribute A, primary key pk(T), foreign key fk(T),
UNIQUE unq(A), NOT NULL nn(A), and CHECK
ck(A)
Output: RDFS subClassOf, Property P, OWL cardinality
properties
Begin
For each table Ti in S loop .
For Column Aj in Ti loop .

```

```

get mapped Class Table Ci of Ti.
If (Aj in (pk(Ti))) then .
    <owl:InverseFunctionalProperty rdf:resource="# hasAj "/>

    /* set maximum car(hasAj) to 1 . */
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:maxCardinality
            rdf:datatype="&xsd:nonNegativeInteger">1
            </owl:maxCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>

Else
if (Aj in (fk(Ti))) then .
    If (Aj in (pk(T’i))) then .
        <rdfs:subClassOf rdf:resource="#C" />
    End if .
        <owl: ObjectProperty rdf:ID="hasA">
        <rdfs:domain rdf:resource="#C" />
        <rdfs:range rdf:resource="#C" />
        </owl: ObjectProperty >

Else
if (unq(Aj)) then .

    <owl:InverseFunctionalProperty rdf:resource="# hasAj "/>

Else
if (nn(Aj) and (!pk(Aj)) then .
    /*set minimum car(hasAj) to 1 .*/
    <owl:Restriction>
        <owl:minCardinality
        rdf:datatype="&xsd:nonNegativeInteger">1
        </owl:minCardinality>
    </owl:Restriction>

Else
if (ck(Aj)) then .
    <rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#hasAj" />
        <owl:hasValue rdf:datatype="&xsd:string" > v(Aj)
        </owl:hasValue>
    </owl:Restriction>
    </rdfs:subClassOf>
End if .
End loop .
End loop .
End .

```

Figure 7. mapConstraint() Algorithm

5) *mapRelationship*

A relationship between tables is identified by a foreign key. MapRelationship() calls a series of sub-procedures to identify the type of relationship between the two tables and

discover any patterns with the rest of the tables in the schema.

```

Procedure mapRelationship (S)
  Input: Table T, Column attribute A, foreign key fk(T)
  Output: Class C, Property P
Begin
  For each table Ti in S loop .
    For each column Aj in Ti loop .
      If (Aj = fk(Ti)) then
        CheckRelationship(Ti, T'i)
        CheckTransitiveChain(Ti, T'i) .
        CheckDisjointness(Ti, T'i)

        where T'i is referenced table by Ti
      End if .
    End loop .
  End loop .
End .
    
```

Figure 8. mapRelationship() Algorithm

6) Check Relationship

CheckRelationship algorithm uses the referencing table T and the referenced table T' to determine whether to create a separate class to represent the relationship or only add a property column to the existing class.

If the foreign key fk(T) is referring to a primary key pk(T') and the foreign key has a NOT NULL constraint, then a new relationship class is created using the two tables (T, T'). An additional "subClass" parameter is also passed to create a "subClass" axiom between the "Class" equivalents of the tables T and T'.

If the above criterion is not satisfied, the foreign key fk(T) would have been added as a property to the referring table equivalent class C using the mapColumn() procedure above.

```

Procedure CheckRelationship(T, T')
  Input: Table T, primary key pk(T), foreign key
  fk(T), NOT NULL nn(T)
Begin
  If (fk(T) = pk(T') and (fk(T) = nn(fk(T)))) then .
    CreateRelationship(T, T', subClass) .
  End if .
End .
    
```

Figure 9. CheckRelationship() Algorithm

7) Check Transitive Chain

Transitive Chain of relations is tested using the foreign key/primary key column attributes between three relational tables.

For any relational tables T1, T2, T3 in Schema S, if there is a foreign key relationship between T1 and T2 and if there is also a foreign key relationship between T2 and T3, then there is a transitive chain between T1 and T3.

The algorithm uses the referenced table's (T') columns to find further foreign key relationship to the rest of the tables. If another relationship other than the one between T and T' is found, a new relationship class is created using the two tables (T, Ti). An additional "Transitive" parameter is also used to create a "transitive" axiom between the "Class" equivalents of the starting table T and the new third table Ti.

```

Procedure CheckTransitiveChain(T, T')
  Input: Table T, Column attribute A, primary key pk(T),
  foreign key fk(T)
Begin
  For each column Ai in T' loop .
    If (Ai in fk(T')) then .
      For each table Ti in S loop .
        If ((Ai in pk(Ti)) and (Ti != T)) then .
          createRelationship(T, Ti, Transitive) .
        End if .
      End loop .
    End if .
  End loop .
End .
    
```

Figure 10. CheckTransitiveChain() Algorithm

8) Check Disjointness

Disjointness is a relationship between two "SubType" tables that share a common "SuperType" but has no relationship between each other.

The foreign key/primary key column attributes between the tables is used to determine the disjunction between tables.

For any relational tables T1, T2, T3 in Schema S, if there is a foreign key relationship between T1 and T2 and there is also a foreign key relationship between T2 and T3 but there is no relationship between T1 and T3, then there is a disjointness between T1 and T3.

The algorithm uses the referenced table's (T') columns to find further foreign key relationship to other tables. If another relationship other than the one between T and T' is found and there is no foreign key relationship between the new table Ti and the starting table T, then a "disjointWith" axiom is created between the starting table T and the new table Ti.

Note that a group of disjoint relationships create a "Star" relation with the "SuperClass" in the middle and the disjoint classes as a branch.

```

Procedure CheckDisjointness(T, T')
  Input: Table T, Column attribute A, primary key pk(T),
  foreign key fk(T)
    
```

```

Output: RDFS subClassOf, OWL Class, disjointWith
Begin
  For each column Ai in T' loop .
    If (Ai in fk(T')) then .
      For each table Ti in S loop .
        If ((Ai in pk(Ti) and (Ti != T)) then .
          if ((ALL) fk(Ti) NOT in
              (ALL) pk (T)) then .

            <owl:Class rdf:ID="Ti">

            <rdfs:subClassOf rdf:resource="#T"/>

            <owl:disjointWith rdf:resource=" #T "/>
            </owl:Class>

            End if .
          End loop .
        End if .
      End loop .
    End loop .
  End .

```

Figure 11. CheckDisjointness() Algorithm

9) Create Relationship

CreateRelationship algorithm is used to create a new relationship class table to represent the relationship between the referencing table T and the referenced table T'. If there is a foreign key in table T that references to a primary key in table T', a new class table is created using a class symbol C and the name of the referencing and referenced tables respectively separated by an underscore as "C_T_T'".

The primary keys of both tables are added as property columns to the new class by adding "has" as a prefix as "hasP" and "hasP'".

In addition to the semantic class, a repository is also created to represent the class table data in RDF triples (subject, predicate, object). The RDF repository reads the class table data and presents it in RDF triples. It is named using the class table names suffixed by "RDF" as "C_T_T'_RDF".

```

Procedure CreateRelationship(T, T', TYPE)
Output: RDFS subClassOf, OWL Class, ObjectProperty
Begin
  If (fk(T) = pk(T')) then .

    create Class C_T_T' .
    set pk(T) as Property hasP of Class C_T_T' .
    set pk(T') as Property hasP' of Class C_T_T' .

    create RDF Repository C_T_T'_RDF
    using Class Table C_T_T' and a TRIPLE
    type attribute .

    get mapped Class Table C of T .
    get mapped Class Table C' of T' .

```

```

If (TYPE = 'subClass') then
  <owl:Class rdf:ID="C">
    <rdfs:subClassOf
      rdf:resource="#C" />
  </owl:Class>

Else
  if (TYPE = 'Transitive') then

    <owl:ObjectProperty rdf:ID="pk(T)">
    <rdf:type rdf:resource
      ="owl:TransitiveProperty"/>
    <rdfs:domain rdf:resource="#C" />
    <rdfs:range rdf:resource="#C" />
    </owl:ObjectProperty>

  End if .

End .

```

Figure 12. CreateRelationship() Algorithm

V. IMPLEMENTATION

The implementation of the algorithms is based on the Space Project Management scenario. An interactive Java application is used to execute the mapping procedures and the resulting Semantic Web repository loaded in Protégé (a free open-source Java tool providing an extensible architecture for the creation of customized knowledge-based applications) OWL editor is shown on the figure below.

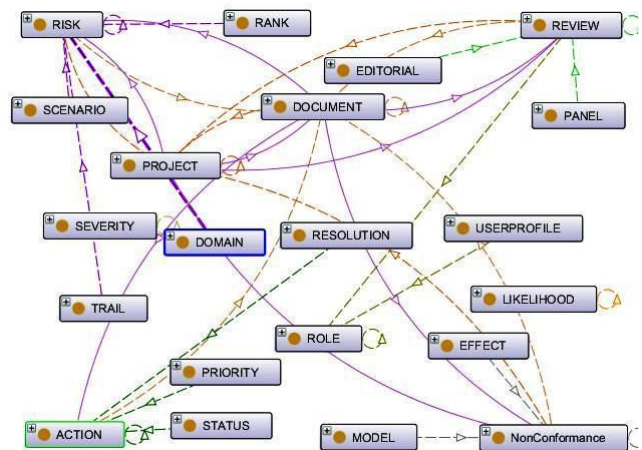


Figure 13. RDF Schema view (Protégé)

The Ontology diagram modeled above is also serialisable using OWL file format as in Figure 14 below.

```

- <rdf:RDF xml:base="urn:spaceProj-Mgmt/">
+ <owl:Ontology rdf:about="SpaceKnowledgeManagement"></owl:Ontology>
- <owl:Class rdf:about="urn:spaceProj-Mgmt/ACTION">
  <owl:FunctionalProperty rdf:resource="urn:spaceProj-Mgmt/hasACTIONID"/>
- <rdf:subClassOf>
  + <owl:Restriction></owl:Restriction>
  </rdf:subClassOf>
- <rdf:subClassOf rdf:resource="urn:spaceProj-Mgmt/DOCUMENT"/>
- <rdf:subClassOf>
  + <owl:Restriction></owl:Restriction>
  </rdf:subClassOf>
+ <rdf:subClassOf></rdf:subClassOf>
+ <rdf:subClassOf></rdf:subClassOf>
</owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/ATTACHMENT"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/CATEGORY"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/CLASSIFICATION"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/COLLECTION"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/COMPETENCY"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/CRITICALITY"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/DISPOSITION"></owl:Class>
+ <owl:Class rdf:about="urn:spaceProj-Mgmt/DOCUMENT"></owl:Class>
  
```

Figure 14- RDF Schema Overview (OWL file excerpt)

The mapping test between the Relational Database Schema (Figure 2) and RDF Schema (Figure 13 & Figure 14) is evaluated using meta-data search comparisons as shown by the sample screenshots in Figure 15 & Figure 16 below.

Table Name	Column Name	Data Type	Data Length
ACTION	ACTIONID	NUMBER	22
	DOCUMENTID	NUMBER	22
	ACTIONSTATUS	VARCHAR2	50
	ACTIONTITLE	VARCHAR2	50
	ATTACHMENTID	NUMBER	22
ATTACHMENT	ATTACHMENTID	NUMBER	22
	CATEGORYID	NUMBER	22
	NONCID	NUMBER	22
CATEGORY	CATEGORYNAME	VARCHAR2	20
	CLASSID	NUMBER	22
	REVIEWID	NUMBER	22
CLASSIFICATION	CLASSNAME	VARCHAR2	20
	COLLECTIONID	NUMBER	22
	DOCUMENTID	NUMBER	22
COLLECTION	COMPETENCYID	NUMBER	22
	SPACEUNITID	NUMBER	22
	COMPETENCYNAME	VARCHAR2	2
COMPETENCY	CRITICALITYID	NUMBER	22
	LIKELIHOODID	NUMBER	22
	RISKID	NUMBER	22
CRITICALITY			

Figure 15. Relational Database Schema search result

Subject	Predicate
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#range
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/2000/01/rdf-schema#range
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
urn:spaceProj-Mgmt/hasACTIONID	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
urn:spaceProj-Mgmt/hasCOLLECTIONID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasCOLLECTIONID	http://www.w3.org/2000/01/rdf-schema#range
urn:spaceProj-Mgmt/hasCOLLECTIONID	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
urn:spaceProj-Mgmt/hasDOCUMENTID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasDOCUMENTID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasDOCUMENTID	http://www.w3.org/2000/01/rdf-schema#domain
urn:spaceProj-Mgmt/hasDOCUMENTID	http://www.w3.org/2000/01/rdf-schema#domain

Figure 16. RDF Triples search result

We have compared our attempt to some of the existing approaches (Direct Mapping, eD2R, R2O, Relational.OWL, Virtuoso, D2RQ, Triplify, R2RML, R3M). We used the comparison criteria specified in the seminal paper [6]. The results of this comparison can be summarized as shown in Table I and Table II.

TABLE I. RDB-TO-RDF MAPPING LANGUAGES COMPARISON LEGEND [6]

Legend	Description
F1	Logical Table to Class
F2	M:N Relationships
F3	Project Attributes
F4	Select Conditions
F5	User-dened Instance URIs
F6	Literal to URI
F7	Vocabulary Reuse
F8	Transformation Functions
F9	Datatypes
F10	Named Graphs
F11	Blank Nodes
F12	Integrity Constraints
F13	Static Metadata
F14	One Table to n Classes
F15	Write Support
F16	Data Patterns
F17	Data Control Languages (DCL)

TABLE II. SUMMARY TABLE OF RDB-TO-RDF MAPPING LANGUAGE COMPARISON [6] WITH RD2SW

	F1	F2	F3	F4	F5	F6	F7	F8	F9
Direct Mapping	(✓)	✗	✗	✗	✗	✗	✗	✗	✗
eD2R	✓	✓	✓	✓	✓	✓	✓	✓	✓
R ₂ O	✓	✗	✓	✓	✓	✓	✓	✓	✓
Relational.OWL	(✓)	✗	✓	✗	✗	✗	✗	✗	✓
Virtuoso	✓	✓	✓	✓	✓	✓	✓	✓	✓

D2RQ	✓	✓	✓	✓	✓	✓	✓	✓	✓
Triplify	✓	✓	✓	✓	✓	✓	✓	✓	✓
R2RML	✓	✓	✓	✓	✓	✓	✓	✓	✓
R3M	(✓)	✓	✓	(✓)	✓	✓	✓	✓	✓
RD2SW	✓	✓	✓	✓	✓	✓	✓	✓	✓

	F10	F11	F12	F13	F14	F15	F16	F17
Direct Mapping	x	x	x	x	x	✓		
eD2R	x	✓	(✓)	x	✓	x		
R ₂ O	x	✓	(✓)	x	(✓)	x		
Relational.OWL	x	✓	(✓)	x	x	✓		
Virtuoso	✓	✓	(✓)	x	✓	x		
D2RQ	x	✓	(✓)	✓	✓	x		
Triplify	x	x	(✓)	x	✓	x		
R2RML	✓	✓	(✓)	✓	✓	x		
R3M	x	x	✓	x	✓	✓		
RD2SW	x	✓	✓	x	✓	x	✓	✓

VI. CONCLUSION AND FUTURE WORK

In the current implementation, the mapping procedure reads the database directly, starting with the data dictionary and then the data. The developed algorithms use a configuration file to choose a relational database driver and access the database dictionary to map into an RDF format. It involves formulation of heuristics that formally define the mapping. The domain specific heuristics have helped harvest the Ontology of the Space Project Management Database. This domain specific heuristics was implemented using an incremental algorithm to extend the domain ontology repository. The formulated heuristics and domain ontology repository have been implemented and tested on a prototype Space Project Management semantic tool as a proof-of-concept to the research. This heuristic-based methodology can be applied and measured on other relational data with different domain ontology.

Currently, we used set of heuristics for accounting the different types of *relational model knowledge* (constraints, data types), *domain specific knowledge* (simple data patterns like transitive chain and disjointness) and *application specific knowledge* (predicates). In the future we plan to extend the algorithm so that it also accounts other types of *domain specific knowledge* like complex data-patterns, *user domain knowledge* (individual and group users, access rights and profiles) and *application domain knowledge* (i.e., triggers and transactions). We also plan to implement a parser for SQL DDL, used to create the database. It will be still necessary to connect to the database in order to elicitate and convert the data stored in it, but this will eliminate the need for using the data dictionary and thus, it will reduce the database dependency.

The process of mapping Relational Databases to Semantic Web (RD2SW) using domain specific knowledge involves most of the current - Semantic Web Layer components- RDF, RDF schema, query languages, rules, logic, etc. Following the Semantic Web standards set by

World Wide Web Consortium will ultimately help us represent ‘web resources’ in a standardized, unambiguous, interoperable and above all Linked-Data format as the next efficient phase of representing knowledge in the 21st century.

ACKNOWLEDGMENT

The authors acknowledge the sponsorship of this PhD research work by the London Metropolitan University Vice Chancellor’s fund. Implementation and evaluation is conducted in collaboration with Sapienza Consulting [16] a leading software provider for space mission and project support.

REFERENCES

- [1] D. Allemang and J. Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann Publishers, CA, 2008
- [2] C. Bizer and R. Cyganiak., “D2RQ Lessons Learned” Position paper for the W3C Workshop on RDF Access to Relational Databases, Cambridge, MA, USA, October 2007, pp. 25-26.
- [3] C. Blakeley, “RDF Views of SQL Data (Declarative SQL Schema to RDF Mapping)” OpenLink Software, 2007.
- [4] K. Byrne, “Having Triplets Holding Cultural Data as RDF” Proceedings of the ECDL 2008 Workshop on Information Access to Cultural Heritage, Aarhus, Denmark, September 2008.
- [5] R. Cyganiak, “A relational algebra for SPARQL” Digital Media Systems Laboratory HP Laboratories Bristol. HPL-2005-170, 2005.
- [6] M. Hert, G. Reif, and H. Gall, “A comparison of RDB-to-RDF mapping languages” In Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria, ACM, 2011, pp. 25-32.
- [7] S. Staab and R. Studer, Handbook on Ontologies. Springer Berlin: 2009.
- [8] S.S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau, S. Auer, J. Sequeda, and A. Ezzat, “A Survey of Current Approaches for Mapping of Relational Databases to RDF” W3C RDB2RDF XG Incubator Report: W3C, 2009.
- [9] A. Sheth, C. Ramakrishnan, and C. Thomas, “Semantics for the Semantic Web: The Implicit, the Formal and the Powerful” Int’l Journal on Semantic Web & Information Systems, 2005, pp. 1-18.
- [10] R. Studer, V. R. Benjamins, and D. Fensel, “Data and Knowledge Engineering” Knowledge engineering: Principles and methods, 1998, pp.161-197.
- [11] E. Marx, P. Salas, K. Breitman, and J. Viterbo, “RDB2RDF: A relational to RDF plug-in for Eclipse” Published online in Wiley Online Library (wileyonlinelibrary.com), 2012 [retrieved: December, 2012].
- [12] F. Manola, E. Miller, and B. McBride, “RDF Primer” W3C Recommendation, Feb. 2004.
- [13] R.V. Guha and B. McBride, “RDF Vocabulary Description Language 1.0: RDF Schema” W3C Recommendation, Feb. 2004.
- [14] D.L. McGuinness and Frank Van Harmelen, "OWL web ontology language overview." W3C recommendation, Feb 2004.
- [15] NCBO BioPortal, The National Center for Biomedical Ontology, [retrieved: December, 2012] from <http://bioportal.bioontology.org>
- [16] Sapienza Consulting, About Sapienza, [retrieved: December, 2012] from <http://www.sapienzaconsulting.com>

A Framework for the Coordination of the Invocations of Web Services

Mohammed Alodib
 Qassim University
 P.O BOX 385
 Buraidah 51411
 Qassim, Saudi Arabia
 alodib@qu.edu.sa

Abstract—Coordinating the various Web services invocations is one of the key challenges of Service oriented Architectures. When services fail due to lack of availability this may violate Service Level Agreements causing financial penalties or customer dissatisfaction to providers. Therefore, it is crucial to develop a method of on-line coordination of these invocations in order to enhance the performance of the systems in place and avoid the overuse of services. This paper aims to present a Model Driven Architecture (MDA) approach to the automated creation and integration of Protocol Services, which are deployed with the system to coordinate invocations between services. The outline of this method is as follows. Business Process Execution Language (BPEL) models of services are parsed and the PartnerLink for each Invoke activity is assigned to the Web Services Description Language (WSDL) file of the Protocol Service using MDA transformations. Then, the Protocol service is computed, generated and integrated automatically into the system. As a proof of concept, an implementation of the suggested approach was created, in the form of an Oracle JDeveloper plugin that automatically produces new Protocol services and integrates them with existing services.

Keywords-Web services; Quality of Service; Coordinating; Model Driven Architecture

I. INTRODUCTION

Service oriented Architecture (SoA) is a framework which provides a layered architecture for organising software resources as services, so that they can be deployed, discovered and combined to produce new services [1]. In real-world business processes, it is crucial to develop architectures to discover the most suitable service in order to avoid excessive use of services, and so enhance the performance of the system.

In the current version of SoA, an invocation request is processed using BPEL activity known as Invoke. This requires assigning a WSDL file of the target service to the Partner Link property of the Invoke activity. If the destination service becomes unavailable for any reasons, this may cause distraction to other services and lead to customer dissatisfaction. In addition, a service may become slow in its responses due to the uncoordinated overuse of its operations by other services. To resolve such issues, the WSDL file for the failed service, or the slow service associated with the Invoke activity can be manually replaced with another WSDL file, one designed for a service that gives the same

result, but it is deployed by a different provider. Therefore, a model-driven approach to automating this replacement is proposed.

The presented approach provides a dynamic technique to discover the best available service using a simple genetic algorithm. This algorithm is based on ranking Web services using the previous invocations history. In this approach, all the invocations are forwarded to a *Protocol service*, which works as a coordinator controlling all invocations. The the Protocol service is initiated by the request from the consumer; it then forwards the request to the target service, obtaining the result from the provider, and returning the result to the consumer.

From a performance perspective, this architecture can potentially result in a bottleneck, as all invocations should be processed by the Protocol service. However, the Protocol service is distributively generated and integrated into the system; i.e. each site has its own Protocol service, which controls the internal invocation requests. When faced with an external invocation for a remote service, i.e. deployed at different server, the Protocol service interacts with the Protocol service located at the external site by forwarding the invocations.

This paper is organised as follows. Section III-A presents a brief review of Service oriented Architecture (SoA). Section III-B reviews the Web Services. An introduction to the fundamentals of the Business Process Execution Language (BPEL) is described in Section III-C. Section IV presents the principles of Model Driven Architecture (MDA). The description of the problem is discussed in Section V. Section VI presents the solution, which is implemented as an Oracle JDeveloper's plugin.

II. DISCUSSION AND RELATED WORKS

Yan et al. [2], [3] proposed a method to monitor Web services in order to trace faults and recover from their effects. Their method utilises Model-Based Diagnosis (MBD) theory [4], which provides techniques to monitor static and dynamic systems using partial observations. Such methods require in-depth knowledge of the system. Their method is designed to monitor failures, such as mismatching parameters when occurrences are thrown up as exceptions. On the other hand,

our approach aims to deal with monitoring and coordinating the invocations in order to avoid failed or overused services. Our goal is to maintain the system at a high level of performance by avoiding dynamically failed or overused services.

Ardissono et al. [5] also proposes a model-based approach to monitor and diagnose Web services. Their approach aims to provide self-healing services, which guarantee autonomous diagnostic and recovery capability. This approach is based on adopting grey-box models for each Web service to expose the dependency relationships between the input and output parameters to the public. The dependency relationships are used by Diagnostosers to determine the service which results in exceptions.

In [6], [7], we present approaches to dealing with monitoring failures caused by undesirable scenarios, such as Right-First Time (RFT) Failure, which occurs when a business process fails to complete a task the First-Time and is forced to repeat a part of the task again (i.e., when a task is executed more than once, indicating incorrect execution of the task in the first place, or the invocation of an erroneous execution). Such occurrences of failure may result in violations of Service Level Agreements (SLA).

III. PRELIMINARIES

A. Service oriented Architecture (SoA)

SoA is directed towards the implementation of business processes via the composition of interactive services [1]. In general, SoA is a prevailing software engineering product, which ends the domination of traditional, distributed system platforms [8]. The growth rate for SoA use in industry has been estimated at over 24%, as measured between 2006 and 2011 [9], and the rapid movement towards SoA has been encouraged by the positive results already recorded; for example, the level of reusability in SoA has, on average, been enhanced to more than 2.5 times that of non-SoA development.

A simple SoA infrastructure involves three independent collaborative components, which are described below [10], [11]; see Figure 1:

- **Service provider:** The service provider is responsible for publishing the services, and is the owner of the services; e.g. companies and organisations.
- **Service requester:** A requester is a client or organisation that wishes to make use of a service that is being provided. The requester searches for the Web services desired from the service registry.
- **Service registry:** A global registry acts as a central service which provides a directory where service descriptions are published by the Service Provider. Then, *Service Requesters* find service descriptions in the registry and obtain binding information for services from the *Service Provider*.

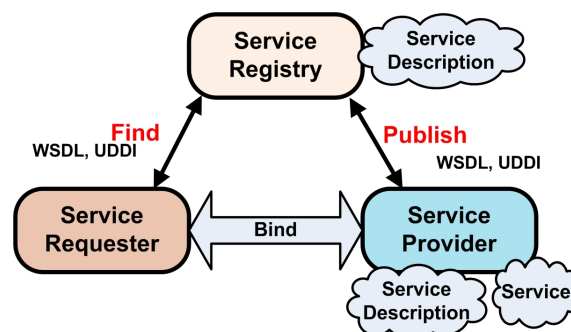


Figure 1. A Basic Service Oriented Architecture [11]

B. Web Services

Web services offer a preferred solution to the problem of integration among autonomous and heterogeneous software systems [12]. They are well-defined, self-contained, loosely coupled, self-describing, modular applications that can be published, located and invoked across the Web network [12]. These features mean that Web services have the ability to be dynamically invoked by other applications or other Web services, and are composed in tandem with other services to achieve complex tasks. In other words, Web services are highly reusable components, acting as building blocks to develop service composition, as well as to solve the application communication and integration issues.

The development of a composite web service is built upon the Service oriented Architectural paradigm [13]. Communication in a composition of Web services is based on the use of well-accepted standards and the XML messaging framework [14]. Such standards can be used to encapsulate the service's business logic and functionality in order to expose the functionality only, not the implementations via the accessible interfaces. Therefore, application programs communicate with one another irrespective of their programming language, operating system and hardware platforms.

Web services communicate using common Extensible Markup Language (XML), XML Schema Definition XSD [15] and standard TCP/IP based communication protocols. Moreover, various XML-based standards are used by Web services in order to describe their architecture, intercommunication, collaboration and discovery [12]. In particular, the communication messages between a Service Requester and a Service Provider are encoded into *Simple Object Access Protocol (SOAP)* messages, which are plain text XML messages. The *Web Services Description Language (WSDL)* is used to describe the invocation details of a Web service, such as the service name, the operations available, and the information related to the input and output variables. The *Universal Description Discovery and Integration (UDDI)* provides protocols for querying and updating Web service information. For communication purposes, Web services

utilise existing standard TCP/IP protocols such as HTTP, HTTPS, SMTP, and FTP [16].

C. Business Process Execution Language (BPEL)

In the past few years, SoA has been adopted and widely used by the IT industry. One of the most popular standards of such adaptations is Business Process Execution Language for web services (BPEL) [17]. BPEL is a modelling language used to specify a sequence of actions that take place within business processes in order to generate enterprise applications. BPEL offers a rich number of diagrammatic notations ideal for supporting the modelling of complex behaviours; i.e. sequential, parallel, iterative and conditional. In addition, similar to traditional programming languages, BPEL offers constructs, in the form of loops, branches, variables and assignments.

Business Process Execution Language for Web Services (BPEL, WS-BPEL, BPEL4WS) is a graphical language that is used for the composition, orchestration, and coordination of Web services [1]. Combining and linking existing Web services and other components to deliver new composition services is referred to as *business processes*; therefore, BPEL is used to specify a set of actions within business processes, in order to achieve a common business goal. The BPEL specification is based on the Web Services Description Language (WSDL) [18], which is an XML language describing services as a set of accessible interfaces, for producing business processes that support interoperability [19].

IV. MODEL DRIVEN ARCHITECTURE (MDA)

Model Driven Architecture (MDA) [20], [21] is a framework introduced by the Object Management Group (OMG) in order to promote the role of modelling in software development. One of the main goals of MDA is model transformation; a process whereby models in a source language are mapped so as to be captured in the destination language. In the MDA context, model transformation is defined by a number of transformation rules, which specify the mapping of the *meta-elements* of the constructs of the *metamodel* of the *source language* into the *meta-elements* of the *destination language*. The metamodels of the source and the target language are specified using a common language, called the Meta Object Facility (MOF) [22]. In general, models in the MDA are instances of metamodels.

Meta Object Facility (MOF) Query/View/Transformation Specification (or QVT for short) [23] is the OMG specification, which is proposed as a method to specify model transformation rules with MOF. QVT provides a declarative and imperative language, structured into a layered architecture consisting of *Relations*, *Core* and *Operational Mappings*. *Relations language* is a high level language that provides a textual and graphical notation for the purpose of defining the mappings, while *Core language* is a small language based on Essential MOF (EMOF) and OCL, which is used to support

pattern matching and the evaluation of conditions. QVT *Operational Mappings language* is a high level imperative language that extends Object Constraint Language (OCL) [24] with essential features (such as the ability to define loops) in order to write complex transformation rules [23]. In this study, we used QVT Operational Mapping language to obtain the specifications for the transformation rules.

The QVT *Operational Mapping* language is specified as a standard method for providing imperative implementations. This language is based on using MOF as a repository for metamodels. The general syntax for the body of an Operational Mapping is depicted in Figure 2, where the *source* is the source of the model transformation. The *mappingFunction* is the name of the model transformation, which may require some inputs, as captured by variable *parms*. The *target* is the destination model of the transformation. The *'init'* part has some code which can be executed prior to implementing the main body of the mapping rules. The *population* is then used to populate the results of the mapping. The code included in the *end* part is executed before the operation completes. The *'when'* part has a Boolean expression that should be verified as true before commencing the execution. The *'where'* part includes the conditions that have to be satisfied by the model elements involved in the mapping (i.e., it acts as a post-condition for the mapping operation).

```
mapping source::mappingFunction(parms):target
  when {...}
  where {...}
  {
    init{...}
    population{...}
    end{...}
  }
```

Figure 2. The general syntax for the body of a mapping operation.

There are many industrial and academic case tools supporting model transformations, such as Kermeta [25], Arcstyler [26], OptimalJ [27], ATLAS [28] and SiTra [29]. In this paper, we will use the Simple Transformer (SiTra) [29] transformation engine to execute the transformation rules. SiTra is a lightweight Model Transformation Framework, which intends to use Java for both writing Model Transformations and providing a minimal environment for transformation execution.

V. DESCRIPTION OF THE PROBLEM

From a SoA point of view, an interaction between two services can be performed with the help of an Invoke activity, which is a BPEL component used to specify the operations of the service that we intend to execute. Such operations are identified using Partner link. To achieve this, the WSDL file for the target service is assigned to the Partner Link

property of the Invoke activity. However, the target service may then become unavailable due to technical issues; such as a failure in the system, updating procedures, or the high load of executions, and this may cause the process to crash and throw exceptions. Consequently, it is critical to identify failed services, so that suitable remedial actions can be taken.

The typical method for resolving those issues caused by unavailable services is to manually replace the WSDL file of the service, as linked to the Invoke activity with another service providing the same functionality, but deployed by a different server. For example, assume that there are two services called *find_flight* and *FlightSearch*. These services provide the same functionality, and it is supposed that there is an Invoke activity used to execute the *find_flight* service. If we assume that the *find_flight* service becomes unavailable for any reasons, it becomes necessary to perform a recovery action so as to solve that issue. This can be achieved by replacing the current WSDL file of the service with another one, such as *FlightSearch*. Although this solves the problem, it is both a costly and time consuming solution as it should be carried out manually by a developer. Therefore, a dynamic approach to enhance and automate the process of this replacement is presented.

The approach presented proposes a framework that provides on-line automated modifications. In other words, the approach aims to provide dynamic executions based on the automatic runtime replacement of the WSDL file, in case the target service becomes unavailable or where it is already overused.

VI. THE MODEL-DRIVEN APPROACH

The approach presented here proposes a service intended for monitoring and coordinating interactions between services. The service introduced is referred to as the *Protocol service* and aims to discover the best available service, depending on its performance and availability. This method requires that all invocations between services are carried out using a Protocol service, whereby each source service provides the name of the target service to the Protocol service. Then, the Protocol service checks all the services that match the request received. From a performance and availability point of view, it then evaluates these services in order to find the most suitable service.

The basic idea of the Protocol service is that it receives an invocation request from the source service and forwards this to the target service. Each invocation request involves the name of the target service, the inputs values for the target service. This request is then validated by the Protocol service to check whether the name of the service is valid, and to ensure that all the values for the required parameters of the destination service are provided. Based on the type of invocation, there are two options for processing the request received. Firstly, if it is an asynchronous invocation, i.e., no result is expected from the target service, then

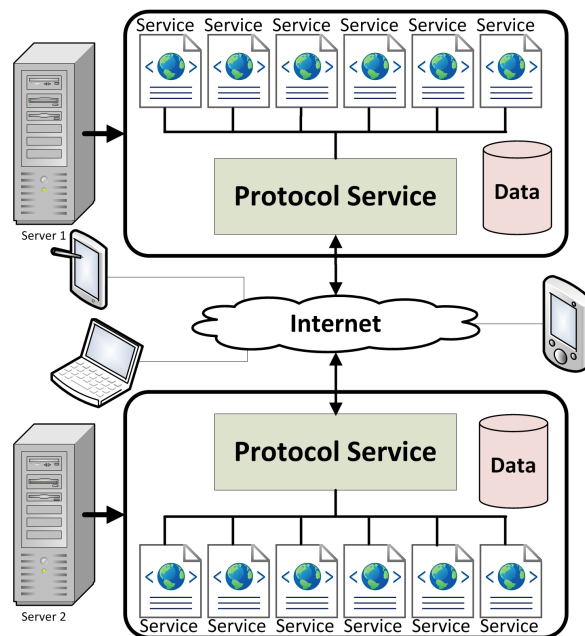


Figure 3. The proposed Architecture with the Protocol service

the Protocol service executes the target service and ends the process. Alternatively, if the request involves *two-way operations* (synchronous), the Protocol service executes the target service and the result is eventually returned to the consumer.

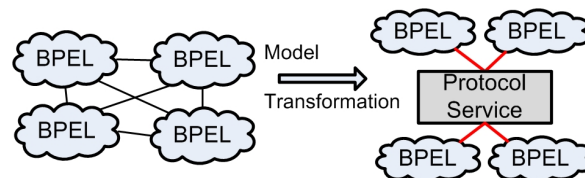


Figure 4. The outline of the Transformation method

The Protocol service is automatically and distributively generated for each site in the system as depicted in Figure 3. Each Protocol service is responsible for monitoring and coordinating interactions on the site in which it resides. An invocation between the two services deployed in two different sites requires that the Protocol service at the first site interacts with the Protocol service at the second site in order to complete the invocation. For example, suppose that we have two services (Service A and Service B) deployed at separate servers; Server 1 and Server 2 respectively. Assume that Service A intends to invoke Service B. To accomplish this invocation, Service A sends a request to the Protocol service in Server 1. This request includes details about the target Service. Next, the Protocol service at Server 1 forwards the request to the Protocol service at Server 2, which then carries out the invocation, and if the invocation is

a synchronous operation, it returns the result to the Protocol service at Server 1. Eventually, the Protocol service for Server 1 returns the results to Service A.

The evaluation task involves selecting the most suitable destination service as based on a simple genetic algorithm to rank Web services using the invocations history. This algorithm aims to check all services in order to identify the best services from a performance point of view. Due to the nature of SoA, which requires the assignation of a specific WSDL file of a service to each Invoke activity at the runtime, it is possible to note an excessive use of that service, despite the fact that there are other services offering the same functionality and that they can be used to avoid such an overuse. This may occur because there is no on-line coordinator for the distribution of the request received for the available services. Therefore, it can be seen that it is necessary to control routing requests between services in a balanced manner, and this is achieved using the Protocol service.

```
<invoke name="CheckCustomerAccount"
partnerLink="CustomerService"
portType="ns1:CustomerService"
operation="CheckCustomerAccount" />
```

Figure 5. A Constructor of an Invoke Activity

VII. INTEGRATION OF THE PROTOCOL SERVICE

For already pre-existing projects, the approach presented can be integrated automatically using a model-driven technique, which is implemented as an Oracle JDeveloper plugin. The implementation follows the outline of the method as depicted in Figure 4. This method requires passing all BPEL files and their XML Schema Definition (XSD) as inputs. For each BPEL file, the set of Invoke activities are extracted. Then, the Partner link for each Invoke activity is automatically replaced with the Partner Link for the Protocol service. For example, Figure 5 depicts a constructor of an Invoke activity used to execute a service called *CustomerService*. This is automatically modified by assigning the WSDL file of the Protocol service to the Partner Link property of the Invoke activity as depicted in Figure 6.

```
<invoke name="CheckCustomerAccount"
partnerLink="ProtocolService"
portType="ns1:ProtocolService"
operation="CheckCustomerAccount" />
```

Figure 6. A replaced Constructor of the Invoke Activity of Figure 5

As discussed in Section VI, the Protocol service requires that the user assigns the name of the target service and its inputs in order to complete the process. For this reason the

Assign activity precedes the Invoke activity, and is used to assign the inputs required by the target service, being also modified in order to assign the inputs and the name of the target service to the Protocol service.

```
<assign name="AssignID">
  <copy>
    <from variable="CustomerID"/>
    <to variable="FindCustomerInfoInput"/>
  </copy>
</assign>
```

Figure 7. A Constructor of Assign Activity

The Assign activity contains one or more *Copy* operations, which are used to copy data from one variable to another, as well as to construct and insert data using expressions [30]. Figure 7 presents a simple example of a construct for an *Assign* used to copy the value of *CustomerID* to *FindCustomerInfoInput*. To accomplish the required modifications, each *to* property of the Copy operations of the Assign activity is replaced and it is assigned to the Protocol Service input variable. The following code depicts a snippet of code that is then used to map each Assign activity to a new Assign activity, where the *to* property of the Copy operation is assigned to the input variable of the Protocol service. The following QVT transformation rule depicts the specification of our transformation explained above:

```
mapping Assign::assign2assign() : Assign
{
  name := self.name;
  foreach(e Element | copy:Copy)
  {
    e.form.variable=e.form.variable;
    e.to.variable="ProtocolServiceInput";
  }
}
```

VIII. CASE STUDY & EVALUATION

The presented approach is tested with the help of a simple case study described by Guillou et al. [31]. This example is based on a typical on-line e-shopping system consisting of three main services: Shop, Supplier and Warehouse.

As depicted in Figure 8, the customer accesses the Shop Web site to search for items. Then, he adds his items to the Shopping Cart which is then passed to the Supplier service by the Shop Service. For each item in the list, the Supplier service sends a request to the Warehouse to check if the item is available. If the item is available the Warehouse service sends an acknowledgement to the Supplier to complete processing the order. Next, the Supplier Service send back the list of available items to the Shop service. Finally, the list is forwarded to the customer who then confirms his order.

Evaluating the resources required to implement the Protocol service is considered a requisite task. Therefore, the

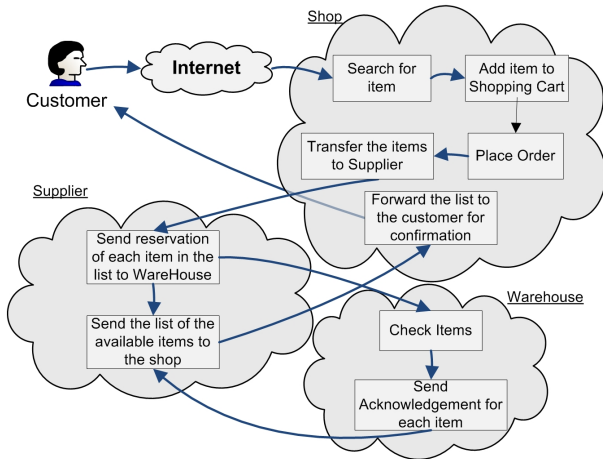


Figure 8. E-shopping Scenario

approach presented here have been evaluated in terms of performance and it is compared with the traditional method which does not use the Protocol service. A common practise in evaluating services of SoA is to utilise the *Stress Test*. The *Stress Test* is a technique used to identify and verify the stability, capacity and the robustness of services [1]. The Stress Test requires defining the number of the concurrent threads that should be allocated to the service, the number of loops, and the delay between invocations. With the performance statistics, we can identify any possible bottlenecks and optimise performance.

The example is implemented in two different methods; one is based on the traditional way, i.e. without the Protocol service, and the second method is to use the presented approach, i.e., using the Protocol service. The Stress Test has been applied to these methods by handling a different number of concurrent threads. This is specified as 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 threads. The delay between invocations is assigned to one second. The machine which is used in this test has the following configuration: Lenovo W520, Intel Core i72820QM 2.30GHz processor, 16G RAM.

The mean of the executions time has subsequently been calculated and the results are depicted as a line chart in Figure 9. The performance of using the Protocol service can be seen as linear and parallel, and it shows better performance.

In addition to the performance, modularity can play a key role in the early design stages of the software architecture discipline [32]. Therefore, using the Protocol service provides a modularised design, which brings to the system the following benefits:

- 1) Reliability: using the *Protocol Service* provides faster and more reliable processes.
- 2) Faster and easier development. The focus would be on

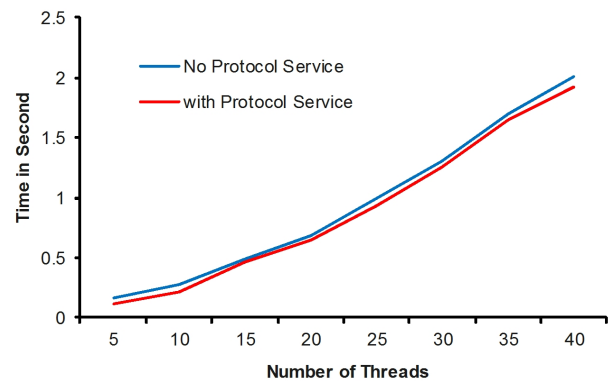


Figure 9. Stress Testing Result

the functionality of code modules rather than on the mechanics of implementation.

- 3) Faster and easier testing.
- 4) Maintainability: this also makes modification of enterprise project easier.

Moreover, using the Protocol service supports the fact that this architecture increases the efficiency of the system as it is considered an Orchestration architecture which is a more flexible paradigm offering the following advantages over the Choreography [1]: i) the coordination of component processes is centrally managed by a known coordinator; ii) Web services can be incorporated without being aware that they are taking part in a business process; iii) alternative scenarios can be put in place in case of a fault. However, it could be argued that using the *Protocol Service* may result in bottlenecks affecting the performance of the system.

IX. CONCLUSION

This paper has presented a method of developing a service that can monitor the execution of invocations in a Service oriented Environment. The underlying concept relies on utilising the capability of MDA to generate a Protocol service, which coordinates invocations between services in order to enhance the performance of a system by avoiding failure or overuse of services. The automation of the creation of the Protocol service is based on parsing the original BPEL services, and generate a set of modified services with an integrated Protocol service. The approach presented is implemented as an Oracle JDeveloper plugin.

REFERENCES

- [1] M. B. Juric, B. Mathew, and P. Sarang, *Business Process Execution Language for Web Services*. Packt Publishing, 2004.

- [2] Y. Yan, Y. Pencole, M.-O. Cordier, and A. Grastien, "Monitoring web service networks in a model-based approach," in *ECOWS05 (European Conference on Web Services)*, Sweden, 2005, pp. 192–203.
- [3] Y. Yan and P. Dague, "Modeling and diagnosing orchestrated web service processes," in *IEEE International Conference on Web Services*, vol. 9, Salt Lake City, Utah, USA, 2007, pp. 51 – 59.
- [4] W. Hamscher, L. Console, and J. de Kleer, Eds., *Readings in model-based diagnosis*. USA: Morgan Kaufmann Publishers Inc., 1992.
- [5] L. Ardissono, L. Console, A. Goy, G. Petrone, C. Picardi, M. Segnan, and D. Dupre, "Cooperative model-based diagnosis of web services," in *In 16th International Workshop on Principles of Diagnosis*, Monterey, 2005, pp. 125–132.
- [6] M. Alodib, B. Bordbar, and B. Majeed, "A model driven approach to the design and implementing of fault tolerant service oriented architectures," in *IEEE International Conference on Digital Information Management (ICDIM)*, London, 2008, pp. 464–469.
- [7] M. Alodib and B. Bordbar, "A modelling approach to service oriented architecture for on-line diagnosis," *the journal of Service Oriented Computing and Applications*, pp. 1–17, 2012.
- [8] D. W. McCoy and Y. V. Natis, "Service-oriented architecture: Mainstream straight ahead," Gartner Research, Tech. Rep., 2003.
- [9] J. B. Hill, M. Cantara, E. Deitert, and M. Kerremans, "Magic quadrant for business process management suites," Gartner Research, Tech. Rep., 2007.
- [10] M. P. Papazoglou, "A survey of web service technologies," 2004.
- [11] H. Kreger, "Web services conceptual architecture." IBM Software Group, 2001.
- [12] F. Leymann, "Web services: Distributed applications without limits," in *10th Conference on Database Systems for Business, Technology and Web (BTW'03)*, Leipzig, 2003, pp. 26–28.
- [13] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services Concepts, Architectures and Applications*. Springer, 2004.
- [14] "Semantic web services: description requirements and current technologies," in *In Proceedings of the International Workshop on Electronic Commerce, Agents, and Semantic Web Services held in conjunction with the Fifth International Conference on Electronic Commerce (ICEC)*, 2003.
- [15] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, "Xml schema part 1: Structures," 2004.
- [16] H. Petritsch, "Service-oriented architecture (soa) vs. component based architecture," Vienna University of Technology, Vienna, Tech. Rep., 2006.
- [17] S. Blanvalet, *BPEL Cookbook: Best Practices for SOA-based integration and composite applications development*. PACKT PUBLISHING, 2006.
- [18] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana, "Web services description language (wsdl) version 2.0," 2006.
- [19] BEA, IBM, Microsoft, A. SAP, and S. Systems, "Business process execution language for web services. version 1.1," 2003.
- [20] D. S. Frankel, *Model Driven Architecture: Applying MDA to Enterprise Computing*. Wiley, 2003.
- [21] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture- Practice and Promise*. Addison-Wesley, 2003.
- [22] MOF, "Meta object facility (mof) 2.0 core specification, object management group, available at www.omg.org," 2004 [retrieved: 11, 2012].
- [23] OMG, *MOF QVT Final Adopted Specification*, 2005, oMG doc. [retrieved: 11, 2012].
- [24] —, *OCL 2.0*, 2006, oMG doc. ptc/06-05-01 [retrieved: 11, 2012].
- [25] kermeta, "<http://www.kermeta.org/>, [retrieved: 11, 2012]."
- [26] Arcstyler, "Arcstyler 5.0- interactive objects." www.interactive-objects.com, 2005, [retrieved: 11, 2012].
- [27] OptimalJ, "Compuware software coporation," 2005.
- [28] OBEO, INRIA, "Atlas transformation language." <http://www.eclipse.org/atl/>, [retrieved: 11, 2012], 2005.
- [29] D. H. Akehurst, B. Bordbar, M. J. Evans, W. G. J. Howells, and K. D. McDonald-Maier, "Sitra: Simple transformations in java," in *the 9th international conference on Model Driven Engineering Languages*, ser. LNCS, vol. 4199, Italy, 2006, pp. 351–364.
- [30] IBM, Microsoft, *Web Services Business Process Execution Language (WS-BPEL) Version 2.0*, OASIS, 2007.
- [31] X. Le Guillou, M.-O. Cordier, S. Robin, and L. Rozé, "Chronicles for On-line Diagnosis of Distributed Systems," Research Report, 2008. [Online]. Available: <http://hal.inria.fr/inria-00282294/en/>, [retrieved: 11, 2012]
- [32] M. Shaw and D. Garlan, *Software architecture: perspectives on an emerging discipline*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

Effects of Web Technologies on Tourism Industry in some Southern European Countries

Sara Ficarelli, Sandra Sendra, Laura Ferrando, Jaime Lloret

Universidad Politécnica de Valencia

Camino Vera s/n, 46022, Valencia, Spain

sara.ficarelli22@gmail.com; sansenco@posgrado.upv.es; laufermo@epsg.upv.es jlloret@dcom.upv.es

Abstract— Information technologies have changed the way to plan our travel and to look at the tourism industry. This field has changed in last few years from every point of view: by the consumers, by the promotional marketing, by the commerce and the way to live the geography. Travel 2.0 is growing up in Web 2.0, which means that travel brands and users of this sector are using Internet in a much defined way. The users are not simple consumers; they become the interactive center of the tourism industry. In this paper, we will show the main effects of the growth of Information Technologies and the consequent change in tourism industry with the purpose to understand why they are so connected. First, we will consider how new technologies influence the hotel industry and the relation between users and accommodations. Then, we will analyze the position of tourism industry in the European e-commerce. Finally, we will show how social media have changed travel brands.

Keywords- *Web Services; Information and Communication Technology; Tourism Industry; e-commerce*

I. INTRODUCTION

Nowadays the development of new technologies has marked the world of tourism. Interactive multimedia platforms and the use of Internet in a new different way have changed the way of life of the entire world from the economic and socio-cultural point of views [1]. Tourism can be defined as an intensive information field and it is strictly connected to new technologies because of it. In fact the nature of tourism includes creation/transformation and transmission of information which builds a particular kind of relation between consumers and business.

Business-to-Consumer (B2C) relation became connected with Information and Communication Technologies (ICT) in two levels, the operative one (when the process located in the chain became digital) and the strategic one (referred on the relation between enterprises and competitive context) [2].

The strictly relation between travels and the ICT in based upon a reason that is important to consider: tourism is a set of services and at the same time an electronic product because destinations and reservations are intangibles businesses. So the ICT has reached, during the year, the need to facilitate processes and way of trading and communicating [3]. In few years the relation between ICT and tourism is changed in a very fast and differential way [1]. Now we are able to talk about “community” and we can use the term

“Travel 2.0”. Search information, share photos and spread your opinion about the destination is going to be more and more immediate: it is a change from a static system to a dynamic and interactive one [4]. Several new technologies as e-commerce and e-ticketing changed the way of travel. Enterprises start to optimize the policy functions, to beat down the costs and to amplify the costumers’ services. The World Wide Web (WWW) starts to become the field of the Destination Marketing Organizations which looks at the internet like an ideal medium for trading and promoting [3].

These changes allowed creating an interactive way to communicate where the knowledge is shared and the costumers are beginning to cover an important place and being more than a simple consumer. In last few years web marketing is compulsory, especially in the field of tourism where the promotion of a touristic destination in the cyberspace has become the real image of the place itself. Over recent years, the mobile phones are becoming more and more important for the places reviews, due to the geolocalisation [1] and for reaching information.

The new way to live the travel has become like a circular way: Inspiration-Research-Planning-Decision-Reservation-Travelling-Sharing. The traveler now is considered with his own marks of references (Know the single person is important) like a single consumers with their own marks of references, who uses internet in order to travel, because the world of tourism is a mix of different platforms of communications and trades [5].

In this paper, we perform an analysis on the impact of web services and the ICT on the tourism industry in some southern European countries. We also make a comparison among southern European countries and analyze the evolution of the number of users that have occurred in recent years and the main information sources that people use to plan their vacation. This work can help us to improve the way to provide information to potential visitors.

The rest of this paper is structured to follows. In Section 2, we can see some papers which analyzed the impact of the IT in the tourism industry sector. Section 3 shows the study of the main activities within the tourism industry, which is affected by the incorporation of IT. In addition, we analyze the activity of IT in the southern European countries, such as Spain, France, Portugal, Italy, and Greece. The results of this analysis are shown in Section 4. Finally, we perform a comparison and discussion of these results in Section 5. Conclusions are shown in Section 6.

II. RELATED WORKS

This section shows several works about how tourism businesses are introducing the use of IT to promote and capture the attention of their visitors.

S. Reino and B. Hay [6] investigate the use of YouTube as a tourism-marketing tool from tourism organizations point of view and the tourist perspectives. The tourism organizations regard Youtube as a useful marketing tool for the accommodation sector since it allows them to create promotional videos, which could then be visualized by people searching for them on YouTube. However, despite the large number of visitors that this platform can receive daily, approximately 30% of the analyzed videos contained tourism-related information. This value varies depending on the type of institution (public or private) and the country where the query is made. Moreover, authors comment that YouTube can offer to tourists the opportunity of searching for very specific activities, watch reviews, and seek help or advice about their destination. They predict that YouTube will keep growing in popularity, and will become an important tool to consider in this field.

The repercussion of IT on the tourism industry should be taken with care, as the result of surveys can vary depending on the type of company that is consulted. Following this philosophy, C. Berné et al. [7], discuss a possible structural change in the distribution of tourism in Spain, affected by the intensive use of IT, from the views of the intermediaries involved. The analysis reveals a predominant use of electronic media in tourism distribution channels, from the structure of tourism distribution system where IT, particularly the Internet, are a meeting point between operators which make up another as transactions central axis. Authors conclude that the use of IT does not seem to exert a significant influence on improving the quality of the tourism product and the ultimate development of best practices in the sector. The reservation centrals (CRS) have a more positive opinion about it, probably in order to justify their own presence in the value-added chain.

Víctor V. Fernández et al. [8] show us, with their study results, that there are seven major specific features of the new tourist customer profiles that appear to be associated with new technologies: shorter stays and novelty seeking, changes in levels of customer satisfaction and loyalty, influence of tourist designing new products, price as important factor in the final decision time, direct experience i-Tourism, importance of the emotional elements and Tourism prominence mail (e-Tourism). Authors say that the main purpose of the marketing website is to persuade visitors to change their attitudes about their tourism products. The marketing and promotion of tourism are in a continuous adaptation and learning process. It requires that its members are familiar with new technologies, understanding needs and expectations of the consumers.

III. ICT IN SEVERAL FIELDS OF TOURISM

In this section we will consider some research and literature that can show the new way to live the tourism and the new face of effective/current travel business.

A. Hotel industry

First of all, we can talk about the new way to look at the accommodation reservation. An example of Spanish analysis of the new kind of competition in this industry is the work, published by F. Calero et al. [9]. This work gives importance to the strategic marketing plan by surveys on touristic technologic profile and customer demand has got the aim to identify connection between technological development and accommodations. According to them new enterprises activities are strongly influenced by a bind between ICT, web services and tourism, connected with the offer and the demand. An offer that does not have a technology infrastructure is left out from the business.

The ICT, in the last years, has been changing from an information aim into a quality control. The hotel reviews published by the costumers in different websites have become more and more important. The historical one-way relation between business and consumer now is an interactive communication: the client can give information and opinions about services and act an immediate control in everything he is using. The accommodation reviews and the judges on hotels and other touristic activities are provided by many website. Tourism 2.0 seems to be in a continuous evolution and tourist enterprises must always be prepared to communicate with more expert and exigent consumer. The new phenomenon of the comparisons websites (like Tripadvisor and Booking) is accompanied to the new requirements of reliability and security which consumers seem to look for during online operations.

In this context operators try to invest stronger effort on their own websites with e-booking and e-commerce systems. In fact, the new purpose is to reduce the distance between travelers by giving important to self organization of holidays.

Geolocalisation, online promotion, good reviews, advertisements, free accessibility for getting information and transport organization are key-words to look at the online-costumers who wants to reduce costs and organize in every detail in their travel.

Security of economic transactions and researching of information are also essential in this context. The so visibility and promotion have to be accompanied to a strong operations control and a very qualified online expert [2].

From 2001 to 2011 the number of reviews on the websites like TripAdvisor, Booking.com and Expedia (the three most used) [4] has risen from 2 million to 238 million, and the forecast for 2013 is about 465 million of reviews. Fig. 1 shows the average number of reviews for the hotels between the years 2001 and 2011, taking into account the estimates for the years 2012 and 2013. Fig. 2 shows the main websites consulted by users and visitors between the years 2001 and 2011 where it is easy to see that the website most visited is "TripAdvisor". As we can see, the evolution of reviews has a rising trend that can be modeled using a 3rd degree polynomial with a correlation coefficient of

$R^2=0.9886$. The impact of this kind of services is showed by the research: 81% finds these reviews important, 46% of people who travels looking at post hotel reviews and 49% does not book without having read reviews before [4]. These data show how ICT plays an essential role in the travel planning and in the decision before the travel. It is compulsory for the travel brand to give importance to the IT for auto-promotion.

For a hotel is compulsory conform to the globalized world in order to mark a difference of the services and offer a value-added and create new connections with users. ICT brings a total communication and permits to understand the demand, controlling every action and every destination [5].

An interesting research about hotels and web services is presented by J.G. Sabater in [10]. The document is referred to the cession of technologies in the hotel industry in Spain. The author says that the ICT innovation is the key issue in Spanish hotels industry. The paper brings two hypotheses: the ICTs offer, from one side, innovation to all the aspects of the industry. From the other side ICTs permit to control the activities and bring development investment to the hotel industry. The paper underlines several conceptual points:

- Technologies used to improve the organization development.
- The access strategies of these technologies.
- The collaboration and relation with the technologies furnishes and the relevance of the organizations strategies.
- The kinds of technologies included in the IT commercialization demand in Spanish hotels are:
 - Administrative software for the commercialization.
 - Central systems for the hotel reservations.
 - Products: promotional multimedia of the services (CD-ROMs, DVDs) [9].

B. Social media and travel brands

The social media are needed in the enterprises' life which is going to be more connected with every kind of new Internet profiles. In 2011, the 100% of the travel brand has a Facebook profile and the 75% were in Twitter. Thanks to

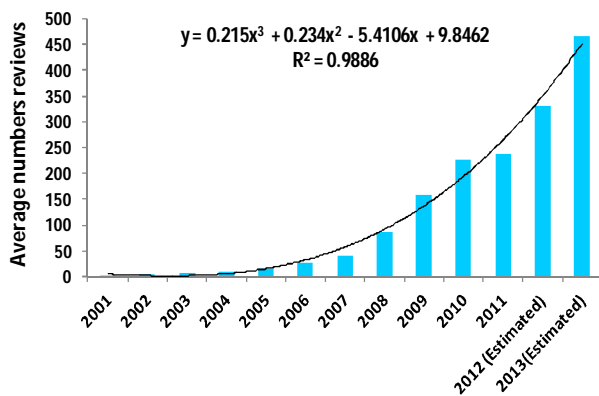


Figure 1. Average numbers reviews per hotel over the years

this kind of social media in 2011, the 33% of travel brands use their marketing budget for promote their enterprise on social media channels and the result is that they have reduced the public relation services and costs (PR Costs) by 24%. In fact every travel brand is enjoy the web because the 71% of them say that social medias have improved engagement and the 20% of the respondents of the survey talk about these products like the most successful marketing format.

In 2011, social network were the second most influential source of online traffic for travel suppliers. A new technology that is becoming essential is the *geolocalisation*. Every enterprises and every person starts to live the geography in a different way thanks to products like Google Local or Foursquare. These are ways to integrate the offline reality to the virtual one and improve the information, which a person need. For tourism it is a new way to contemplate the transports, the localization and the promoting.

The mobile phone is related to this topic because is an instrument which makes the travel more personal and it's used as a commercial vehicle operation.

In 2011, 20% of travel brands used this medium for direct sale and 25% for building awareness. The direct bookings via mobile are growth up to 30%. This sector, in a short period of time, is going to become a main way of commerce and a compulsory choice for the enterprises: mobile social network application has surged 126%, that is, 38.5 million in 2011 [11].

The web 2.0 is changing the existence of enterprises the ICT and they are so important for the enterprises development because:

- The nature of the products/services are commented by users and it's a way to meet them
- The digital communication is referred in a defined way to young people, which is an important sector of trades.
- IT permit to create relations between users and to place the products/services in the market
- IT is a way to create, apply and control a marketing and communication strategy [12].

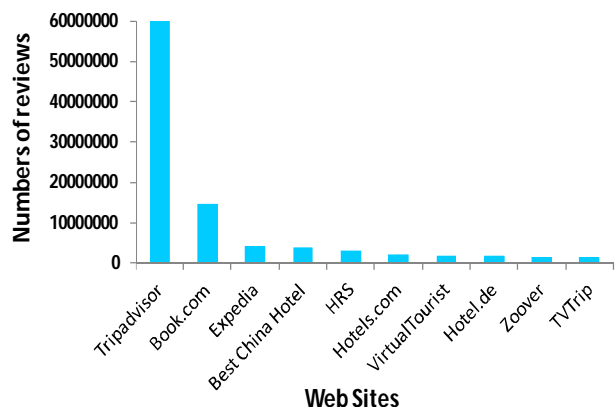


Figure 2. Top 10 reviews sites by numbers of review

C. Tourism in the European e-commerce

E-commerce on tourism is connected principally with 3 areas [2]:

- e-ticketing, manly for flights
- Hotel reservations
- OLTA (OnLine Travel Agencies) world, which includes touristic packaging

The use of the resources available on the Internet and the introduction of ICT in a country depend on many factors. Countries such as Iceland, Norway and Sweden, present the highest penetration rates of Internet (near to 90%), compared to the United States that presents rates of 77.3% [13]. The bottom of this list is occupied by countries with warmer climates, such as Spain and Hungary, with rates above 65%. If we analyze the global behavior in Europe, compared with the rest of world (see Fig. 3), we can see that Europe presents average rates of 61.3%, compared to 28.9% of the average rate for the rest of the world. Fig. 3 shows the 10 European countries with the highest number of Internet users. These data are taken until December 31, 2011 [14].

The best way to get information about the holidays destinations is to take recommendations from friends, colleagues or relatives. "Sharing" is a key word of new way intending tourism; in fact reviews and social networks are connected with principal sources chosen by travelers for get information on travels. At the second place we can see Internet websites, at the third the personal experience. Other sources for reaching information are: travel/tourist agencies and free catalogues and brochures are important. Less important than the other are: newspapers, radio and TV, paid-for guidebooks and magazines and social media sites (see Fig.4). [13]. Fig. 5 shows the most important methods used for arranging the holidays. These are the results of several inquests, with multiple responses, of the document "Attitudes of European Towards Tourism", published by the European Commission, where analyzes the behavior of European tourists in 2012 [13]. As we can see, the main method used is Internet with a 53%, almost the double of the second method most used. These data can bring us to reflect about the importance of Internet and the experience (ours experience or people who we know) in comparison with other sources. From these results, it is easy to see the importance of the ICTs and web services in the tourist sector.

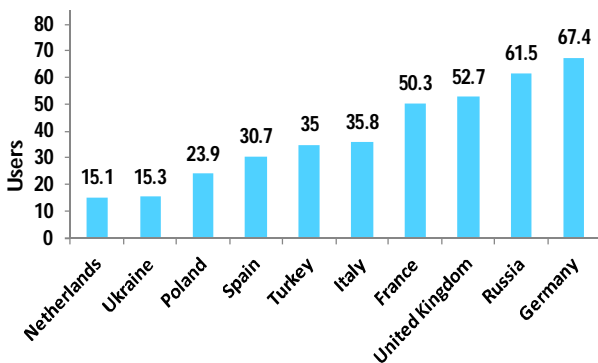


Figure 3. Top 10 internet countries in Europe.

Tourism is looking at the web because the travel sector is the principal voice of the e-commerce: in Europe it is valued on 71,3 billions of euro (29% of the total). The e-commerce permitted also to increase the interactive ways of communication between tourism and people who decide to planning travels online. In fact the World Wide Web is globally widespread: the penetration in Europe is about 48,5% and in the north of the continent we can find also country with 90% of people which use internet. This is connected with percentages of people who chose Internet as principle way for reaching information and organize travels. The e-commerce represent the world of people who buy tickets online, reserve accommodations and buy holidays services (renting cars, reserve activities...)

D. Web services and ICT in some Southern European Countries

So far we have analyzed and shown global values of Europe. European population presents several preferences in the election of resources for arranging the holiday. In this section we discuss the statistic values for southern European countries. Spain, France, Portugal, Italy and Greece are analyzed and compared. We have selected these countries because they share a tradition of direct intervention in the cultural field, contrary to anglo-saxon or scandinavian countries [15]. In addition, there are some studies which link climate issues with the choosing of tourist destinations [16]. Analyzed countries are located within the 20 most visited countries in the world [17]. But we can find other factors. South Europe is observed like a real brand from the tourist point of view of offer and demand. In one hand, Statistics show that Europe is divided in two parts taking into account how population uses Internet. Northern Europe people are keen on using the web and it is related to their way to organize their travel. Southern European countries (which concentrate the most important tourist destinations) populations use Internet less than Northern people. These differences between north and south show, from cultural and social point of view, that people have different requirements which influence the choice of using Internet as a source for organizing travels.

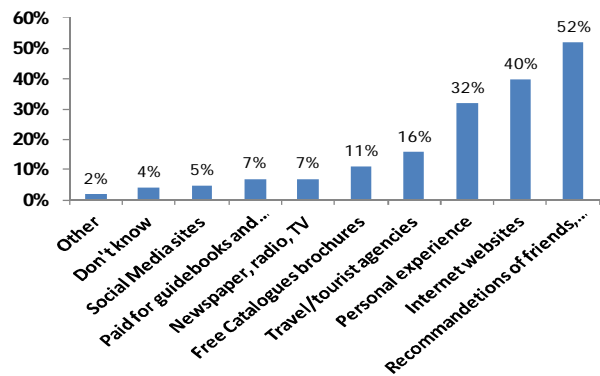


Figure 4. Information sources considered to make a decision about travel plans (multiple responses)

On the other hand, countries with a higher Public Interest Litigation (PIL) are closer to the ICT utilization for getting information and organize tourism. In the south of Europe the use of Internet is getting more and more spread while the tourist activity is becoming important. Tourist field is strictly affected by this period of economic uncertainty. In this paper we are going to analyze some countries between the most affected by the crisis. However, the unequal use of internet in different countries makes it hard to get useful information and unified, in order to be able to process it. For example, in Italy the touristic sector does not permit a general view of the touristic organization because it is divided into the 21 regions existing in the country, which obstacles to unify the offer. In Portugal and Spain the touristic inbound is increased considerably thanks to northern European countries. Their tourist force is based on the climate situation and prizes change situation in Europe because of the crisis. In 2011, Portugal and Spain has a B2C increase of the inbound and in Greece, a negative decrease. In these countries situation is worst from the outbound point of view: flows of tourism in crisis affected countries are decreased while the demand is stabilized in the North-center Europe countries. These data shows the complexity between relations in tourism field in different Europeans areas [18]. From 2008, France and Spain start to change their tourist planning offer in order to adequate themselves to the crisis emergency needs. In Spain the Spanish government and “*Tourespaña- Insituto del Turismo de España*” [19] develop a new marketing strategic plan in collaboration with tour operators and autonomy Regions, which included the positioning of the Spain Brand via online marketing and policy adaptation on new technologies. France modified the structural attitude via creating a new organism, *Atout France* [20], which give importance to the online promotion by increasing their own website and the on line communication with customers [21].

1) Spain

In 2012 we can see in which products the Spanish people decide to use internet as a commerce vehicle use by 27% of

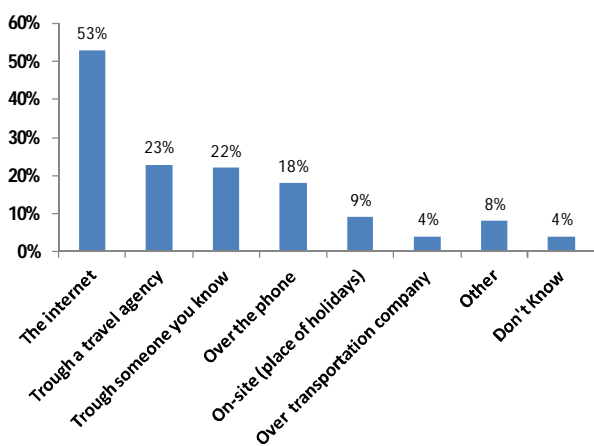


Figure 5. Methods used for arranging the holidays (multiple responses). 60% from the total number of respondents (Those who went on holiday for at least four nights in 2011)

people [22]. We can identify the 10 activities with the biggest online turnover in Spain. In the list we find travel agencies and tour operators (12,7%), air transports (12,0%), Direct Marketing (7,2%), earth transport of people (5,4%), Hazard games (4,5), artistic and sportive shows (4,1%), clothes (3,1%), advertisement (2,9%) and household appliances (2,5%) and hotels (2,1%) [23]. Fig. 6 shows the most important activities. It's easy to see how internet is important for Spanish people with reference to organized travel, looking at the direct commerce online.

2) Italy

In Italy only the 15% of people buys online but actually has got a very fast development. In the online turnover main sectors are the spare time market, (56.9%), tourism (24.8%) assurances (5,9%), electronic products (5,3%), editorial (2,3%), online shopping center (1,8%), feeding (1,2%), fashion (1,1%), house and furniture (0,3%), health care and beauty (0,2%) (See Fig. 7) [24].

In 2008 the Italian e-commerce increased for billions of euro: 75% of this is due to the tourism. Despite this we can see a slowdown in growth about 40-50% related with the sector which represent more than the 90% of online transactions: the e-ticketing and hotel reservations. The package business is increased instead of the 41% [25].

E-commerce is giving surge to the tourism industry. In 2001 Italy travel trades with credit cards online is about 6,4 billions of euro in the months of July and august, which represent 13,9% of the total. The touristic e-commerce abroad is the 51,3% of the 890 millions of euro spent by Italians for travelling in the summer months [26]. Principal destinations planned by e-commerce for Italians are France (15%), UK (11,2%) and Spain (9,3). Tourist which came from Portugal, Spain and Greece acted like a 3,4%, with an annual decrease of 16,7. Countries which generate online transactions for the “Italian product” are: USA (38,7%), France (10,7%), Belgium (10,5%), UK (9,8%), Italy (7,9%) [16].

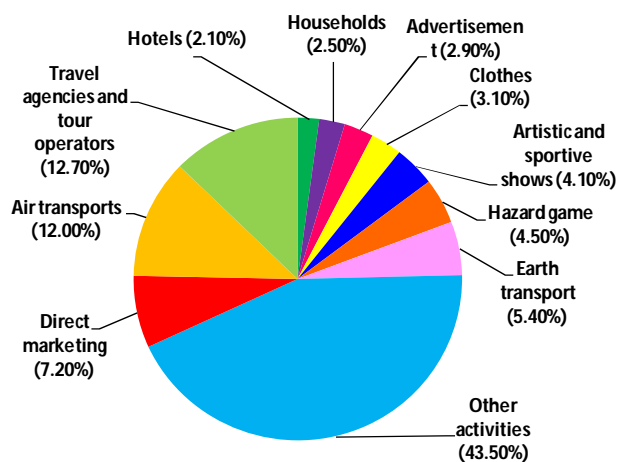


Figure 6. Ten activities with the biggest turnover in the global electronic commerce in the first 2012 trimester

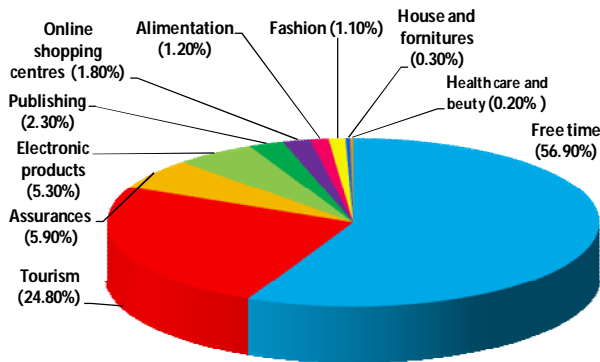


Figure 7. Italian e-commerce turnover at the end of 2011

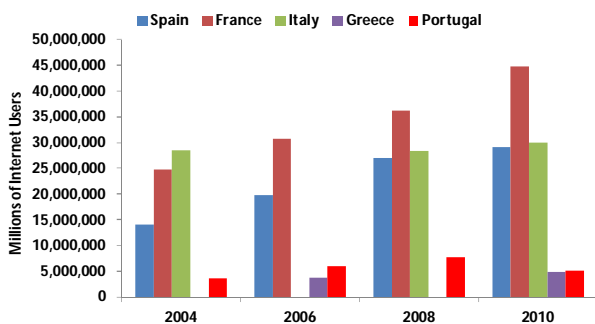


Figure 8. Millions of Internet Users

3) Greece

Now we will see how internet is located in touristic behavior of Greece. 31,2 % of Greece people think that the friends and relatives suggestion is the most important source for reaching information about travels, for 23,1% is internet, 18,5personal experience, 7,2% travel agencies and 6,7% commercial guides and journals. In Greece people who have internet access in 2010 represent from the 48,4% and the 44,4%. In general internet is use mainly by young people from 25 to 34 years old with a middle-high education level. The mobile internet access is mainly used by men (68,8%). The new kind of connection which is used is mainly the broadband (88,8%), followed by modem or ISDN, used by 10,2% of users and 5,3% connection with mobile. The ages of those traveling in Greece, is distributed as follows (see Table 1) [26]. As we can see, the people with ages between 44 and 64 years are those most travelers. They are followed by the young aged between 25 and 34 years.

4) Portugal

In 2010, the 48,8% of houses had an Internet access. Young people in Portugal are much related with new technologies; in fact smart phones and new mobile phone are going to be very used. According to Google Portugal travels are the products most bought, which represent 48% of the total of e-commerce [26].

5) France

The e-tourism in France is increasing and French people used very frequently to look at the Internet before organizing their travels in a travel agency. In 2010 81% of

TABLE I. AGES DISTRIBUTION OF INTERNET USERS IN GREECE

Ages of users	Internet Users in Greece	
	Percentage	Number of people
16-24 years old	20,6%	2.319.643
25-34 years old	32,5%	3.659.630
35-44 years old	27,1%	3.051.568
45-54 years old	14,2%	1.598.976
55-64 years old	4,4%	495.474
65-74 years old	1,2%	135.125

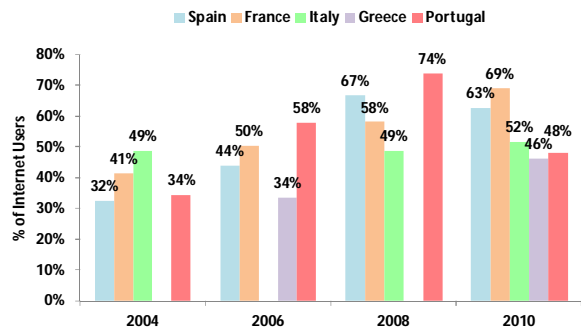


Figure 9. Percentage of Internet Users in function of the total population

French people (mainly from 25 to 34 years old, in a middle-high education level) use Internet to plan their travels. In 2010 10 millions of French people bought their holidays on web. 4% reserved their travel via smart phone or tablet. In this sector the users are very young: the average age decrease and the mobile phone reservation increase:

- 18-24 years old users: 9%.
- 25-34 years old users: 6%.
- 35-49 years old users: 3%.

This sector is used mainly by men (5,7%) instead women (2,7%) [26]. In 2012, the 56% of internet users (in France) used internet in order to travel. In other countries as Germany or U.K, this value is higher than the 64%, with a value of 71%, for U.K.

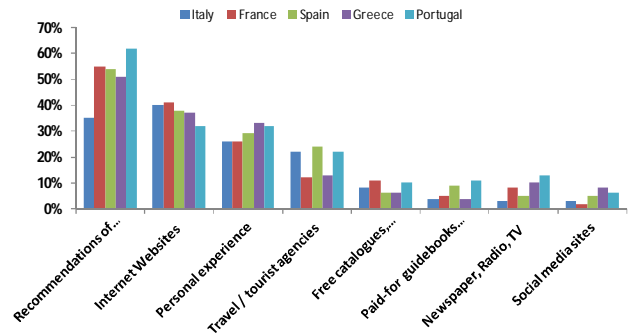


Figure 10. Main information sources to make decision about travel destination.

6) Comparison between the five countries

In order to understand the evolution of ICT in the tourism industry, we have analyzed the number of Internet users for each country that has been discussed. Figs. 8 and 9 show the evolution of the number of internet users in the five countries during the years 2004-2010. As Fig. 8 shows, in 2010, Spain and France register an increase of nearly 200%, with respect to the number of recorded users' in 2004. In the other side, Portugal registers a decrease of 28% of users in only 6 years (from almost 10 million to less than 5 million). Greece and Italy maintain the same number of Internet users during these years. During these years (with crisis), Internet is being used as a tool or vehicle to improve the countries' economy from the tourism sector, especially in Italy, Spain and France. We could not obtain data from Greece, for the years 2004 and 2008, and data for Italy for 2006.

Regarding to the information sources used in the five countries to make a decision about the travel plans [25], we can distinguish the following:

- Recommendations of friends, colleagues or similar.
- Internet Websites.
- Personal experience.
- Travel / tourist agencies.
- Free catalogues, brochures.
- Paid-for guidebooks and magazines.
- Newspaper, Radio, TV.
- Social media sites.

Fig.10 shows the main information sources that citizens of these 5 countries usually use. As we can see, that the most used resource in the majority of countries remains the recommendations of friends and family. The second information source is the Internet resources. The case of Italy shows that the first information source is the Internet resources, followed by recommendations from friends and family. Social media sites such as Internet forums, weblogs, social blogs, microblogging, wikis, social networks, podcasts, etc., are placed in the last position.

For our purpose is interesting to look at the e-commerce tendencies in Europe. In fact the e-commerce and the use of web services are strongly related with tourism, which is always connected with online trading. In Europe, the e-commerce has a good situation for the increasing in the last years: this sector represent the new trading way for the enterprises of tourism [24]. The ICT and web services are going to be widespread at the same level in every country and tourism is actually a field which is referred to online transitions [25].

IV. DISCUSSION

After this research and these statistics we can talk about the relation between tourism and internet. The possibility to buy online travel products is directly proportional to the accessibility of the country to the web world. In fact in the Northern Countries, where using internet is more widespread than in the south, we have higher numbers of users and people who use internet as vehicle of travel organizing. We can say that in Southern Countries the internet is mainly used by

you people from 25 to 34 years old, with a middle-high education and a good access to the web.

France is the first country by number of people who use internet for reaching information about the travel. Portugal is the last one: this can be compared with the possibility by the population to use internet. Generally reaching information by suggestion of people we know is the most used vehicle. This reason underlines the importance to get information by other people and look, for example, at social networks. Internet is going to be the most used way to organizing a holiday.

Portugal is the country which needs an effort to arrive the average European level of using internet for reaching information about travels.

For Italy and Spain the e-commerce is very important and, despite the crisis, they are increasing the internet accessibility. Internet in these countries is a strong changing in the field of tourism. Despite the crisis it brings promotional and marketing good consequences. It is due to the strictly relation between internet users and consumers. Relations who decrease costs and let meet the demand and the offer.

The mobile phone is going to be very important and it is used by a very young set of population (even less than 18 years old) and this is the new generation of people who will reserve everything in the travels by using mobile applications.

V. CONCLUSION

We have demonstrated that Internet provides many benefits to the field of tourism. Web navigation can bring the consumers to a high availability (24x7x365 service, 24 hours, 365 days a year), providing the possibility of reaching a heterogeneous and big quantity of information. Web is a cheaper way to plan holidays and look at different and competitive information for a more personal decision, without intermediary costs and time wasting.

The conclusions drawn from this study may serve to mitigate/harmonize the regional tourist requirements. It can also be used to adapt and improve existing portals and web services to new demands and customer needs, considering their comments. We really believe that this document can help creators and researchers of tourism websites.

The dawn of internet era enabled to go from a "funnel" situation into a "platform" one, from an oligopoly situation into a free-market one: the tourist is no more a simple actor but has become the center of the industry. Many technologies have changed the tourism industry by create a new B2C: the web marketing and websites of reviews are essential for the hotel industries, the social medias are a new vehicle of communication and trading and the tourism is located in a very central position in the e-commerce turnover. Have a look on this phenomenon helped us to achieve a better understanding of new touristic trends and new ways of marketing and promoting.

Moreover, unify the offer is a correct way to promote a specific destination areas: the aim of this work is identify specific needs of some countries in a more general optical view.

The future of the IT in B2C tourism field has some perspectives which can be referred to the present evolution. Firstly, we will assist to a total digitalization of the costumers, so the entire organization of travels will be based on the dynamic packaging phenomenon. Challenge between products and destination will grow up very strongly and tourist will change totally their way to travel and organize holidays. It is important how this natural change of these perspectives will affect the touristic jobs. Travel agencies will lose their role as intermediaries. Moreover, e-marketing and destination marketing will adapt their role for new requirements. Furthermore, enterprises will focus the personal choose on computer-knowledge and its experience.

ACKNOWLEDGMENT

Authors would like to thank Carola Giannino for her help in improving this paper and to Gerson Beltran for his comments to gather information for this paper.

REFERENCES

- [1] G. Beltran, Geolocalización, redes sociales y turismo. (2012). Available at: <http://gersonbeltran.com>. (Last access: Sep. 25, 2012).
- [2] Regione Sicilia "ICT e turismo" L'impatto di Internet e delle ICT sull'industria turistica e le sue implicazioni future Available at: http://www.thinktag.it/system/files/1151/Estratto_XV.pdf?1292012530. (Last access: Sep. 25, 2012)
- [3] World Tourism Organization Business Council (WTOBC), "Marketing Tourism Destinations Online. Strategies for the Information Age", World Tourism Organization (WTO), Madrid, 1999.
- [4] Olery, Turn guest into ambassadors. (2012). Available at: <http://www.olery.com>. (Last access: Sep. 25, 2012).
- [5] G. Beltran, Interview of "Geolocalización, redes sociales y turismo". Interviews in Social Media Day 2010. Available at: <http://gersonbeltran.com/entrevistas>. (Last access: Sep. 25, 2012)
- [6] S. Reino and B. Hay, "The Use of YouTube as a Tourism Marketing Tool". In Proceedings of the 42nd Annual Travel & Tourism Research Association Conference, London-Ontario, Canada, June 19-21, 2011.
- [7] C. Berné, M. García, M. E. García and, J. M. Múgica, "La influencia de las TIC en la estructura del sistema de distribución turística" Cuadernos de Turismo, Vol. 28, pp. 9-22. July-December, 2011. Available at: <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=39821278001>. (Last access: Sep. 25, 2012)
- [8] V. V. Fernández and A. Mihi, "Nuevas tendencias en los comportamientos de consumo de los viajeros internacionales: un análisis bajo el enfoque de la aplicación de las nuevas tecnologías en el marketing y la promoción turística". Revista del CES Felipe II, Vol. 14. June 2012. Available at: <http://www.cesfelipesegundo.com/revista/Articulos2012/Fern%C3%A1ndezMihi.pdf> (Last access: September 25, 2012)
- [9] F. Calero, E. Parra, and A. Santana, "Vigilancia tecnológica e inteligencia competitiva: un análisis de la demanda tecnológica en alojamientos turísticos en canarias", Revista de analisis turistico, no. 9, 2011.
- [10] J.G. Sabater, "La transferencia de tecnología en la industria hotelera española. El papel de los proveedores de conocimiento como fuente de innovación", Estudios turísticos, Instituto de Estudios Turísticos, no. 182, pp. 7-29, 2009.
- [11] Eye for travel. (2012). Available at: <http://www.eyefortravel.com>. (Last access: Sep. 25, 2012).
- [12] J. Celaya and P. Herrera, "Comunicación empresarial 2.0. La función de las nuevas tecnologías sociales en la estrategia de comunicación empresarial" Grupo BPMO Ediciones, 2007.
- [13] The Gallup Organisation-European Commission, "Survey on the attitudes of Europeans towards tourism", (2012). Available at: http://ec.europa.eu/enterprise/newsroom/cf/itemdetail.cfm?item_id=4093&tpa=136&tk=&lang=es. (Last access: Sep. 25, 2012).
- [14] Internet World Stats web site. Available at: <http://www.internetworldstats.com>. (Last access: Sep. 25, 2012)
- [15] A. Perry, "Impacts of climate change on tourism in the Mediterranean: Adaptive responses", Nota di Lavoro, Vol. 35, Fondazione Eni Enrico Mattei, Milan, 2000.
- [16] World Tourism Organization UNWTO, "Barómetro OMT del Turismo Mundial: Comprometidos con el turismo y con los Objetivos de Desarrollo del milenio", Vol.6, no2, Junio 2008. Available at: http://www.unwto.org/facts/eng/pdf/barometer/UNWTO_Barom08_2_sp_LR.pdf (Last access: Sep. 25, 2012)
- [17] E. Negrier, "Cultural Politics: France and South Europe" Journal of "Política y Sociedad", Vol. 44 No. 3, pp. 57-70, 2007
- [18] TTG Italia (2012) Available at: http://www.ttgitalia.com/stories/internazionale/81827_turismo_incoming_sud_europa_in_affanno_per_leuro (Last access: Sep. 25, 2012)
- [19] Tour España web site. Available at: <http://www.tourspain.es/es-es/Paginas/index.aspx> (Last access: Sep. 25, 2012)
- [20] Atout France web site. Available at: <http://www.atout-france.fr/prehome> (Last access: Sep. 25, 2012)
- [21] ISNART in collaborazione con Union Camere (2010) Available at: http://www.ontit.it/opencms/export/sites/default/ont/it/documenti/files/ONT_2010-06-08_02390.pdf (Last access: Sep. 25, 2012)
- [22] INSEM SPA. (2012). Available at: <http://www.insem.it>. (Last access: Sep. 25, 2012).
- [23] Web site of "Comisión del Mercado de las Telecomunicaciones". (2012). Available at: <http://www.cmt.es>. (Last access: Sep. 25, 2012).
- [24] Casaleggio Associati, Strategie di Rete. (2012). Available at: <http://www.casaleggio.it>. (Last access: Sep. 25, 2012).
- [25] EuroStat, European Distributors or Statistical Software. (2012). Available at: <http://www.eurostat.com>. (Last access: Sep. 25, 2012).
- Tourism National Agency of Italy website. Available at: <http://www.enit.it>. (Last access: Sep. 25, 2012).

Algorithm for Automatic Web API Composition

Yong-Ju Lee

School of Computer Information, Kyungpook National University, 386 Gajangdong, Sangju, South Korea
yongju@knu.ac.kr

Abstract—Data mashup is a special class of mashup application that combines Web APIs from several data sources to generate a new and more valuable dataset. Although the data mashup has become very popular over the last few years, there are several challenging issues when combining a large number of APIs into the data mashup, especially when composite APIs are manually integrated by mashup developers. This paper proposes a novel algorithm for automatic composition of Web APIs. The proposed algorithm consists of constructing a directed similarity graph and searching composition candidates from the graph. We construct a directed similarity graph which presents the semantic functional dependency between the inputs and the outputs of Web APIs. We generate directed acyclic graphs (DAGs) that can produce the output satisfying the desired goal. We rapidly prune APIs that are guaranteed not to involve the composition in order to produce the DAGs efficiently. The algorithm is evaluated using a collection of REST and SOAP APIs extracted from ProgrammableWeb.

Keywords—automatic composition algorithm; semantic data mashup; ontology learning method; Web API

I. INTRODUCTION

A mashup is a Web application that combines data, presentation, or functionality from several different sources to create new services. An example of the mashup is HousingMaps [1], which displays available houses in an area by combining listings from Craigslist with a display map from Google. A *data mashup* is a special class of the mashup application that combines data from several data sources (typically provided through Web APIs; these API types are usually SOAP, REST, JavaScript, XML-RPC, Atom, etc.) to generate a more meaningful dataset. Data mashups have become very popular over the last few years. For example, as of August 2012, ProgrammableWeb [2] has published more than 7000 Web APIs. Several mashup tools such as Yahoo's Pipes, IBM's Damia, and Intel's Mashmaker have been developed to enable users to create data mashups without programming knowledge.

Although the data mashup has emerged as a common technology for combining Web APIs, there are several challenging issues. First, since a portal site may have a large number of APIs available for data mashups, manually searching and composing compatible APIs can be a tedious and time-consuming task. Therefore, mashup developers wish to quickly find the desired APIs and easily integrate them without having to expend considerable programming efforts. Second, portal sites typically only support keyword search or category search. These search methods are insufficient due to their bad recall and bad precision. To

make mashups more efficiently, we need a semantic-based approach such that agents can reason about the capabilities of the APIs that permit their discovery and composition. Third, most mashup developers want to figure out all the intermediate steps needed to generate the desired mashup automatically. An infrastructure that allows users to provide some interesting or relevant composition candidates that can possibly incorporate with existing mashups is needed.

To solve the above issues, we present an algorithm for automatic discovery and composition of Web APIs using their semantic descriptions. Given a formal description of the Web API, a desired goal can be directed matched to the output of a single API. This task is called *discovery*. If the API is not found, the agent can search for two or more APIs that can be composed to satisfy the required goal. This task is called *composition*. Since the discovery is a special case of the composition where the number of APIs involved in the composition is exactly equal to one, discovery and composition can be viewed as a single problem.

We define API descriptions to syntactically describe Web APIs, and use an ontology learning method [3] to semantically describe Web APIs. We propose a Web API composition algorithm based on the ontology learning method. The proposed algorithm consists of constructing a directed similarity graph and searching composition candidates. The composition process can be described as that of generating directed acyclic graphs (DAGs) that can produce the output satisfying the desired goal, where the DAGs are gradually generated by forward-backward chaining of APIs. In order to produce the DAGs efficiently, we filter out APIs that are not useful for the composition. The main contributions from this paper are as follows:

- The paper proposes a new efficient algorithm for solving the Web API composition problem that takes semantics into account. The proposed algorithm automatically selects the individual APIs involved in the composition for a given query, without the need for manual intervention.
- Selecting and integrating APIs suitable for data mashups are critical for any mashup toolkits. We show in this paper how the characteristics of APIs can be syntactically defined and semantically described, and how to use the syntactic and semantic descriptions to aid the easy discovery and composition of Web APIs.
- A semantic-based data mashup tool is implemented for lowering the complexity of underlying programming efforts. Using this tool, the composition of APIs does not require in-depth programming knowledge. Users are able to integrate APIs with minimal training.

The rest of this paper is organized as follows. Section 2 begins by introducing our ontology learning method. Section 3 describes automatic Web API discovery and composition algorithms. Section 4 describes an implementation and experiment. Section 5 discusses related work, and Section 6 contains conclusions and future work.

II. ONTOLOGY LEARNING METHOD

The successful employment of semantic Web APIs is dependent on the availability of high-quality ontologies. The construction of such ontologies is difficult and costly, thus hampering Web API deployment. Our ontology learning method [3] automatically generates ontologies from Web API descriptions and their underlying semantics.

A. Parameter Clustering Technique

We have developed a parameter clustering technique to derive several *semantically meaningful concepts* from API parameters. We consider the syntactic information that resides in the API descriptions, and apply a mining algorithm to obtain their underlying semantics. The main idea is to measure the co-occurrence of terms and cluster the terms into a set of concepts. Formally, we can define an API as follows:

Definition 1: A Web API $W = \langle I, O \rangle$ where I is the input and O is the output. Each input and output contains a set of parameters for the API.

The input/output parameters are often combined as a sequence of several terms. We utilize a heuristic as the basis of our clustering, in that the terms tend to express the same concept if they frequently occur together. This allows us to cluster terms by exploiting the conditional probability of their occurrences in the input and output of Web APIs, specifically we are interested in the *association rules* [4]. We use the agglomerative hierarchical clustering algorithm to turn the set of terms $T = \{t_1, t_2, \dots, t_m\}$ into the concepts $C = \{c_1, c_2, \dots, c_n\}$. For example, the terms $\{\text{zip}, \text{city}, \text{area}, \text{state}\}$ can be treated as one concept, they are grouped into one cluster.

B. Pattern Analysis Technique

The pattern analysis technique captures *relationships between the terms* contained in a parameter, and matches the parameters if both terms are similar and the relationships are equivalent. This approach is derived from the observation that people employ similar patterns when composing a parameter out of multiple terms. Based on the experimental observations, the relationships between the terms are defined in Table 1. Two ontological concepts are matched if and only if one of the following is true; (1) one concept is a *property* of the other concept, and (2) one concept is a *subclass* of the other concept.

From the above rules, an agent would be able to find a match based on the similarities of the API. For example, assume that a parameter `CityName` was to be compared against another parameter `CodeOfCity`. The keyword search would not count these as a possible match. However, if the `City` term had the relationships “X **propertyOf** Y” in

its pattern rule, the matching logic will return a matching score because these two parameters are closely related (perhaps using the rules “CityName **propertyOf** City” and “CodeOfCity **propertyOf** City”).

TABLE I. RELATIONSHIPS BETWEEN TERMS

No	Pattern	Relationships
1	Noun ₁ +Noun ₂	Parameter propertyOf Noun ₁
2	Adjective+Noun	Parameter subClassOf Noun
3	Verb+Noun	Parameter subClassOf Noun
4	Noun ₁ +Noun ₂ +Noun ₃	Parameter propertyOf Noun ₁
5	Noun ₁ +Preposition+Noun ₂	Parameter propertyOf Noun ₂

C. Semantic Matching Technique

The semantic matching technique estimates the similarity of the input and output by considering the underlying concepts the input/output parameters cover. Formally, we describe the input as a vector $I = \langle p_i, C_i \rangle$ (similarly, the output can be represented in the form $O = \langle p_o, C_o \rangle$), where p_i is the set of input parameters and C_i is the concept that is associated with p_i . Then, the similarity of the input can be found using the following two steps (the output can be processed in a similar fashion); (1) we split p_i into a set of terms, we then find synonyms for these terms, and (2) we replace each term with its corresponding concepts, and then compute a similarity score.

The similarity score is defined to select the best matches for the given input. Consider a pair of candidate parameters p_i and p_j , the similarity between p_i and p_j is given by the following formula:

$$\text{Sim}(p_i, p_j) = \frac{2 \times \|\text{Match}(p_i, p_j)\|}{m+n}$$

where m and n denote the number of valid terms in parameters, $\|\text{Match}(p_i, p_j)\|$ returns the number of matching terms. Here, the similarity of each parameter is calculated by the best matching parameter that has a larger number of semantically related terms. The overall similarity is computed by a linear combination [3] to combine the similarity of each parameter.

Since existing matching techniques based on the clustering consider all terms in a cluster as an equivalent concept and ignore any hierarchical relationships between the terms, matches might exist that are irrelevant to the user's intention (i.e., false positives). Thus, a pruning process is necessary to improve the precision of the results. The basic idea is to improve the precision of the matching technique by applying the pattern relationships defined in Table 1. For details, readers may refer to our previous work [3].

III. WEB API DISCOVERY AND COMPOSITION

A. Discovery Problem

Given a query and a collection of APIs stored in the registry, automatically finding an API from the registry that

matches the query requirement is the Web API discovery problem. For example, we are looking for an API to search a hotel. Table 2 shows the input/output parameters of a query and an API. In this example a Web API W satisfy the query Q . Q requires `HotelName` as the output and W produces `HotelName` and `ConfirmNumber`. The extra output produced can be ignored. W requires `CountryCode` and `NameOfCity` as the input and Q provides `CountryID`, `StateName`, and `CityName` as the input. An API parameter can be matched with the other parameter only if there is a semantic relationship between them. Here, although `CountryCode` and `CountryID` are different forms, they have the same semantics since they are referred to the same concept. Also `NameOfCity` and `CityName` have the same semantics since they are properties of the same object (i.e., `City`). Therefore, the agent is able to infer that Q and W input parameters have semantically the same classes.

TABLE II. EXAMPLE FOR DISCOVERY PROBLEM

API	Input Parameters	Output Parameters
Q	CountryID, StateName, CityName	HotelName
W	CountryCode, NameOfCity	HotelName, ConfirmNumber

We describe an *automatic Web API discovery algorithm* similar to the one in [5]. An API matches a query when an API is sufficiently similar to the query. This means that we need to allow the agent to perform matches that recognize the degree of similarity between APIs and the query. We define the matching criteria as follows:

Definition 2: An API W matches a query Q when all the output parameters of Q are matched by the output parameters of W , and all the input parameters of W are matched by the input parameters of Q .

Definition 2 guarantees that the API found satisfies the needs of the query, and the query provides all the input parameters that the API needs to operate correctly. Our discovery algorithm is shown in Algorithm 1. This algorithm adopts strategies that rapidly prune APIs that are guaranteed not to match the query, thus improving the efficiency of the system. A query is matched against all APIs stored in the registry. A match between a query and an API consists of matching all the output parameters of the query against the output parameters of the API; and all the input parameters of the API against the input parameters of the query. If one of the query's output parameters is not matched by any of the API's output, the match fails. Matching between inputs is computed by the same process, but with the order of the query and API reversed. The similarity score of a match between two parameters is calculated by the semantic matching technique described in the previous section. The APIs are returned in the descending order of similarity scores.

Algorithm 1: Discovery Algorithm

```
//input: query (Q), APIs
//output: matched APIs
for all APIs
  if Matching(Q, API) then result.append(API)
return Sort(result)
Matching(Q, API)
  SemanticMatch(Q.O, API.O)
  SemanticMatch(API.I, Q.I)
```

B. Composition Problem

Given a query and a collection of APIs, in case a matching API is not found, searching a sequence of APIs that can be composed together is the composition problem of Web APIs. It means that the output generated by one API can be accepted as the input of another API. For example, we are looking for APIs to find a hotel's location. Table 3 shows the input/output parameters of a query Q , and two Web APIs W_1 and W_2 in the registry. Suppose the agent cannot find a single API that matches the criteria, then it composes n APIs from the set of Web APIs available in the registry. In this table, W_1 returns `HotelName` as the output. W_2 receives it as the input and returns `Location` as the result. So, the subsequent W_2 may use the output produced by the preceding W_1 as the input.

TABLE III. EXAMPLE FOR COMPOSITION PROBLEM

API	Input Parameters	Output Parameters
Q	CountryID, StateName, CityName	Location
W_1	CountryCode, NameOfCity	ConfirmNumber, HotelName
W_2	HotelName	Location

Now we can define the Web API composition problem as follows:

Definition 3: If an API W_1 can produce O_1 as its output parameters and an API W_2 can consume O_1 as input parameters, we can conclude that W_1 and W_2 are *composable*. Then, the Web API composition problem can be defined as automatically finding a DAG of APIs from the registry.

We describe a Web API as $\langle W.I, W.O \rangle$ and a query as $\langle Q.I, Q.O \rangle$. A composition is valid if the following conditions are satisfied:

- 1) $\exists W_i (Q.I \supseteq W_i.I)$
- 2) $\exists W_j (Q.O \subseteq W_j.O)$
- 3) $\forall W_i, W_j$, there exists at least a path from W_i to W_j .

In other words, the APIs in the first stage of the composition can only use the query input parameters. The outputs produced by the APIs in the last stage of the composition should contain all the output parameters that the query requires to be produced. The output from an API at any

stage in the composition should be able to provide as the input to the next API.

The composition problem is just achieving a desired goal from the initial request, while not making it know the underlying composition details. The mashup developers can now simply describe a goal in form of the query, and submit the requirement to our system. If the desired goal can be directly matched to the output of a single Web API, the composition problem reduces to the discovery problem. Otherwise, it can be accomplished by searching a sequence of APIs that can produce the desired output. Such sequence composition of APIs can be viewed as a searching DAG that can be constructed from an initially given query. In particular, when all nodes in the graph have not more than one incoming edge and not more than one outgoing edge, the problem reduces to a linearly linked APIs problem. Because the discovery problem is a simple case of the composition where the number of APIs involved in the composition is exactly equal to one, the discovery and composition can be viewed as a single problem.

C. Constructing Directed Similarity Graph

In order to speed up the calculation of possible composition plans, we use a pre-computed directed similarity graph that chains the output of one API into the input of another API. The connection of the nodes is based on the semantic similarity between the output and input of the nodes. Algorithm 2 illustrates the construction procedure for the graph. At the beginning, we assign each API in the registry to vertexes iteratively. We then establish edges between the vertexes. For each vertex v_i , we check whether its corresponding output can be accepted as an input by a v_j by computing the similarity score. If the output of v_i is semantically similar to the input of v_j (i.e., $\text{Sim}(v_i.O, v_j.I) > 0$), then we add a directed edge from v_i to v_j (in the reverse direction) and assign a similarity score. We also check if there exists a vertex v_j , whose output can be consumed by v_i as an input, in the similar manner. After constructing the directed similarity graph, we solve the composition problem within this graph. This initial graph is dynamically modified if new APIs become available.

Algorithm 2: Graph Construction Algorithm

```

//input: APIs
//output: a directed similarity graph
for all APIs
     $v_i = \text{addVertex}(\text{API})$ 
for each  $v_i \in V$ 
    for each  $v_j \in V$ 
        if  $\text{Sim}(v_i.O, v_j.I) > 0$  then  $\text{addEdge}(v_i, v_j)$ 
        if  $\text{Sim}(v_i.I, v_j.O) > 0$  then  $\text{addEdge}(v_j, v_i)$ 
    
```

D. Graph-based Composition Algorithm

Our graph-based composition algorithm can be described as that of generating DAGs that can produce the output

satisfying the desired goal. In order to produce the DAGs efficiently, we rapidly filter out APIs that are not useful for the composition. We extend our discovery algorithm to handle the composition problem. The algorithm is based on a modified Breath-First Search (BFS) algorithm [6] which can find a shortest path from a source vertex to a target vertex. We solve the composition problem in four main stages: searching sub-graphs, adding start nodes, validating candidates, and ranking candidates.

Searching sub-graphs: First, we search the API registry about any API that has all the output parameters of the query (we call “last nodes”), and any API that has at least one of the input parameters of the query (we call “first nodes”). After this searching it is assumed that non empty sets are obtained for the first and last nodes. The next is to create n -ary trees for every last node by visiting all the nodes connected to a particular last node. Such tree is constructed by recursively including nodes and edges from the directed similarity graph until we reach the first nodes. We use the BFS algorithm to solve this problem. Now we can find all the possible composition candidates from the trees. Figure 1 shows a general overview of the query and the matching APIs before constructing the overall composition plans.

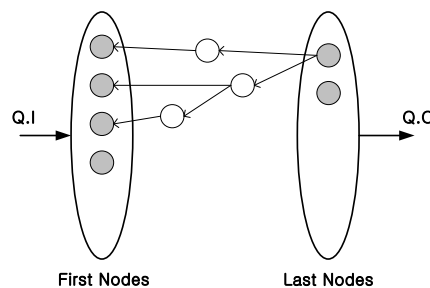


Figure 1. General Overview of Query and Matching APIs

Adding start nodes: In this stage, a start node is added to each of the trees. The start node is a special dummy node for a dynamically created API, namely the API that provides the input of the query. The start node is represented as $W_0 = \langle \emptyset, Q.I \rangle$, namely W_0 is an API in a tree with no input, having only an output. Finding a possible composition candidate consists in generating a DAG from the start node to the last node in the trees. When a possible composition candidate has been found, all the nodes participating in the composition should be validated in the next stage.

Validating candidates: A possible composition candidate is valid if all nodes in the composition can be executed (non-)sequentially in order to produce the desired results. This validating is done by starting from the start node working our way backwards. At this point, first nodes consist of all the APIs such that all their inputs are provided by the start node. Let O_1 be a union of all outputs produced from the first nodes in the composition, and I_1 (i.e., $Q.I$) be the query input. Inputs for the second nodes are all the outputs

produced by the previous nodes and the query input, i.e., $I_2 = O_1 \cup I_1$. The combination I_2 will be the available input for the next nodes. This transition (i.e., $I_{i+1} = O_i \cup I_i$) is repeated until the last node is reached, removing redundant nodes which do not contribute to the optimal path at each step.

Ranking candidates: A DAG is considered as a composition candidate only if it meets the requirements of the output and input described in the query. It means all output parameters of the query must be obtained, and partly or fully the input parameters of the query must be consumed. After a composition candidate has been found, we gather all the similarity data from the edges involved in the composition in order to compute a similarity score. This score is calculated by the average value of all the similarity data related to the edges, and the ranking of the composition candidate is determined by the score. The list of composition candidates is ordered according to this ranking score and the head of the list is considered the best, recommended option for the user. Algorithm 3 illustrates our graph-based composition algorithm.

Algorithm 3: Composition Algorithm

```

//input: query (Q), a directed similarity graph
//output: ranked composition candidates
if SemanticMatch(Q.O, API.O) is empty then fail
if SemanticMatch(API.I, Q.I) is empty then fail
for each last node
    Call BFS algorithm
    Create n-ary trees
for each tree
    Adding a start node to the tree
    Generating a DAG from start node to last node
    //Validating possible composition candidates
     $i = 1, I_i = Q.I$ 
     $L_i = \text{NextApiList}(i)$ 
    while Not (last node  $\wedge \forall v_i \equiv \emptyset$ )
         $O_i = \text{UnionAllOutputs}(L_i)$ 
         $I_{i+1} = O_i \cup I_i$ 
         $L_{i+1} = \text{NextApiList}(i+1)$ 
        Removing redundant nodes
         $i = i+1$ 
    endwhile
endfor
Ranking composition candidates
    
```

IV. IMPLEMENTATION AND EXPERIMENT

We developed a semantic-based data mashup tool. The system architecture is shown in Figure 2. The composition planner is responsible for planning to achieve the composition relevant to the desired goal. It captures the current composition states and dynamically composes relevant APIs that can be added to the mashup. The mashup engine interprets the composition of corresponding APIs and displays the immediate results. In the graphical user interface (GUI), mashup developers can obtain the immediate composition results

visually, and iteratively refine their goals until the final results satisfying. The ontology learning method automatically builds semantic ontologies from Web API descriptions.

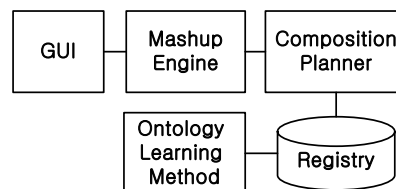


Figure 2. System Architecture

To experiment with the data mashup tool we extracted a collection of REST and SOAP APIs from Programmable-Web. To avoid potential bias, we chose different APIs from different domains. We first collected a subset which associated REST APIs for three domains: weather, travel, and mapping. This set contains 63 APIs. Next, we collected a subset containing 17 SOAP APIs from three domains: zip-code, location, and search. In Figure 3, we show a directed similarity graph which obtained from our experimental dataset. The graph consists of 80 nodes and 123 edges.

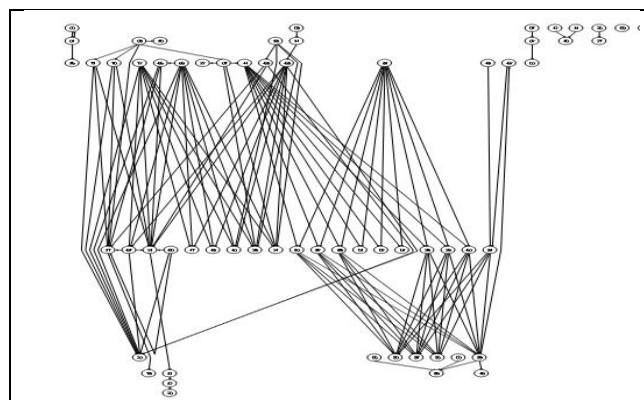


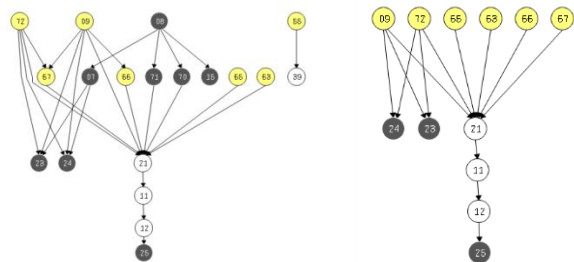
Figure 3. Directed Similarity Graph

A possible query for the Web API composition is given as follows: $Q.I = \{\text{zipcode}\}$, $Q.O = \{\text{city, latitude, longitude}\}$. The composition result is exemplified by part of the directed similarity graph as shown in Figure 4. From the registry our engine has discovered 8 last nodes (dark grey circles) and 7 first nodes (light grey circles). We call the BFS algorithm and create an *n*-ary tree for each last node. This is repeated until all the last nodes are reached.

A total of 3 possible composition candidates have been automatically generated from the graph. As we have mentioned in Section 3.D, a start node W_0 is added to each tree and the validation of candidates is performed for optimal paths. After running the validation, final composition candidates are selected and similarity scores are calculated. In Table 4, we list these ranked composition candidates.

To evaluate our composition quality, we check how many of desired goals are captured by the composition algorithm. We can observe that two third of all the recommen-

ded results in Table 4 have desired or relevant goals. Although the 3rd ranking result turns out to be invalid as it does not satisfy the user requirement, top 2 ranking results have desired composition plans. These results have shown that our algorithm can generate most user desired outputs.



(a) Discovered last and first nodes (b) *n*-ary trees
Figure 4. Result of Graph-Based Composition Algorithm

TABLE IV. LIST OF RANKED COMPOSITION CANDIDATES

Rank	Score	DAG
1	0.625	$W_0 \rightarrow (9, 72) \rightarrow 23$
2	0.550	$W_0 \rightarrow (9, 72) \rightarrow 24$
3	0.222	$W_0 \rightarrow (9, 72, 65, 66, 67) \rightarrow 21 \rightarrow 11 \rightarrow 12 \rightarrow 25$

V. RELATED WORK

Most researches handling the automatic composition problem have been focusing on the composition of SOAP-based Web services. Many various techniques have been used for this study, such as graph-based search algorithm [7] and AI planning [8]. However, the work presented in this paper is not limited to composing SOAP-based Web services, but also considers REST, JavaScript, XML-RPC, and Atom Web APIs.

The use of graph-based search algorithms to solve the composition problem has been studied before. Kona et al. [7] propose an automatic composition algorithm for semantic Web services. Rodriguez-Mier et al. [9] propose a heuristic-based search algorithm for automatic Web service composition. Shiaa et al. [10] present an incremental graph-based approach to automatic service composition. These works are similar to our study. However, they cannot find an optimal solution, and do not support various Web API protocols.

We recently proposed an automatic Web API composition algorithm [11] to handle the sequential composition problem. This paper is an extension of our previous work and focuses on the (non-)sequential composition that can be represented in the form of directed acyclic graphs (DAGs). This is the most general case of the Web API composition.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents an algorithm for the automatic Web API composition. This algorithm is based on the graph-based approach, where composition candidates are gradually

generated by forward-backward chaining of APIs. Our algorithm can get optimal plans by applying strategies that rapidly prune APIs that are guaranteed not to match the query. A key issue is how to locate the desired APIs. The efficient discovery can play a crucial role in conducting further API composition. We define API descriptions that syntactically describe Web APIs, and use an ontology learning method that semantically describes APIs. These syntactic and semantic descriptions allow the agent to automate the composition of Web APIs.

Our future work is focusing on the investigation of the performance and scalability measures for the proposed graph-based composition algorithm. By this we aim to optimize the functionality of our system. We are also exploring various optimization techniques that can apply to the algorithm. For example, a heuristic AI planning technique can be used to find an optimized solution with a minimal number of paths. The use of dynamic optimization techniques over the graph helps greatly in obtaining the effectiveness and efficiency of our approach.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (No. 2010-0008303).

REFERENCES

- [1] <http://www.housingmaps.com>
- [2] <http://www.programmableweb.com>
- [3] Y. J. Lee and J. H. Kim, "Semantically Enabled Data Mashups using Ontology Learning Method for Web APIs," Proceedings of the 2012 Computing, Communications and Applications Conference, 2012.
- [4] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proceedings of the 1993 ACM-SIGMOD International Conference Management of Data, 1993.
- [5] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities," Proceedings of the International Semantic Web Conference (ISWC), 2002.
- [6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algorithms (Second Edition), MIT Press, 2001.
- [7] K. Kona, A. Bansal, M. Blake, and G. Gupta, "Generalized Semantics-based Service Composition," Proceedings of the IEEE International Conference on Web Services (ICWS), 2008.
- [8] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN Planning for Web Service Composition using SHOP2," Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, No. 4, pp. 377-396, 2004.
- [9] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "Automatic Web Service Composition with a Heuristic-based Search Algorithm," Proceedings of the International Semantic Web Conference (ISWC), 2011.
- [10] M. Shiaa, J. Fladmark, and B. Thiell, "An Incremental Graph-based Approach to Automatic Service Composition," Proceedings of the International Semantic Web Conference (ISWC), 2008.
- [11] Y. J. Lee and J. S. Kim, "Automatic Web API Composition for Semantic Data Mashups," Proceedings of the 4th International Conference on Computational Intelligence and Communication Networks (CICN), 2012.

Describing Semantics of 3D Web Content with RDFa

Jakub Flotyński, Krzysztof Walczak
Poznań University of Economics, Poland
email: {flotyński, walczak}@kti.ue.poznan.pl

Abstract—The paper presents a method of describing semantics of 3D web content with RDFa—Resource Description Framework in Attributes. Dependencies between 3D web components are typically more complex than dependencies between standard web pages as they may relate to different aspects of the 3D content—spatial, temporal, structural, logical and behavioural. Semantic Web standards help in making data understandable and processable for both humans and computers. RDFa is an RDF-compliant standard designed for creating semantic descriptions embedded into web resources, but it has been indented mostly for 2D web pages and not for 3D web content. The main contribution of this paper is a method of creating lightweight attribute-based built-in semantic descriptions of X3D web content. The method utilizes the standard syntax and structure of X3D documents providing a mapping of RDFa attributes to metadata nodes in 3D models. Due to the use of the standardized solutions, the proposed method enables flexible semantic descriptions of content for use in a variety of 3D applications on the web.

Index Terms—3D content, semantic description, X3D, RDFa, 3D web.

I. INTRODUCTION

Interactive 3D technologies enable significant progress in the quality and functionality of human-computer interfaces. The widespread use of interactive 3D technologies, including virtual reality (VR) and augmented reality (AR), has been recently enabled by significant progress in computing hardware performance, increasing availability of versatile input-output devices as well as rapid growth in the available network bandwidth. However, the potential of 3D/VR/AR technologies in everyday applications can be fully exploited only if accompanied by the development of efficient and easy-to-use methods of creation, publication and sharing of interactive 3D multimedia content.

Building, searching and combining distributed three-dimensional interactive content is a much more complex and challenging task than in the case of standard web pages. The relationships between components of an interactive three-dimensional virtual scene may include, in addition to its basic meaning and presentation form, spatial, temporal, structural, logical, and behavioural aspects.

Opportunities for widespread dissemination of 3D content may be significantly increased by applying the Semantic Web approach. Research on the Semantic Web has been initiated by T. Berners-Lee and the W3C (World-Wide Web Consortium) in 2001. This research aims at evolutionary development of the current web towards a distributed semantic database, linking structured content and documents. Semantic description

of web content makes it understandable for both humans and computers, achieving a new quality in building web applications that can "understand" the meaning of particular components of content and services, as well as their relationships, leading to much better methods of searching, reasoning, combining and presenting web content.

The Resource Description Framework (RDF) [1] has been developed as the foundation of the Semantic Web, enabling semantic descriptions of various types of web resources. The Resource Description Framework in Attributes (RDFa) [2] is an RDF-compliant solution designed for creating lightweight semantic descriptions of web content with attributes built into described documents.

To enable 3D content description on the web, a number of proprietary data formats have been devised. In contrast to them, the Virtual Reality Modelling Language (VRML) [3] and its successor—the Extensible 3D (X3D) [4] have been developed by the Web3D Consortium as open standards. The openness determines the common use of X3D in a variety of applications, as well as attempts to combine it with other open standards, in particular for the Semantic Web. Currently, X3D provides basic mechanisms for including metadata into 3D models, but it does not standardize creation of semantic descriptions of resources. In turn, RDFa is intended mostly for 2D web pages and not for 3D web content. Embedding semantics directly into 3D content has several important advantages in comparison to decoupling semantics from 3D models. In particular, this enables more concise semantic descriptions, faster and less complicated authoring and analysis of semantically described 3D content, and permits storing the 3D content in simpler databases.

The main contribution of this paper is a method of creating lightweight attribute-based built-in semantic descriptions of X3D web content. The method utilizes the standard syntax and structure of X3D documents, providing a mapping of RDFa attributes to metadata nodes in 3D models. Using the standard syntax and structure of X3D documents preserves the compatibility of the proposed approach with available X3D browsers. Due to the use of the standardized solutions, the method enables flexible semantic descriptions and widespread dissemination of content for use in a variety of 3D applications on the web.

The remainder of this paper is structured as follows. Section II presents the X3D standard in terms of metadata description. Section III provides an overview of the state of the art in the

domain of semantic descriptions of web resources, in particular 3D web content. Section IV presents a mapping between RDFa attributes and X3D metadata nodes that enables lightweight semantic descriptions of 3D web content. Finally, Section V concludes the paper and indicates future works.

II. METADATA DESCRIPTION IN X3D

There are two types of metadata descriptions in X3D documents: metadata describing the whole X3D document and metadata describing 3D content included in the document—elements reflecting the geometry, appearance and behaviour of 3D objects in a scene.

Example X3D content described by metadata is presented in Listing 1. The first group of metadata elements—describing the X3D document—are contained in the head—the first X3D element preceding the Scene node. It may include metadata indicating additional required components (line 2) and expressing the semantics using [name, content] tuples (3-4)—alike in (X)HTML documents.

Listing 1. Example X3D content with metadata

```

1 <X3D ... >
2 <head><component name='Geospatial' level='1' />
3 <meta name='title' content='Sculpture' />
4 <meta name='subject' content='http://.../sc1t.html' />
5 </head>
6 <Shape><Sphere ... />
7 <Appearance><Material ... /></Appearance>
8 <MetadataSet name='example_metadata' reference='http://
  www.web3d.org/spec_editors/abstract/Part01/
  components/core.html#MetadataSet' containerField='
  metadata' >
9 <MetadataString name='creator' reference='http://purl
  .org/dc/elements/1.1/creator' value='http://www.
  kt.ue.poznan.pl/' containerField='value' />
10 <MetadataString name='description' reference='http
  ://.../description' value='Example sculpture'
  containerField='value' />
11 <MetadataFloat name='Mass' reference='http://www.qudt
  .org/qudt/owl/1.0.0/quantity/Instances.html#Mass'
  value='0.5' containerField='value' >
12 <MetadataString name='Unity' reference='http
  ://.../#Unity' value='http://.../#Kilogram'
  containerField='metadata' />
13 </MetadataFloat>
14 </MetadataSet>
15 </Shape>...
  
```

The second group of metadata elements are structured according to the abstract X3DMetadataObject interface defined in the X3D specification [4]. The interface is inherited by concrete metadata nodes of different types: integer, float, double, string, as presented in Fig. 1. These nodes derive two attributes from the X3DMetadataObject: the name of the metadata field and an optional reference to a specification defining the unambiguous field name. The additional value attribute is an array of values of the appropriate type. A specific node is the MetadataSet containing an array of metadata nodes of different types. Besides inheriting from the X3DMetadataObject, the metadata nodes are also descendants of the abstract X3DNode that includes an X3DMetadataObject element. Hence, each metadata node may be described by a nested metadata sub-node. Furthermore, all elements included in the X3D document (concrete nodes) implement the X3DNode interface, thus all elements of the described virtual scene may have metadata specified.

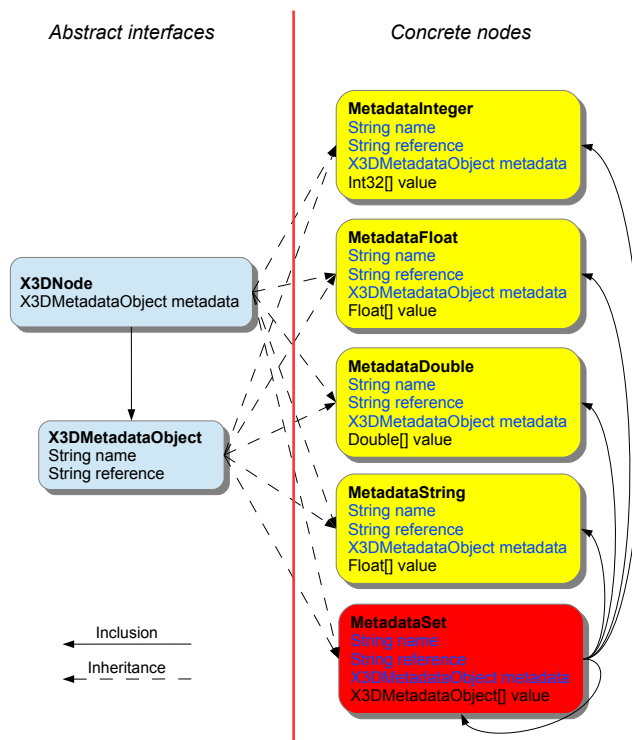


Fig. 1. X3D interfaces and metadata nodes

In the example in Listing 1, the shape is described by three elements: geometry, appearance and metadata. The whole semantics is enclosed in the MetadataSet and includes both simple (subject, creator, description—lines 9-10) as well as complex (Mass, 11) nested nodes. Every node is assigned a name and a reference to the name definition (a web document). The containerField associates the node with the appropriate field of its parent node.

III. SEMANTIC DESCRIPTIONS OF WEB CONTENT

In this section, the state of the art concerning semantic descriptions of web content is presented. In particular, basic techniques for describing semantics of web documents, methods of attribute-based semantic descriptions as well as semantic descriptions of 3D objects are considered. The first two domains are currently closely related to web pages, while the last one constitutes an emerging field of research on semantically described virtual and augmented reality.

A. Foundations for the Semantic Web

The primary technique for data semantics description on the web is the Resource Description Framework (RDF) [1]—a standard devised by the W3C. RDF introduces general rules for making statements about resources. Each statement is comprised of three elements: a subject (a resource described by the statement), a predicate (a property of the subject) and an object (the value of the property).

An example statement expressed in RDF is: "Bob (subject) likes (predicate) shopping (object)". According to the RDF specification [1], the subject may be either a resource identified by a URI (not necessary accessible via HTTP) or a blank node.

The object may be either a resource with a URI or a literal. To ensure unambiguous representation of the relationship between the subject and the resource, the predicate must have a URI assigned. RDF data sets may be encoded in different formats such as XML, N-Triples, Turtle and JSON.

RDF introduces classes (as types of resources), containers and lists to provide basic concepts for semantic descriptions. However, these notions are often insufficient for describing the semantics of complex resources. To overcome limitations of RDF, the RDF Schema (RDFS) [5] and the Web Ontology Language (OWL) [6] have been proposed as W3C standards based on RDF, providing higher expressiveness for semantic descriptions of web resources, e.g., hierarchy of classes and properties, constraints, property restrictions as well as operations on sets.

While RDF and RDF-based techniques permit creation of ontologies and knowledge bases, SPARQL [7] is a language for querying RDF data sources. To provide a common space for identifiers of resources and properties on the web, a number of ontologies and knowledge bases have been proposed for various domains, e.g., describing relationships between people [8], media resources [9], images, audio, video [10], quantities, units, dimensions [11] and chemical compounds [12].

B. Attribute-based methods of creating semantic descriptions

It is often desirable to combine both resources and their semantics in a single web document, e.g., in a web profile with personal data and relationships between people. In such cases, web documents (e.g., (X)HTML) may be enriched with additional attributes describing data semantics.

Among a few approaches to attribute-based semantic descriptions of web content, the Resource Description Framework in Attributes (RDFa) is the most powerful one and has been standardized by the W3C. RDFa [2] defines a set of markup attributes extending web documents with semantic descriptions compliant with RDF:

- 1) *about*, *src*—the URI of a subject (external or embedded into the document) described by the metadata;
- 2) *typeof*—a list of types of the subject;
- 3) *property*, *rel*, *rev*—a list of predicates specifying properties, relationships and reverse relationships between resources;
- 4) *href*, *resource*—the navigable/non-navigable URI of an object resource;
- 5) *content*—an object resource specified as a literal overriding the value of the element when using the *property* attribute;
- 6) *datatype*—the optional data type of the literal, that may be specified for the *property* value;
- 7) *vocab*, *prefix*—a vocabulary/list of prefixes used to abbreviate URIs of resources and properties specified in the metadata;
- 8) *inList*—a list of literals/URIs associated with the predicate.

An example web page described with RDFa attributes is presented in Listing 2. The document is described by metadata

in the head element (*title* and *creator*—lines 2-4). The presented object (a sculpture) has a URI and a type (5), properties (*subject* and *Mass*, 7-8) as well as a relationship with an external resource (an image—6). In the document, prefixes to global (1) and local (5) namespaces are defined and used.

Listing 2. An example web page with RDFa attributes

```

<html prefix="dc:_http://purl.org/dc/terms/"> 1
<head><title property="dc:title">Exhibition</title> 2
<meta name="dc:creator" content="DIT" /> 3
</head> 4
<body><div about="http://example.org/sculpture" typeof=" 5
http://.../sculpture" prefix="dcimtype:_http://purl.
org/dc/dcmitype/_qudt:_http://.../Instances.html#">
 6
<span property="dc:subject" content="sculpture"/> 7
<span property="qudt:Mass" content="0.5" datatype=" 8
xsd:float"/>
</div></body> 9
</html> 10

```

In some cases, it is desirable to decouple semantics from data and present it as a separate RDF document (e.g., to load into a triplestore). The Gleaning Resource Descriptions from Dialects of Languages (GRDDL) [13] is a W3C standard designed for this purpose.

Microdata [14] is another approach proposed by W3C to embed semantics into web content. Alike RDFa, Microdata describes triples comprised of the subject, property and object using attributes for specifying types, scopes, ids, properties and references. To enable common semantics on the web, some vocabularies have been defined for Microdata to describe, e.g., people, organizations, products, etc. [15][16].

Microformats [17] are a solution for describing metadata embedded into web pages, that is simpler than RDFa and Microdata. Microformats make use of the *class* and *rel* attributes to express classification and relationships for web resources. These attributes may be built into various (X)HTML elements such as *span*, *div*, *ul*, etc.

C. Semantic descriptions of 3D models

Several works have been conducted to combine X3D content with semantic descriptions. In [18], an integration of X3D and OWL using scene-independent ontologies and the concept of semantic zones have been proposed to enable querying 3D scenes at different levels of semantic details and implement a guided tour through the Venetian Palace. In [19], interfaces for annotating 3D worlds in X3D and a search module have been described. A few projects have been conducted on extending MPEG-7 with semantic annotations of 3D objects. In [20], some descriptors have been introduced to optimize specification of semantics of X3D objects, in particular their sizes, types, curvatures, etc. The works [21][22] consider a generic semantic annotation model for describing semantics of X3D content using MPEG-7. In [23], 3D digital assets in COLLADA have been combined with semantic tags to present sculptures and monuments. In [24], ontology-based RDF tags are separated but linked to 3D models via identifiers to enable web presentation of museum artefacts. A generic modelling framework for metadata based on semantic web standards and

selected multimedia formats has been presented in [25]. Some core patterns have been designed for describing provenance, structure and values of objects. A video search engine for vehicles described by a set of attributes has been presented in [26]. Some other solutions are devoted to structured composition of interactive behaviour-rich 3D web applications [27]-[29], describing interactivity of 3D objects [30], their interfaces [31], as well as finding 3D objects by their properties [32].

The aforementioned projects address different aspects of semantics of multimedia content in various application domains. However, they do not focus on standardized built-in attribute-based solutions for lightweight inclusion of metadata in web resources. Embedding metadata directly into 3D content has a few important advantages in comparison to approaches that decouple resources from metadata describing them. First, with embedded metadata, 3D models are unambiguously and inextricably linked with their descriptions. Second, this enables more concise semantic descriptions as well as faster and less complicated authoring and analysis of semantically described 3D content. Furthermore, it facilitates combining the semantic descriptions of 3D content with descriptions of web pages that embed the content. Finally, it permits storing the 3D content in structurally simpler databases.

IV. MAPPING OF RDFa ATTRIBUTES TO X3D METADATA NODES

To enable lightweight attribute-based built-in semantic descriptions of 3D web content, a mapping of RDFa attributes to X3D metadata nodes is proposed in this paper. The presented approach utilizes the standard syntax and structure of X3D documents and decouples descriptions of the semantics of 3D components from descriptions of their geometry, appearance and behaviour by putting the metadata into additional elements embedded in X3D nodes (as opposed to RDFa attributes nested in HTML tags).

The mapping between the RDFa attributes and the X3D metadata nodes is depicted in Fig. 2. Example X3D content described with RDFa attributes is presented in Listing 3. The scene (lines 11-65) presents a room in a museum with a shelf (12-34) on which there is a sculpture (35-51) and a complex model of a plough (52-64). Some geometrical and behavioural elements and attributes have been omitted as they are not crucial in the context of the presented method.

```

Listing 3. An example X3D document with RDFa attributes
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD_X3D_3.0//EN" "http://
www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile='Immersive' version='3.0' xmlns:xsd='http://
www.w3.org/2001/XMLSchema-instance'
xsd:noNamespaceSchemaLocation='http://www.web3d.org/
specifications/x3d-3.0.xsd' xmlns:dc='http://purl.org/
dc/terms/'>
<head>
<meta name='dc:identifier' content='http://.../m.x3d' />
<meta name='dc:title' content='Museum room' />
<meta name='dc:created' content='2012-10-05' />
<meta name='semanticDescription' content='http://www.w3.
org/TR/rdfa-syntax' />
</head>
<Scene DEF='MuseumRoom'>
<Shape DEF='VirtualShelf'>
<Appearance>...</Appearance><Box />

```

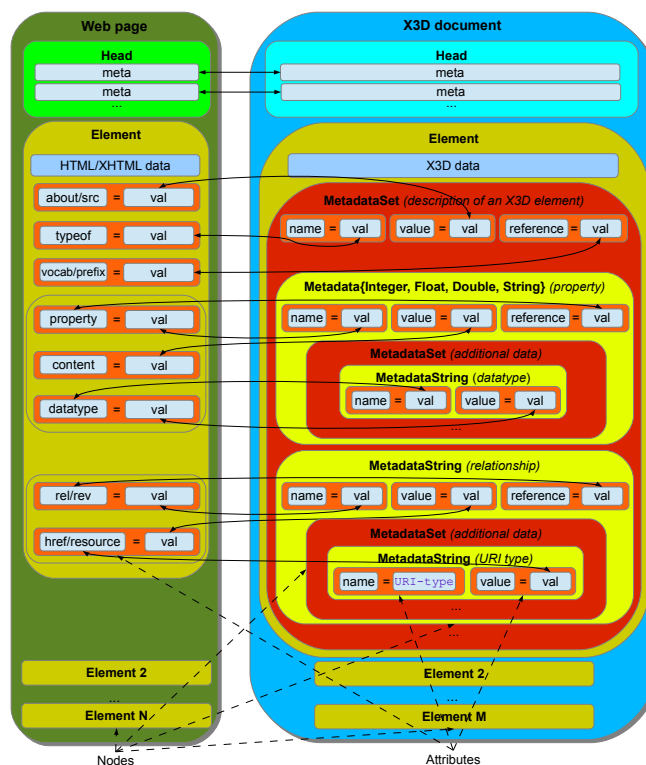


Fig. 2. Mapping between RDFa attributes and X3D metadata nodes

```

<MetadataSet>
<MetadataSet name='http://www.w3.org/ns/ma-on.t.rdf'
value='VirtualShelf'
<MetadataString name='description' value='A model ...'
reference='property' />
<MetadataString name='hasFormat' value='X3D' reference
='property' />
<MetadataString name='creationDate' value='2012-10-02'
reference='property' />
<MetadataString name='datatype' value='xsd:date' />
</MetadataString>
<MetadataString name='hasContributor' reference='
property' />
<MetadataSet name='http://www.w3.org/.../ns.rdf'>
<MetadataString name='fn' value='DIT' reference='
property' />
<MetadataString name='tel' value='+48-618-480-549'
reference='property' />
</MetadataSet>
</MetadataString>
</MetadataSet>
<MetadataSet name='http://example.org/museum/types/
shelf' value='http://example.org/museum/shelf'>
<MetadataString name='dc:medium' value='wood metal'
reference='property' />
<MetadataString name='dc:creator' value='http://.../
carpenter' reference='rel' />
<MetadataString name='URI_type' value='href' />
</MetadataString>
</MetadataSet>
</MetadataSet></Shape>
<Transform DEF='VirtualSculpture'>
<Appearance>...</Appearance>
<IndexedFaceSet>...</IndexedFaceSet>
<MetadataSet>
<MetadataSet value='http://.../museum/sculpture'
reference='foaf:http://xmlns.com/foaf/spec/#term_
qudt:http://www.qudt.org/.../Instances.html#>
<MetadataFloat name='qudt:Mass' value='1.5' reference
='property' />
<MetadataString name='foaf:depiction' value='
VirtualSculpture' reference='rel' />
<MetadataString name='URI_type' value='resource' />
</MetadataString>
</MetadataSet>

```

```

<MetadataSet value='VirtualSculpture' reference='http
  ://xmlns.com/foaf/spec/#term_'> 46
  <MetadataString name='depicts' value='http://.../ 47
    museum/sculpture' reference='rel'>
    <MetadataString name='URI_type' value='href' /> 48
  </MetadataString> 49
</MetadataSet> 50
</MetadataSet></Transform> 51
<Transform DEF='VirtualPlough'> 52
  <Transform DEF='VirtualBox' /><Transform DEF=' 53
    VirtualHook' />
  <Transform DEF='VirtualCylinder'> 54
    <MetadataSet value='VirtualCylinder'> 55
      <MetadataString name='dc:hasPart' value=' 56
        VirtualPlough' reference='rev'>
        <MetadataString name='URI_type' value='resource' /> 57
      </MetadataString> 58
    </MetadataSet></Transform> 59
  <MetadataSet value='VirtualPlough'> 60
    <MetadataString name='dc:hasPart' value=' 61
      VirtualCylinder' reference='rel'>
      <MetadataString name='URI_type' value='resource' /> 62
    </MetadataString> 63
  </MetadataSet></Transform> 64
</Scene> 65
</X3D> 66

```

For the first type of metadata, the mapping is easy, as the resulting meta elements are equivalent to meta elements in a web page (5-10). The standard to which a particular semantic description conforms may be indicated by the appropriate meta node (9) to enable proper interpretation of the document.

The second type of metadata in X3D documents, in the presented method, is used for describing the semantics of both real objects and their corresponding 3D models. The mapping is performed between RDFa attributes that are intended for (X)HTML web pages and metadata nodes of X3D documents. As a web page may include many (X)HTML elements with RDFa attributes, an X3D document may incorporate a number of XML nodes corresponding to different components of a 3D scene, which may be described by metadata nodes.

The primary unit of the semantic description of such resources is the `MetadataSet` node. The presented method encompasses all the RDFa attributes mentioned in Section III-B, putting their names and values into attributes of X3D metadata nodes. A semantically described X3D element contains a `MetadataSet` node (in addition to nodes expressing its geometry, appearance and behaviour) that may include multiple metadata sub-nodes to which particular RDFa attributes are mapped as follows:

- 1) `typeof` and `about/src` are mapped to the `name` and `value` attributes of the `MetadataSet`, respectively. Both attributes are optional and they are used in the same way for real objects such as a worker, a sculpture, a museum (28), and virtual objects, e.g., a document, an image, a 3D model (15). The type should be defined by a data structure with properties. Data types and attributes used in the presented example belong to FOAF, Media Resources, Dublin Core and QUDT domains that are intended for RDF. New item properties may be added to the `MetadataSet` independently of the specified data type. The optional `value` attribute specifies the URI of the described web resource (not necessary navigable). Since

the method enables description of the semantics of real objects and their 3D models, both types of resources are referenced in the same manner. If a particular 3D component is to be linkable, it has to be assigned a URI in the X3D `DEF` attribute (11, 35). Such a solution has three important implications. First, it conforms to the language specification. Second, it permits referencing both local 3D resources—through their URIs—and remote nested 3D components—preceding their URIs by HTTP URIs of their parent web resources. Third, any real or virtual object may be described by any `MetadataSet` independently of their relative location on the web. In particular, a `MetadataSet` no longer needs to describe only the parent 3D component.

If the `about/src` attribute is not mapped for a particular `MetadataSet` node (no URI of the described resource is specified), the `MetadataSet` represent a blank node (22-25). Blank nodes are used in semantic descriptions when a subject is in a relationship with a complex resource described by multiple fields, that has no URI specified. In the presented example, a blank node is used to describe the creator of the 3D model of the shelf.

- 2) `vocab` (46), `prefix` (40)—are optionally mapped to the `reference` attribute of the `MetadataSet`. Both the attributes are applied to the entire scope of the node, including all its descendants. The `prefix` attribute may include a list. In the presented example, these attributes simplify using FOAF (47) and quantity (41) definitions.
- 3) `property/rel/rev` and `content/href/resource` attributes are mapped to the `name` and `value` attributes of a typed node. For properties, nodes of all types may be utilized (integer, float, double, string), while relationships (`rel` and `rev` attributes) are only reflected by `MetadataString` nodes. To differentiate navigable from non-navigable URIs of objects given in relationships, an additional `MetadataString` sub-node is introduced with the `name` set to `URI-type` and the `value` set to `href` or to `resource`. In the presented example, relationships describe links to external content (30, 42, 47) and hierarchical dependencies between 3D components (56, 61), while properties describe various features of resources such as descriptions, formats, medium, etc. (16-18, 29). When no `value` attribute is given for the property, the descendant blank node is used as its value, e.g., contributor described by multiple properties (22-25). To describe some additional aspects of a property, sub-nodes may be introduced, e.g., for mapping the RDFa `datatype` attribute when typed metadata nodes are not sufficient to express sophisticated types (19).
- 4) `inList`—since X3D metadata nodes may have several values separated with spaces, this attribute is simply mapped to such an array. In the example, the shelf is made of wood and metal (30).

The meaning of particular X3D attributes in the proposed method differs minimally from their semantics described in the X3D specification. However, X3D metadata nodes have been designed for simple metadata description and not for use with Semantic Web standards. The small extension of the meaning of their attributes enables full adoption of the presented technique without disturbing the current syntax and structure of X3D and conflicting with widely-used 3D browsers.

An approach alternative to the presented one could be based on the ordinary extension of X3D syntax with RDFa attributes embedded into particular X3D nodes (alike in HTML). Although little more concise, such solution is outperformed by the proposed method in terms of the compatibility with available X3D browsers and flexibility in describing 3D components distributed across the web. Despite the difference between the syntaxes of these approaches, the resulting semantic descriptions are equivalent in terms of expressiveness.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, a method of creating lightweight attribute-based built-in semantic descriptions of 3D web content has been proposed. Although several solutions have been developed for the Semantic Web, they are intended mostly for describing standard web pages. Lack of commonly accepted approach to describing semantics of 3D resources is one of the important obstacles for widespread creation, dissemination and reuse of 3D content on the web.

The presented method combines RDFa with X3D and it is compliant with well-established web standards. It provides a bidirectional mapping between RDFa attributes that are used in typical web pages and metadata nodes used in X3D documents without introducing any modifications to their standard syntax and structure. Both descriptions are semantically equivalent. Embedding semantics into the described documents permits adoption of the method without the need for implementing additional repositories and tools.

Possible directions of future research incorporate several facets. First, additional mappings should be elaborated for other formats of attribute-based semantic descriptions, in particular for Microdata and Microformats. Second, the proposed approach stresses the compatibility with available 3D browsers but not with RDFa parsers. To harvest metadata from semantically described X3D documents and evaluate the proposed method, a GRDDL agent should be implemented (e.g., based on XSLT transformations). Third, a SPARQL engine could be developed to enable querying X3D models. Finally, the method may be combined with semantics derived from spatial, temporal, structural, logical and behavioural components of 3D models.

REFERENCES

- [1] Resource Description Framework (RDF). <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Retr. September 18, 2012.
- [2] RDFa Core 1.1. W3C Recommendation 07 June 2012. <http://www.w3.org/TR/rdfa-syntax/>. Retr. September 18, 2012.
- [3] The Virtual Reality Modelling Language. <http://www.web3d.org/x3d/specifications/>. Retr. September 18, 2012.
- [4] Extensible 3D (X3D), ISO/IEC 19775-1:2008. <http://www.web3d.org/files/specifications/>. Retr. September 18, 2012.
- [5] Resource Description Framework Schema. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>. Retr. September 18, 2012.
- [6] OWL Web Ontology Language Reference. <http://www.w3.org/TR/owl-ref/>. Retr. September 18, 2012.
- [7] SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>. Retr. September 18, 2012.
- [8] The Friend of a Friend (FOAF) project. <http://www.foaf-project.org/>. Retr. September 18, 2012.
- [9] Ontology for Media Resources 1.0. <http://www.w3.org/TR/mediaont-10/>. Retr. September 18, 2012.
- [10] The Dublin Core Metadata Initiative. <http://dublincore.org/>. Retr. September 18, 2012.
- [11] QUDT—Quantities, Units, Dimensions and Data Types in OWL and XML. <http://www.qudt.org/>. Retr. September 18, 2012.
- [12] Chemical Entities of Biological Interest. <http://www.ebi.ac.uk/chebi/>. Retr. September 18, 2012.
- [13] Gleaning Resource Descriptions from Dialects of Languages (GRDDL). <http://www.w3.org/TR/grddl/>. Retr. September 18, 2012.
- [14] HTML Microdata. <http://www.w3.org/TR/2011/WD-microdata-20110525/>. Retr. September 18, 2012.
- [15] Schema.org. <http://schema.org/>. Retr. September 18, 2012.
- [16] Data-Vocabulary. <http://www.data-vocabulary.org/>. Retr. Sept 18, 2012.
- [17] Microformats. <http://microformats.org/>. Retr. September 18, 2012.
- [18] Pittarello F., Faveri A., Semantic description of 3D environments: a proposal based on web standards, In: *Proc. of the 11th Int. Conf. on 3D web Techn.*, Columbia, USA, April 18-21, 2006, pp. 85-95.
- [19] Pittarello F., Gatto I., ToBoA-3D: an arch. for managing top-down and bottom-up annot. 3D obj. and spaces on the web, In: *Proc. of the 16th Int. Conf. on 3D Web Techn.*, Paris, France, June 20-22, 2011, pp. 57-65.
- [20] Spala P., Malamos A., Doulamis A., Mamakis G., Extending MPEG-7 for efficient annotation of complex web 3D scenes, In: *Multimedia Tools and Applications*, vol. 59, Num. 2, 2012, pp. 463-504.
- [21] Bilasco I., Gensel J., Villanova-Oliver M., Martin H., On Indexing of 3D Scenes Using MPEG-7, In: *Proc. of the 13th Annual ACM Int. Conf. on Mult.*, Hilton, Singapore, November 6-11, 2005, pp. 471-474.
- [22] Bilasco I., Gensel J., Villanova-Oliver M., Martin H., An MPEG-7 framew. enhancing the reuse of 3D models, In: *Proc. of the 11th Int. Conf. on 3D web Techn.*, Columbia, USA, April 18-21, 2006, pp. 65-74.
- [23] Rodriguez-Echavarria K., Morris D., Arnold D., Web based presentation of semantically tagged 3D content for public sculptures and monuments in the UK, In: *Proc. of the 14th Int. Conf. on 3D Web Technology*, Darmstadt, Germany, June 16-17, 2009, pp. 119-126.
- [24] Yu C., Groza T., Hunter J., High Speed Capture, Retrieval and Rendering of Segment-Based Annotations on 3D Museum Objects, In: *Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation*, LNCS, 2011, vol. 7008, pp. 5-15.
- [25] Saathoff C., Scherp A., Unlocking the semantics of mult. presentations in the web with the multimedia metadata ontology, In: *Proc. of the 19th Int. Conf. on WWW*, Raleigh, USA, April 26-30, 2010, pp. 831-840.
- [26] Feris R., Siddique B., Zhai Y., Petterson J., Brown L., Pankanti S., Attr.-based vehicle search in crowded surveillance videos, In: *Proc. of the 1st ACM Int. Conf. on Mult. Retr.*, Trento, Italy, April 17-20, 2011.
- [27] Walczak K.: Flex-VR: Configurable 3D Web Applications, In: *Proc. of the IEEE Int. Conf. on HSI 2008*, Cracow, 2008.
- [28] Walczak K.: Configurable Virtual Reality Applications, In: *Wydawnictwa Uniwersytetu Ekonomicznego w Poznaniu*, 2009.
- [29] Walczak, K., Building Configurable 3D Web Applications with Flex-VR, In: *Interactive 3D Multimedia Content—Models for Creation, Management, Search and Presentation*, ed. Cellary, W., and K. Walczak, Springer, London, 2012, pp. 103-135, ISBN 978-1-4471-2496-2.
- [30] Chmielewski J., Describing Interactivity of 3D Content. In: *Interactive 3D Multimedia Content—Models for Creation, Management, Search and Presentation*, ed. Cellary, W., Walczak K., Springer, London, Dordrecht, Heidelberg, New York, 2012, pp. 195-221, ISBN 978-1-4471-2496-2.
- [31] Chmielewski J., Metadata Model for Interaction of 3D Object, In: *The 1st International IEEE Conference on Information Technology*, ed. Stepnowski, A., M. Moszyński, T. Kochaski, J. Dbrowski, Gdańsk, May 18, 2008, Gdańsk University of Technology, 2008, pp. 313-316.
- [32] Chmielewski J., Finding inter. 3D obj. by their inter. prop., In: *Mult. Tools and Applications*, Springer, Netherlands, 2012, ISSN: 1380-7501.

The Rise of the Web for Agents

Ruben Verborgh, Erik Mannens, Rik Van de Walle
 iMinds – Multimedia Lab – Ghent University
 Gaston Crommenlaan 8 bus 201
 B-9050 Ledeborg-Ghent, Belgium
 {ruben.verborgh, erik.mannens, rik.vandewalle}@ugent.be

Abstract—Autonomous intelligent agents are advanced pieces of software that can consume Web data and services without being preprogrammed for a specific domain. In this paper, we look at the current state of the Web for agents and illustrate how the current diversity in formats and differences between static data and dynamic services limit the possibilities of such agents. We then explain how solutions that strive to provide a united interface to static and dynamic resources provide an answer to this problem. The relevance of current developments in research on semantic descriptions is highlighted. At every point in the discussion, we connect the technology to its impact on communication. Finally, we argue that a strong cooperation between resource providers and developers will be necessary to make the Web for agents emerge.

Keywords—Software agents; Semantic Web; Web services

I. INTRODUCTION: THE WEB FOR AGENTS

A. Imagining intelligent agents

Artificial intelligence has always been a dream of mankind [1], and the Web is bringing new opportunities to create automated, intelligent behavior. After all, the World Wide Web is arguably the largest collective work of knowledge ever produced. A significant amount thereof is publicly available, giving any human access to massive amounts of information. Any *human* indeed, but what about machines?

While today, the Web houses many search engines [2], their functionality largely comes down to *keyword search*: to answer a certain question, a search engine can help us find documents with related keywords, but we have to read and interpret those documents ourselves. Although very convenient, it does not mark the endpoint of our desires: it would be so much easier if a machine could directly answer our question.

It is not hard to imagine an *intelligent agent* [3] that looks up facts online—and we could also contemplate on all sorts of other tasks such an agent might take over from us: order groceries, plan a holiday, submit tax returns, *etc.* However easy it is to imagine, it turns out very hard to implement such a universal agent. With the release of Apple's Siri on the iPhone, we have seen a glimpse of the potential of agent technology, but even Siri is still programmed specifically for certain domains [4]. A truly universal agent must be able to use the Web to do things it has not been preprogrammed for.

In this paper, we investigate exactly how far away we are from such a universal intelligent agent. We take a look at current possibilities for machine agents, and identify missing links that need to be resolved before we all can witness the rise of the Web for agents.

B. The Web's role in communication history

To understand the significance and impact of the Web for agents, we have to view it in a richer historical perspective. Technology has always played an important role throughout the evolution of human communication, to the extent that several technological advances have contributed to the development of new models. Figure 1 illustrates the evolution through the four communication models detailed below.

- **one to one**—The invention of *writing* [5] made it possible to communicate complex messages from one person to one other person at the same time, without the requirement for these individuals to meet. Traditional writing only recently lost its dominant position in interpersonal communication to electronic media.
- **one to many**—Conveying the same message to larger audiences involved manual copying until the invention of the *printing press* [6]. Even in today's technological society, printed works remain an important means of spreading knowledge.
- **many to many**—The *World Wide Web* [7] made a radical change in the communication model by enabling bidirectional interactions. The scalable nature of the Web makes it indeed possible for more people than ever before to engage in worldwide communication.
- **between humans and machines**—The *Semantic Web* [8] aims to introduce another group of actors in the model: machines, which can range from software programs to any kind of electronic device. However, the degree of autonomy of such automated agents is currently limited.

Therefore, in the next section, we address the agent concept as a communication question, and elaborate on the current barriers between humans and machines on the Web. In Section III, we look at necessary changes in the future. Finally, we conclude in Section IV with a summary of what agents need.

II. MACHINE-ACCESSIBLE RESOURCES TODAY

On today's Web, many resources are already machines-accessible in different ways and with varying degrees of automated interpretability. In most cases, clients need to be programmed for a specific purpose and tailored to a specific resource implementation. We will mention machine-accessible data embedded in HTML documents (both standardized and non-standardized), machine-accessible data in separate documents, and on-demand resources generated by Web services, and discuss their advantages and disadvantages.

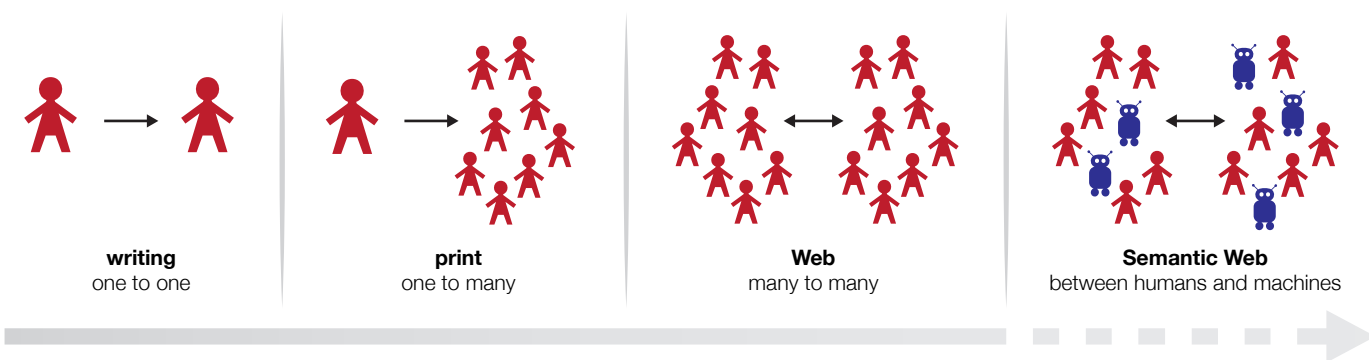


Fig. 1. Technology made communication evolve from a one-to-one to a many-to-many model. The future could bring human-machine intercommunication.

A. HTML with annotations

The Semantic Web’s most common data format, RDF (Resource Description Framework, [9]), has a well-known counterpart that can be embedded in HTML, called RDFa [10] (derived “from RDF in attributes”). The recent 1.1 update supersedes the 1.0 version, which exclusively focused on XHTML. RDFa has become somewhat of a success story of the Semantic Web that also gained visibility outside of the scientific community. A prominent example of this is GoodRelations [11], an ontology to describe products, which was adopted by Google to enhance product search results [12]. Google thereby provided a clear incentive for product providers to enhance their HTML representations with machine-accessible information.

A second major incentive to use RDFa came from Facebook, who based the initial version of the Open Graph protocol on RDFa [13]. This protocol enables HTML pages to become objects people can talk about and share on Facebook and other social networks [14]. Given the importance of social media marketing nowadays, many website owners chose to provide RDFa metadata in the Open Graph vocabulary. This latter property is also the downside of Open Graph: the documentation seems to imply that the vocabulary choice is fixed. Instead of reusing existing ontologies [15], a new one was created, without providing links to define meaning in terms of other ontologies.

Another example of non-reuse was the introduction of Schema.org [16] by Google, Bing, and Yahoo!, widely considered an answer to Facebook’s Open Graph protocol. Schema.org leaves the RDFa path by annotating human-readable text with HTML5 microdata [17], again with a specifically designed vocabulary. The Semantic Web community reacted quickly and released an RDFS schema for Schema.org [18], which eventually resulted in an official OWL version of the schema. The Schema.org annotation method has also been made compatible with RDFa Lite [19].

By now, the major issue with having similar but different vocabularies should be apparent: in how many vocabularies do we need to annotate a single HTML page to be understandable by all consumers? If any major RDFa consumer can impose a vocabulary on webmasters, annotating Web pages becomes

a never-ending task. Only recently, one year after the launch of Schema.org, Twitter announced another annotation mechanism called Twitter Cards [20], which has a considerable overlap with the Open Graph vocabulary—and, consequently, with Schema.org. This implies that the *same* semantic content has to be expressed differently *three* times to be interpretable by the major Web traffic sources Google, Facebook, and Twitter. This is an alarming observation, since RDF ontologies were precisely created to enable interchangeable data, which is, after all, supposed to be the added value of semantic technologies.

From a communicational standpoint of view, we could say that, while common (and even standard) languages are used, every agent refuses to communicate in a lingo different from its own. It remains an open question whether any of the consumers of annotated HTML will support formats or vocabularies endorsed by others. As its Structured Data Testing Tool [21] shows, Google is able to extract RDFa and other data marked up with different vocabularies (including Open Graph metadata), but it is unsure whether this data will be used and if so, whether it will carry the same importance as Schema.org metadata. For the moment, only limited search result enhancements are performed, even if the annotations are written in the Schema.org vocabulary. However, the provided metadata fields are sufficiently powerful to enable a broad automated understanding of basic content properties, so increased usage can be expected in the future. To verify this, the aforementioned tool can generate a preview of how additional semantic annotations currently affect search results.

The diversity also forms a burden for implementors of intelligent agents, who cannot rely on one standardized vocabulary. For example, HTML itself has a single way of specifying a page title, namely the title tag, whereas each of the three aforementioned annotation mechanisms have *different* means of expressing the same thing. Fortunately, independent implementors are not bound by the business-driven decisions that prompted Facebook, Google, and Twitter to each favor or even exclusively support a different technique. If we assume the availability of an ontology that brings together similar terms from the different vocabularies, Semantic Web reasoning techniques can translate content from one vocabulary to another [22]. Therefore, support for one format could be sufficient for agents to support all of them.

B. Machine-targeted document formats

In addition to embedding machine-processable data in human-targeted representation, it can also be expressed directly in a machine-targeted format. One such format is of course RDF. We can distinguish two cases: either the RDF version is a machine-friendly alternative representation of a human-targeted document, or either the RDF version is the unique representation of the data. The first case is not unlike RDFa, where an HTML document always accompanies the embedded RDF data. However, isolating machine data from human data has the benefit of separating concerns, since only one of the data streams is needed at a time. Thanks to the Linked Data movement and principles, many datasets have already been made available as RDF [23].

At the moment, RDF is however not the default choice for many Web developers to expose data in a machine-processable way. While a few years ago XML was a common structured format, the JavaScript Object Notation (JSON, [24]) has become increasingly popular and is now ubiquitous, mainly thanks to its simplicity and native compatibility with the dominant Web client language JavaScript. JSON allows to represent complex hierarchical data efficiently, but unfortunately does not have any inherent semantics associated with data fields. As a result, clients have to be programmed to understand a specific type of JSON information. Furthermore, since JSON lacks native identification support (such as URIs in RDF), it is difficult to identify individual resources or to make circular references.

The JavaScript Object Notation for Linking Data (JSON-LD, [25]) provides a solution to these problems by bridging between JSON and RDF. It adds an `id` property to JSON fragments to enable identification of resources, and a `context` property to identify the semantics of data fields. Communication-wise, JSON-LD and RDF have equal expressive power, but the latter has the benefit of native JavaScript support, providing maximal parsing speed and familiarity for developers without prior Semantic Web knowledge. In that sense, JSON-LD is a hybrid language: its semantic grounds in RDF make it interpretable by automated clients, whereas its JSON format offers an accessible interface for developers of such clients.

C. Web services

As stated in the introduction, the envisioned tasks of agents are not limited to information retrieval. Far more possibilities—and a greater complexity—reside in performing so-called *world-changing* actions, which alter the state of digital or real-world objects. Many Web services [26] offer such actions, for example posting comments, ordering books, or reserving tickets. Preprogrammed clients do not need to know what task a service performs, because the author of the client is the one who interprets the service's functionality. The situation is different for an autonomous agent: if it wants to complete a certain task, it has to find a service that offers the desired functionality and it then needs to figure out how to invoke that service. Similarly to the role of machine-interpretable metadata in HTML documents, clients need a service's metadata to understand its capabilities and modalities.

Currently, most Web service documentation is only written in human language for developers. Many frameworks for discovery and machine-processable description of invocation modalities have been created, notably UDDI [27] and WSDL [28]. However, these frameworks do not reach beyond the technical aspects of a service, and therefore serve as assisting technologies for application developers instead of as metadata for autonomous agents.

The crucial difference with static data is the world-changing aspect: when *retrieving* information resources, no harm can be done because the application state doesn't change. Since a service can have *side-effects*, it can potentially be dangerous to issue requests without understanding what is going to happen. While the irrelevance of data can be determined afterwards, upon which that data can safely be discarded, the determination of a service's results after its invocation comes too late if that action cannot be rolled back. The write aspect is indeed as important as the read aspect, but currently underdeveloped [29]. Therefore, autonomous intelligent agents on today's Web are scarce, and those that exist are limited to a specific, pre-programmed domain, such as is the case with Apple's Siri.

III. MACHINE-ACCESSIBLE RESOURCES FOR THE FUTURE

We will now have a look at techniques to make resources machine-accessible that are currently under development or research, of which we believe they will play a major role on the future Web for agents. Some of these techniques are already in use today, while others are still in the research stage or awaiting adoption.

A. The uniform interface

A key part in our vision is the unification of static resources and services, blurring the distinction between them, and providing a uniform interface to access all resources. This makes agent development considerably simpler since it needs to be programmed only against a single interface, instead of requiring different bindings for every service. Such a uniform interface is featured in the work of Roy T. Fielding, whose doctoral dissertation details the architectural style that underpins the original design of the Web's HyperText Transfer Protocol (HTTP, [30]), and is called REpresentational State Transfer (REST, [31]). The REST architectural style defines several *constraints* on the communication between clients and servers. These constraints introduce certain desirable properties in the systems that obey them, including reliability and scalability. Although HTTP provides the necessary interfaces to build applications that function according to the REST architectural constraints, not many of today's Web applications actually implement all of them.

According to Fielding, four constraints contribute to the uniform interface: *identification or resources*, *manipulation of resources through representations*, *self-descriptive messages*, and *hypermedia as the engine of application state*. We will now go through each of these constraints, and indicate why they are important for autonomous agents.

1) *Identification of resources*: On the Web, resource identification is achieved through the use of Uniform Resource Locators (URLs, [32]). A resource is defined as a temporarily varying membership function, of which the *mapping definition* (the function itself) remains constant, but the *mapped entity* can vary in time. For example, the resource identified by the URL <http://magazine.example.org/issues/current> could be defined as “the latest issue of Example magazine”, but could, depending on the month, be the January or February issue. This resource would be separate from “the January issue of Example magazine”—even if the current month is January and the mapped entities are thus the same—because the mapping definition is different.

The benefit for agents is clearly simplification, because everything is a resource and is identifiable by a URL. This means there is no need (or space) for a “service”, since everything is modelled in terms of resources. For example, instead of a service that returns tomorrow’s weather forecast for a specified city, the server provides a resource that gives the same thing and has a distinct URL. That way, this resource is indistinguishable from what we have previously called “static data”—and there is no apparent reason why it should not be. The reason we often *do* see a distinction is because the details of the underlying server implementation are inadvertently surfacing (e.g., the forecasts are retrieved by a specific script, which is wrapped as a service rather than modelled as a set of resources with identifiers).

2) *Manipulation of resources through representations*: In HTTP communications, resources themselves are not exchanged or manipulated, but rather representations thereof. A single resource might have multiple representations in different formats, which allows clients and servers to perform *content negotiation* to mutually decide on a representation they both understand. For example, a browser would indicate (through Accept headers) a preference for HTML in English or Spanish (which can be set by the user). Another client might have a preference for plaintext or JSON. The server will try to accommodate those preferences to the extent possible.

From a communication viewpoint, this is very beneficial for agents. They can access, work with, and communicate about the *same* resources as those on the human Web. In fact, the Web for humans and the Web for agents are the *same* Web: only the representations are different, since machines are yet incapable of understanding human language. For the time being, agents can employ content negotiation to ask for RDF or JSON-LD content, and perhaps even indicate their preference for a specific vocabulary. Should the preferred version not be available, the server can choose to serve a best-effort representation such as HTML with RDFa.

3) *Self-descriptive messages*: One of the aspects of self-descriptiveness of messages is *statelessness*: every message should contain all metadata necessary to understand its meaning. For example, to get from the first to the second page of a listing, a client does not send a “next” message, but rather a request for the second page resource. This ensures that no other message is required to understand the request.

Another aspect of self-descriptiveness is the use of *standard methods*. HTTP provides only a few generic methods, such as GET, PUT, POST, and DELETE. This small number of methods shifts the focus from sending messages to objects (as is the case in object-oriented programming) to retrieving and manipulating representations of resources. For example, instead of sending the `findWeather` message with “Boston” as an argument, we perform a standard GET request on the “current weather in Boston” resource. This again considerably simplifies agent development, plus it offers *guarantees* for several methods: GET guarantees that it will preserve resource state, while DELETE does not. Both methods guarantee that they can be executed multiple times without causing additional effects from the first execution, which POST cannot.

Some Web applications today violate statelessness and/or do not respect the guarantees of the standard methods. This can be problematic for agents, as they need to correctly assess the consequences of requests they make. Even on the human Web, problems can arise if a method that should not induce side-effects suddenly does [33].

4) *Hypermedia as the engine of application state*: The last constraint necessary to achieve a uniform interface is known as the hypermedia or HATEOAS (Hypermedia As The Engine Of Application State) constraint. Due to lack of time, Fielding’s dissertation does not elaborate on this constraint, but he did so in later blog posts [34]. Basically, the hypermedia constraint says that a representation of a resource should contain the necessary *controls* to chose possible next steps or actions. For example, in the hypermedia format HTML such controls are links, forms, buttons, *etc.*

If we look at the human Web, we see that the hypermedia constraint is well-implemented: people never have to manually type a URL in the address bar to change a page within the same website. The situation is different for machine-targeted resources: often, the developer has to configure an agent to use or construct specific URLs. This limits the capabilities of agents, since they need hypermedia controls as much as humans do. With RDF content, such controls are implicitly defined if URLs (as a special case of URIs) are used as resource identifiers, since these identifiers then serve as links that can be followed to look up more information about the resource (known as *dereferencing*). However, this only concerns static resources, as the RDF standard currently does not define semantics other than retrieval.

If we look at the act of communication, the hypermedia constraint makes messages provide the *context* necessary to gain more insight in the meaning of the resource, both for humans and agents. Note that, although the *message* should be self-descriptive, the *representation* carried within that message can have controls to other resources—this is exactly the purpose of hyperlinks. Although it is possible to add controls to machine-targeted representations at a later stage [35], this often involves remodeling the Web application in a resource-oriented way. The hypermedia constraint is, however, a requirement for agents that want to autonomously discover and consume resources.

B. Semantic description of functionality

While the uniform interface thus creates the necessary environment for agents, the RDF content type does not provide sufficient semantics to perform all possible tasks. This is because RDF is intended for static documents in the first place. Concretely, if an RDF document contains a URL of a certain resource, then an agent can predict what will happen if a GET request is issued: the server will return a representation of the identified resource. The HTTP specification [30] also describes the effect of other methods: PUT will place the entity supplied with the request at the specified URL, and DELETE will remove the identified resource (given the agent has sufficient permissions to issue such requests).

However, as the HTTP specification states, “[t]he actual function performed by the POST method is determined by the server and is usually dependent on the Request-URI.” This indicates that HTTP provides no means for an agent to predict the effect of issuing a POST request to a resource. This is a major issue, since the POST method is needed often: the specification mentions examples such as posting messages, handling form data, and annotating resources. The issue does not occur on the human Web, because the POST form is usually accompanied by textual and visual clues, and because we understand the context in which the form is presented to us. Agents do not have similar clues at their disposal, since RDF only describes resources statically. Because of this, agents require a description of the dynamic aspects of Web resources in order to understand the effect of methods such as POST.

Earlier description techniques, such as OWL-S [36] and WSMO [37], were designed with the Web service model in mind. Web services employ a so-called *overloaded* form of POST, the semantics of which do not correspond to those in the HTTP specification. Therefore, these techniques are not the right match for Web applications with a REST architecture.

Several description techniques that specifically target REST are currently the subject of research. One approach to combine the REST principles and RDF is to start from the SPARQL query language [38], since it supports update operations from version 1.1 onwards [39]. This can indeed be a way for RDF-aware agents to perform state-changing operations on resources. However, SPARQL is a technology specifically for machines and is therefore not suited for environments where other representations are also important, as in the Web for agents that we envision. Linked Open Services (LOS, [40]) expose functionality on the Web using a combination of HTTP, RDF, and SPARQL in a way that is more friendly towards different representation formats, although the hypermedia constraint is currently not addressed by this technique.

RESTdesc [41] is a semantic format specifically designed to describe the effects of POST requests in hypermedia-driven Web applications. It aims to complement static RDF descriptions of resources with a dynamic view in an RDF-like language. Its purpose is to support autonomous agents in the execution of non-preprogrammed actions on dynamic resources on the Web.

From a communication perspective, finding a way of letting machines consume dynamic resources with the same ease as static resources comes down to semantically expressing what the effects of manipulating a specific resource are. Eventually, this should enable agents to choose specific resource actions based on the functional goals they want to achieve. That way, a user could instruct an agent to perform a task, which the agent then can break down in different resource manipulations. Furthermore, such a resource-oriented approach integrates well with existing machine-readable data on the Web, whereof Linked Data is the most prominent example. Agents can then use this data seamlessly to achieve their goals [42].

IV. CONCLUSION: WHAT THE WEB FOR AGENTS NEEDS

In this paper, we have zoomed in on the obstacles for agents on today’s Web. Many competing machine-processable annotation techniques exist, as well as a large variety of RDF documents. World-changing actions are performed through Web services, which are separate from other documents on the Web. In an ideal Web for agents, there is a uniform interface to all resources, both static and dynamic, removing the current distinction between documents and services. At the same time, agents will need a mechanism to understand the effects of performing state-changing operations on resources.

As long as machines are not able to understand human language, semantic technologies will remain important for agents to derive meaning from resources on the Web. We believe, however, that this is best achieved in a transparent way such as with the resource and representations model, which exposes the *same* resources for both human and machine Web clients. Or in the words of the famous Scientific American article: “*The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.*” [8]

If we want to take the step to the next communication model, in which machine agents have the same capabilities on the Web as humans, we must be willing to work on the current issues and start building Web applications with all aspects of the uniform interface. We have only seen the tip of the iceberg of what is possible with agent technology, and opportunities will only increase as the number of devices that join the Web grows on a daily basis. However, it will take a strong cooperation between Web resource providers and agent developers to make the rise of the Web for agents happen in the not-too-distant future.

ACKNOWLEDGMENTS

The described research activities were funded by Ghent University, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union.

REFERENCES

- [1] B. G. Buchanan, "A (very) brief history of artificial intelligence," *AI Magazine*, vol. 26, no. 4, pp. 53–60, 2005.
- [2] A. Halavais, *Search Engine Society*, ser. Digital media and society series. Cambridge: Polity, 2008.
- [3] J. Hendler, "Agents and the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 30–37, Mar–Apr 2001.
- [4] J. Aron, "How innovative is Apple's new voice assistant, Siri?" *The New Scientist*, vol. 212, no. 2836, p. 24, 2011.
- [5] H. Haarmann, *Geschichte der Schrift*, ser. Wissen in der Beck'schen Reihe. C. H. Beck, 2002, vol. 2198.
- [6] T. Carter and L. Goodrich, *The invention of printing in China and its spread westward*. New York: Ronald Press, 1955.
- [7] T. Berners-Lee and M. Fischetti, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco: Harper, 2000.
- [8] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [9] G. Klyne and J. J. Carroll. (2004, Feb.) Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- [10] B. Adida, M. Birbeck, S. McCarron, and I. Herman. (2012, Jun.) RDFa core 1.1. W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/2012/REC-rdfa-core-20120607/>
- [11] M. Hepp, "GoodRelations: An ontology for describing products and services offers on the Web," *Knowledge Engineering: Practice and Patterns*, pp. 329–346, 2008.
- [12] E. Franzon. (2012, Nov.) Google recommends using RDFa and the GoodRelations vocabulary. [Online]. Available: http://semanticweb.com/google-recommends-using-rdfa-and-the-goodrelations-vocabulary_b909
- [13] I. Facebook. (2010) The Open Graph protocol. [Online]. Available: <http://opengraphprotocol.org/>
- [14] M. Zuckerberg. (2010, Apr.) Building the social Web together. [Online]. Available: <https://blog.facebook.com/blog.php?post=383404517130>
- [15] E. Simperl, "Reusing ontologies on the Semantic Web: A feasibility study," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 905–925, 2009.
- [16] Google, Inc., Yahoo, Inc., and Microsoft Corporation. (2011, Jun.) Schema.org. Specification. [Online]. Available: <http://schema.org/docs/schemas.html>
- [17] Web Hypertext Application Technology Working Group. HTML – microdata. [Online]. Available: <http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html>
- [18] M. Hausenblas and R. Cyganiak. (2011, Jun.) What is schema.rdfs.org? [Online]. Available: <http://schema.rdfs.org/>
- [19] M. Sporny. (2012, Jun.) RDFa lite 1.1. W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/2012/REC-rdfa-lite-20120607/>
- [20] A. Roomann-Kurrik. (2012, Jun.) Twitter Cards. [Online]. Available: <https://dev.twitter.com/blog/twitter-cards>
- [21] Google. Structured data testing tool. [Online]. Available: <http://www.google.com/webmasters/tools/richsnippets>
- [22] A. Hogan, J. Pan, A. Polleres, and Y. Ren, "Scalable owl 2 reasoning for Linked Data," in *Reasoning Web. Semantic Technologies for the Web of Data*, ser. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 2011, vol. 6848, pp. 250–325.
- [23] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data – The Story So Far," *International Journal On Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.
- [24] D. Crockford. (2006, Jul.) The application/json media type for JavaScript Object Notation (JSON). IETF Request for Comments. [Online]. Available: <http://www.ietf.org/rfc/rfc4627>
- [25] M. Lanthaler and C. Gütl, "On using JSON-LD to create evolvable RESTful services," in *Proceedings of the Third International Workshop on RESTful Design*, Apr. 2012, pp. 25–32.
- [26] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to Web services architecture," *IBM Systems Journal*, vol. 41, no. 2, pp. 170–177, Apr. 2002.
- [27] T. Bellwood, S. Capell, L. Clement, J. Colgrave, M. J. Dovey, D. Feygin, A. Hatley, R. Kochman, P. Macias, M. Novotny, M. Paolucci, C. von Riegen, T. Rogers, K. Sycara, P. Wenzel, and Z. Wu. (2004, Oct.) UDDI version 3.0.2. OASIS. [Online]. Available: <http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>
- [28] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. (2001, Mar.) Web Services Description Language (WSDL) 1.1. W3C Note. [Online]. Available: <http://www.w3.org/TR/wSDL>
- [29] S. Coppens, R. Verborgh, M. Vander Sande, D. Van Deursen, E. Mannens, and R. Van de Walle, "A truly Read-Write Web for machines as the next-generation Web?" in *Proceedings of the sw2012 workshop*, Nov. 2012. [Online]. Available: http://stko.geog.ucsb.edu/sw2012/sw2012_paper3.pdf
- [30] R. T. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. (1999, Jun.) Hypertext Transfer Protocol – HTTP/1.1. IETF Request for Comments. [Online]. Available: <http://www.ietf.org/rfc/rfc2616>
- [31] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, May 2002.
- [32] T. Berners-Lee, L. Masinter, and M. McCahill. (1994, Dec.) Uniform Resource Locators (URL). IETF Request for Comments. [Online]. Available: <http://www.ietf.org/rfc/rfc1738>
- [33] R. Verborgh. (2012, Jul.) GET doesn't change the world. [Online]. Available: <http://ruben.verborgh.org/blog/2012/07/19/get-doesnt-change-the-world/>
- [34] R. T. Fielding. (2008, Oct.) REST APIs must be hypertext-driven. Untangled – Musings of Roy T. Fielding. [Online]. Available: <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>
- [35] O. Liskin, L. Singer, and K. Schneider, "Teaching old services new tricks: adding HATEOAS support as an afterthought," in *Proceedings of the Second International Workshop on RESTful Design*. ACM, 2011, pp. 3–10.
- [36] D. Martin, M. Burstein, J. Hobbs, and O. Lassila. (2004, Nov.) OWL-S: Semantic Markup for Web Services. W3C Member Submission. [Online]. Available: <http://www.w3.org/Submission/OWL-S/>
- [37] H. Lausen, A. Polleres, and D. Roman. (2005, Jun.) Web Service Modeling Ontology (WSMO). W3C Member Submission. [Online]. Available: <http://www.w3.org/Submission/WSMO/>
- [38] E. Wilde and M. Hausenblas, "RESTful SPARQL? you name it! – aligning SPARQL with REST and resource orientation," in *Proceedings of the 4th Workshop on Emerging Web Services Technology*. ACM, 2009, pp. 39–43.
- [39] E. Prud'hommeaux and A. Seaborne. (2008, Jan.) SPARQL Query Language for RDF. W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/rdf-sparql-query/>
- [40] R. Krummenacher, B. Norton, and A. Marte, "Towards Linked Open Services and Processes," in *Proceedings of the Third future internet conference on Future Internet*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 68–77.
- [41] R. Verborgh, T. Steiner, D. Van Deursen, S. Coppens, J. Gabarró Vallés, and R. Van de Walle, "Functional descriptions as the bridge between hypermedia APIs and the Semantic Web," in *Proceedings of the Third International Workshop on RESTful Design*. ACM, Apr. 2012.
- [42] J. Domingue, C. Pedrinaci, M. Maleshkova, B. Norton, and R. Krummenacher, "Fostering a relationship between Linked Data and the Internet of Services," in *The Future Internet*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6656, pp. 351–364. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20898-0_25

Semantic Annotation of unstructured Wiki Knowledge according to Ontological Models

Roberto Boselli, Mirko Cesarini, Fabio Mercurio, Mario Mezzanzanica
 Department of Statistics and Quantitative Methods / CRISP Research Centre
 University of Milan Bicocca
 Milan, Italy

Email: {roberto.boselli@unimib.it, mirko.cesarini@unimib.it, fabio.mercurio@unimib.it, mario.mezzanzanica@unimib.it}

Abstract— The paper deals with the issue of supporting users to enrich unstructured wiki contents with semantic annotations. The authors present the development of a semantic wiki that provides semantic annotations compliant to ontological models. The semantic wiki developed, called WiWork, is presented with two related ontological models, the WiWork Core ontology and the Labour Market ontology. Finally, a methodology to define ontology concepts and properties is proposed by using Information Retrieval and statistical techniques.

Keywords: *Semantic Web; Semantic wikis; Ontology modeling; Unstructured data; Information Retrieval*

I. INTRODUCTION

Wikis are collaborative web-based environments allowing users to create, share and reuse useful knowledge. Wiki's knowledge is mainly represented by an amount of unstructured textual contents. Unfortunately, the big volume and the lack of structure make often such contents inaccessible or hard to reuse.

A recent trend focuses on enhancing wikis with Semantic Web technologies and formal languages to access and reuse data, included the unstructured ones: namely semantic wiki. Semantic wikis are promising tools providing the users an easy way to manage machine-processable knowledge, allowing the creation of added-value services based on the semantics of Web pages. They support metadata insertions through semantic annotations and link relations between wiki pages. Annotations are required to refer to an ontological model defining concepts and properties that can be associated to pieces of wiki contents.

In the Semantic Web, ontologies are mainly developed and encoded with formal languages like RDF (Resource Description Format) [1] or OWL (Web Ontology Language) [2]. Moreover, ontology modeling and updating tasks are still hard to be accomplished for common users, and semantic wikis need ontologies as conceptual models to structure their contents.

The work described in this paper aims to provide to the users an easy way to annotate and to structure wiki knowledge without requiring the learning of formal ontology languages like RDF or OWL. The research questions investigated in this paper are, firstly, how to guide users in annotating semantic wiki contents according to ontological models, and secondly, how to build domain ontologies

taking unstructured knowledge as source of concepts and relations. The proposed solution for the first question is the development of a semantic wiki, called WiWork, based on Semantic Mediawiki (SMW) [3]. The latter solution is addressed through the synergy between various research fields: ontology modeling, Information Retrieval (IR) and statistical methodologies. This synergy supported authors in defining two domain ontologies, the WiWork Core ontology and the Labour Market ontology.

The paper is organized as follows: in Section 2 the issue of annotating wiki's unstructured knowledge is discussed, and the semantic wiki WiWork is presented with some of its main functionalities; Section 3 describes the methodology used for ontology modeling based on Information Retrieval and statistical techniques and shows the produced ontologies; in Section 4 a brief survey of related works is provided; finally, some concluding remarks and the future works are outlined in Section 5.

II. UNSTRUCTURED WIKI KNOWLEDGE

Wikis are being confirmed as the most widespread collaborative technology supporting knowledge creation and sharing on the Web [4]. Mainly, wikis allow users to collaboratively create and maintain textual unstructured content by adding, changing, and sharing contents by means of a web browser and a simple markup language. A wiki limitation is that the contents are expressed using natural language text and its meaning is not directly accessible to automated semantic processing tasks. Knowledge in a traditional wiki is freely structured through pages, hyperlinks and user-generated tags (e.g., categories to label pages). Currently, the features available for searching and reusing wiki knowledge are based on full text keyword search. Therefore, wiki knowledge is not automatically accessible for functionalities such as querying, reasoning and semantic browsing. Semantic wikis add these functionalities (with respect to ordinary wikis) for managing knowledge in more formal, machine-processable ways [5]. To achieve this goal semantic wikis have to structure knowledge according to some conceptual models (i.e., ontologies) and to annotate texts with metadata (semantic links and properties). Ontologies written in RDF or in OWL are used in semantic wikis as a reference for concepts used in metadata and annotations.

Adding semantic annotations to wiki's textual content is a complex and laborious process, mainly for common users

who have no competencies in formal languages or knowledge representation methodologies. One solution can be to provide tools facilitating the annotation tasks, a step further is to suggest the users the metadata to use during the creation of pages. Since metadata and semantic annotation are machine-computable in a semantic wiki, they can be either represented directly in RDF. RDF is a common standard to enrich content with formal semantic metadata, but is not easily understandable to common users. To avoid this issue it is necessary to transform RDF triples extracted from ontologies in user-friendly semantic wiki annotations. Various types of semantic wiki annotations can be provided, distinguishing between formal and semi-formal annotations. Tagging is a type of semi-formal annotation. Tags usually consist of keywords (i.e., categories and properties) that the users include in the text of wiki's pages. Semantic links are another type of semi-formal annotation, they consist in structured tags.

According to [6], structured tagging is a semi-formal annotation type in the form of *keyword:value* pairs. This annotation type allows for a simple representation of formal RDF triples: the resource to annotate is the subject, the keyword is the predicate and the value is the object of the triple. Such semi-formal annotation provides an intuitive and easy means to the users for transforming knowledge from human-only to machine-processable content and vice versa. In this way, they provide low barriers for user participation on wiki content enrichment with metadata. The methodology presented in this paper aims to help users to define semi-formal annotations derived from formal ones.

In the next sections, the authors present the main requirements to address the development of a semantic wiki, and, specifically, to enrich wiki content with semantic annotations according to well-defined ontologies.

A. *WiWork, a semantic wiki for public services domain*

In this section the authors describe how a semantic wiki for the CRISP (Interuniversity Research Centre on Public Services) [7] was developed. The research centre develops models, methodologies and tools for collecting, analyzing, and supporting data useful to define and improve services and policies for the public sector. Specifically, the centre designs, develops and uses information systems, Statistical Information Systems and portals for analyzing labour demand and supply [8]. The semantic wiki introduced in this paper, WiWork, was designed with the goal to provide documentations and manuals to domain experts and common users interested in CRISP activities and topics (WiWork is actually not accessible on the Web, but only on the CRISP intranet). Indeed, the wiki has actually about 14000 pages containing knowledge about the public services and focusing on the labour market, on the health, and on the education domains. WiWork has been developed to enrich labour market knowledge with semantic annotations to support tasks, such as matching between different texts, querying, and reasoning.

The content in WiWork is mainly unstructured text devoted to describe the taxonomy of occupations of the Italian labour market. The wiki pages on occupations were

generated taking as primary source the Italian classification scheme of occupations (called CP2011) created by Italian National Institute of Statistics (ISTAT) [9]. This scheme is substantially based on the International Standard Classification of Occupation (ISCO08) criteria [10].

The classification scheme arranges all the occupations in four level groups: Major, Sub-major, Minor and Unit. Moreover, each occupation is classified according to the kind of work performed, the skill level, and the specialization required to fulfill tasks and duties of the job. The scheme provides, for each occupation, a textual description explaining: specific tasks, competencies, required field of knowledge, tools and machinery used, materials used, kind of goods and services produced, etc. The corpora of those textual descriptions contain the most of the domain terminology, mostly in an unstructured form. An example of textual description is the following, extracted from ISCO08: "*Information and communications technology service managers plan, direct, and coordinate the acquisition, development, maintenance and use of computer and telecommunication systems.*" Such a description contains some terms that can be processed only if properly structured, thus requiring the development and the implementation of various semantic technologies and the modeling of ontologies to enrich WiWork content with semantics.

B. *Semantic annotations in WiWork*

WiWork is based on Semantic MediaWiki (SMW) [3], an extension of the well-known MediaWiki, the software used for running Wikipedia. SMW allows one to add metadata to wiki pages and to perform queries over the metadata. Such extension makes the users able to enrich the text using semantic annotations in a way that is accessible both to the human readers and to the machines. The content can be created and maintained collaboratively, while the semantic insertion creates a flexible, extensible, and structured knowledge representation that can be automatically processed, enabling features such as semantic search and RDF data export.

In particular, SMW provides the means to develop a flexible, structured data schema consisting of properties and classes. Properties correspond to semantic links or tags inserted into the page text; classes correspond to categories grouping pages. Both classes and properties are the metadata that a user can manipulate in the wiki using the *keyword:value* form.

The following string `[[uses::computer]]` is an annotation example for the occupation description mentioned above, expressed in SMW syntax. This annotation can be transformed into a RDF triple where an instance of the *Occupation* concept is the subject having a semantic property *uses* as predicate, and an instance of *Tool* concept, *computer*, as object.

This kind of annotation in SMW can be used for semantically enriching both the link between two pages (if *computer* would be a page title) and the property value (if *computer* would be managed as a string of a property value). In both cases it is necessary to define the meaning of properties and concepts and make it available for wiki users.

What follows is the RDF code exported by WiWork of the example:

```
<property:uses
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"> computer
</property:uses>
```

Authors used the Pywikipediabot framework [11] for creating an initial draft of the pages containing the raw text descriptions of the occupations taxonomy. A bot is an application for creating, accessing, and modifying wiki pages via scripts. To this scope some templates were created: a first template to structure all the pages containing information about occupations, including the description, the related categories and the occupation position into the classification scheme (e.g., to which Major group the occupation belongs); a second template structures pages according to the economic sectors related to the occupations, and a third template structures all the information about professional skills linked to occupations. The bot manages the three templates to automatically create pages on the occupations.

Moreover, Semantic Forms, an extension of SMW for data editing [12], was used to facilitate the users to annotate wiki texts with metadata, linking the templates with the forms. The forms provide a graphical user interface allowing the users to create and exploit the templates without requiring the usage of wiki markups. Forms can be composed of several types of fields, e.g., title of occupation, name of economic sector etc. Each field can be related to a specific semantic annotation (e.g., *uses* property), by entering a value in a field (e.g., *computer*), the value is assigned to the corresponding property. All the information inserted via forms is automatically annotated using the properties specified during the template design time. Therefore, the combination of forms and templates allows the users to have a set of predefined semi-formal framework for structuring and annotating information.

Furthermore, the authors plan to develop a program that reuse Protégé-OWL API [13] and Pywikipediabot to automatically transform RDF and OWL triples created with the ontology editor Protégé [14] in SMW constructs (semi-formal annotations) to insert in WiWork pages. A list of categories and properties is necessary to enrich the fields of templates and forms for automatically create annotated pages. The list can be provided taking OWL classes and properties modeled in domain ontologies, as is described in the next Section.

III. DOMAIN ONTOLOGY MODELING

In the following, the authors explain how two domain ontologies were built, the WiWork Core ontology and the Labour Market ontology, representing the conceptual models of the WiWork's content. Ontologies define a set of primitives, e.g., classes and properties, modeling a domain of knowledge [15]. Domain ontologies provide a shared and formal description of the concepts and relationships of a domain grounded on the knowledge and terminology in use in the domain [16].

The goal of this ontology modeling task is to enrich the semantic wiki WiWork with conceptual models that can be

used to structure and to manage knowledge in a machine-processable way.

The adopted ontology development methodology was divided in two steps. First step was supported by the Protégé ontology editor to define the preliminary classes and properties in OWL [14], i.e., the CRISP core concepts and the categories used in WiWork pages. Second step enriched and expanded the ontologies with additional concepts and instances, i.e., the domain concepts. Both the ontology modeling tasks were performed with the support of Information Retrieval techniques to extract concepts from the texts. More precisely, the latter task was performed to identify a set of candidate words and relations to model the occupation terminology.

A. *WiWork Core Ontology and Labour Market Ontology*

Ontologies are designed to define concepts and relationships related to the WiWork knowledge and they are still under development. The methodology described in this paper helped designing and structuring two different OWL ontologies. The first is the WiWork Core ontology representing the conceptual model defining the main topics, research fields, activities, communities, users, and technologies managed by the CRISP.

The second is the Labour Market ontology, being the conceptual model of one specific WiWork topic, used to model the wiki pages about labour market knowledge. Specifically, this ontology has the specific aim to provide a set of predefined metadata that should facilitate users in annotating the occupation descriptions within the wiki. Although the ontologies are being developed separately, they can be seen as two integrated parts of a global conceptual model. In this model the WiWork Core concepts are in the upper levels, and the Labour Market concepts are in the lowest levels. The integration between the ontologies is obtained through the identification of relationships connecting similar or equivalent concepts, and through the identification of relationship types connecting the concepts and the instances extracted from the documents, as described in next Section B.

B. *Modeling via IR techniques*

Since textual documents have an important role in sharing knowledge and concepts about a domain, a corpus of textual documents related to labour market was selected to automatically extract domain knowledge. Documents are mainly unstructured texts focusing on occupation descriptions derived from the ISTAT classification scheme, and on textual documents produced by domain experts.

An experiment has been performed by focusing on a subset of occupation descriptions extracted from the whole corpora (about 400 descriptions). The subset has been identified in this way: data about worker transitions among the several occupations has been considered. The data of labour market are extracted from the administrative archives of an Italian Region. The term transition refers to a worker dismissing an employee and starting a new one having a different qualification. The transitions from an occupation to

a different one have been counted, in a given observation period. A high transition number means that the two considered occupations are related. Then an occupation subset was selected performing a cluster analysis using the transition count as relatedness criterion [17][18]. This allows one to identify groups of related occupations representing consistent areas and coherent worker career paths. It is out of the scope of this paper to describe in detail the clustering algorithm, it is enough to report that several clusters were identified, and a cluster of 30 occupations related to ICT domain (in the following ICT cluster) has been selected for further analysis. A few operations performed on the ICT cluster are described below for extracting terms and expanding the ontology.

1) *Cleansing and stopwords elimination*

In this first operation, the documents containing the occupation descriptions are cleansed by removing the stop words, e.g., articles, pronouns, adjectives, prepositions and conjunctions, common adverbs and non-informative verbs (e.g., to be). The stop words carry on little informative content, indeed they get discarded. A list of stop words was built and checked in the documents contents to eliminate these words from it.

2) *Keyword extraction, indexing and weighting*

The second operation is extracting the keywords and measuring their degree of importance within the processed documents corpora. To this scope, an IR technique was used, namely the TF-IDF (Term Frequency, Inverse Document Frequency) weighting scheme to extract, and assign weights to distinguished terms in a document [19][20][21]. Each document is described by a set of representative keywords called index terms. An index term is simply a word whose semantics distinguishes the document's main themes. The assignment of numerical weights captures the ability of an index term to distinguish the document's main themes.

The intuition behind using the TF-IDF is that the best term candidates for inclusion in the ontology are those featured in certain individual documents, capable of distinguishing them from the remainder of the collection. This implies that the best terms should have high term frequencies (within the document), but low overall collection frequencies (i.e., it appears in very few other documents). The term importance is therefore obtained by multiplying the term frequency with the inverse document frequency (see for further details [22]). The result of the operation is a set of weighted keywords (i.e., nouns and verbs). The ones having the highest values identify the meaning and the terminology of documents. In the experiment performed 619 index terms were extracted from the ICT cluster.

3) *Grouping terms in two groups, common and specific concepts*

In this phase the set of index terms is analyzed to distinguish the more generic from the more specific. This distinction is just required to identify the position that each term could take within an ontology (i.e., in which class to include them). Some of the generic terms have been allocated to fill the WiWork Core ontology classes, while the

specific terms have been used to fill the Labour Market ontology classes. On the basis of the results of previous operations, it has been observed by domain experts that the generic terms are those having the lowest TF-IDF values, while the specific ones have the highest values. Through few operations of further cleansing (e.g., all the repetitions arising when considering the terms of several documents) a set of 360 index terms was obtained. In this group, the 70 highest TF-IDF values were selected representing the specific terms of the ICT cluster (19%). These terms are the best candidates to be defined and included as instances in the Labour Market ontology. Some of the remaining terms are the candidates for the WiWork Core ontology.

4) *Relationships identification*

This task is still under development, during the experiment it was performed manually, but in the future the authors would improve these steps with semi-automatic processes. This operation took as input the two groups of terms (i.e., the specific and the generic set) and started manually building the relationships connecting the ontology concepts and instances. Some relationships can be mainly induced by the verbs connecting relevant words in the occupation descriptions. The set of 70 highest TF-IDF values includes some verbs representing specific actions or tasks of only a few occupations. An example of this type is the verb *to write* extracted from the description of *Journalist* occupation. It can be easily put as instance of class *Task*, in relation with the instance *Journalist* of class *Role* (see below the ontology classes). While the most generic verbs, e.g., *to control*, *to perform*, *to use*, that more frequently appear in the occupation descriptions (and having the lowest TF-IDF values), describe common actions and tasks of many occupations. They can be managed as relations linking instances of, e.g., *Occupation* or *Role* classes with *Tool* or *Product* classes.

5) *Concepts and relationships integration into ontologies*

Nevertheless, ontologies modeled with index terms extracted from corpora of documents hardly represent an exhaustive terminology of a domain. An integration of domain concepts is required to provide a complete knowledge of that domain.

The approach planned for integrating other concepts and relationships into the ontologies is based on the comparison of terms and relations (identified during the previous operations) with classes and properties defined in domain dictionaries or specialized taxonomies as external data sources. In the Labour Market domain some specialized taxonomies, i.e., ILO (International Labour Organization) and O*Net [23][24], are internationally shared as the best semantic resources for the domain terminology. They provide the skeleton on which such ontologies can be built, mainly contributing to the classification of domain concepts.

Moreover, searching RDF or OWL ontologies existing on the Web that already defined and modeled those terms and relations could enrich such task. Thus is possible, for example, by using Semantic Web search engines (e.g.,

Sindice [25]), or by querying Linked Data [26] repositories via SPARQL (Simple Protocol and RDF Query Language) endpoints [27] (e.g., Dbpedia [28]), to retrieve useful semantic resources and to reuse concepts and relationships.

C. Some results

The experiment on the ICT cluster produced a set of candidate terms modeled as concepts and instances in the ontologies. The WiWork Core ontology concepts are mainly generic terms, including: *Person, Service, Institution, Technology, Knowledge Occupation*. The Labour Market ontology main concepts include *Economic Sector, Product, Material, Tool, Event, Role*. Within the latter ones several entities (i.e., concepts, relationships, and instances) could be defined from the set of candidate terms extracted with the IR techniques. Nevertheless, in many cases authors noted that some terms are very specific and appear in only one occupation description, these terms could be managed as instances of ontology classes.

In Table 1 some candidate terms are classified according to the Labour Market ontology classes.

TABLE I. CANDIDATE TERMS

Economic Sector	Product	Tool	Role	Event
cinema	calculation	airplane	client	congress
electromechanics	design	calculator	journalist	cruise
electronics	document	computer	personnel	exhibition
management	video/sound recording			fair
maritime	filming			manifestation
office management	performance			stage
public administration	planning			
theatre	receipt			
tourism	report			
training	survey			

All those terms derive from the occupation descriptions and belong to the group of specific terms with the highest TF-IDF values (see point 3 in the previous Section B). Ontology classes and relationships have to be refined from the initial classification taxonomy of candidate terms. By taking as main references the ILO and O*Net resources it is possible to specify some ontology parts, e.g., *Sector* class with its sub-classes and instances as represented in Figure 1.

Classes		Instances
Sector		
EconomicSector		
Industry		
ElectricalIndustry		electromechanics
ElectronicComputerIndustry		electronics
PublicSector		
GovernmentPublicAdministration		public_administration
ServiceSector		
Education		training
EntertainmentIndustry		cinema
Tourism		hotel
Transport		
SeaTransport		maritime

Figure 1. Classes and instances of Labour Market ontology

The candidate terms may be then annotated by users in WiWork pages using the property and category labels defined in the ontologies, and transformed in SMW constructs with the support of Protégé-OWL API and Pywikipediabot with an upcoming program described above. Moreover, users could propose new terms or lexical variations to include in the ontological models by exploiting the collaborative wiki functions.

IV. RELATED WORK

Several semantic wikis based on SMW extension exist, in various domains. To the best of the authors knowledge a similar semantic wiki on the labour market domain does not exist. One of the closest wikis to the one introduced in this paper is that of [29], that showed how difficult it is to find the right balance between structured and unstructured data in a semantic portal of a research centre. Another wiki based on ontological models is LinkedLab [30], a Linked Data solution for data management regarding research communities publishing and linking structured data on the Web using RDF.

Important for this paper is also the work of [31] giving an overview of how semantic wikis manage structured, semi-structured and unstructured data. In the field of ontology engineering the use of wikis is a widespread research topic, confirmed by works of [32][6].

Within the enormous literature related to ontology building some methodologies helped authors. In particular, the methodologies characterized by the synergy between various disciplines such as text mining, knowledge acquisition from texts, IR, corpus linguistics or even terminology [16]. In this context, authors found similarities with the research presented in this paper in the work of [33] that uses the TF-IDF to select the best candidate terms for ontologies, with the difference that their work is based on n-gram detection. It is also relevant the work of [34] that uses TF-IDF to identify different types of relationships in a specific step of his ontology building methodology.

V. CONCLUSION AND FUTURE WORK

The paper proposed a semantic wiki, WiWork, to enrich wiki unstructured contents using semantic annotations referring to ontological models. The paper showed also how to model domain ontologies required to structure the labour market domain knowledge managed by the wiki. The semantic wiki showed several advantages: it facilitates the task of structuring the knowledge according to ontologies, it guides users in annotating texts in an easy way and it provides means to process and reuse machine-readable knowledge.

Moreover, a particular methodology was presented based on IR and statistical techniques to support the ontology modeling tasks. The results of an initial experimentation in this direction are showed, and they represent part of a work that authors are pursuing within a multidisciplinary research team.

In the future, the authors plan to include into the WiWork architecture some reasoning and querying features and to experiment it with both structured data (e.g., user-generated annotations encoded in RDF) and unstructured one (e.g., wiki pages about health domain). Moreover, WiWork will be made available on the Web and linked with triple stores and SPARQL endpoints, with the aim to support publishing of semantic documentation for Linked Open Data sets about public sector domains.

REFERENCES

- [1] D. Beckett, RDF/XML syntax specification (Revised) - W3C Recommendation 2004.
- [2] D. McGuinness and F. van Harmelen, OWL Web Ontology Language - W3C Recommendation 2004.
- [3] M. Krötzsch, D. Vrandečić, and M. Volkel, "Semantic Mediawiki," Proc. 5th International Semantic Web Conference (ISWC 06), vol. 4273, Springer, 2006, pp. 935-942.
- [4] B. Leuf and W. Cunningham, *The WIKI way quick collaboration of the Web*, Addison-Wesley, 2001.
- [5] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel, "Semantic wikis," *Software*, vol. 25, No. 4, 2008, pp. 8-11.
- [6] F. Bry, M. Eckert, J. Kotowski, and K. Weiland, "What the user interacts with: reflections on conceptual models for semantic wikis," Proc. 4th Semantic Wiki Workshop (SemWiki 09), Hersonissos, Greece, 2009.
- [7] CRISP Research Centre Web page, <http://www.crisp-org.it>.
- [8] M. Martini and M. Mezzananza, "The federal observatory of the labour market in Lombardy: models and methods for the construction of a statistical information system for data analysis," in *Information systems for regional labour market monitoring - State of the art and perspectives*, C. Larsen, M. Mevius, J. Kipper, and A. Schmid, Eds. Rainer Hampp Verlag, 2009.
- [9] ISTAT Web page: <http://www.istat.it/>
- [10] ISCO08 Web page: <http://www.ilo.org/public/english/bureau/stat/isco/isco08/index.htm>
- [11] Pywikipediabot Web page: <http://www.mediawiki.org/wiki/Manual:Pywikipediabot>
- [12] Semantic Forms Web page: http://www.mediawiki.org/wiki/Extension:Semantic_Forms
- [13] Protege OWL API Web page: <http://protege.stanford.edu/plugins/owl/api/>
- [14] N. F. Noy and D. L. McGuinness, *Ontology development 101: a guide to creating your first ontology*, 2001, [Online]: http://protege.stanford.edu/publications/ontology_development/ontology101.html.
- [15] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, No. 5-6, 1995, pp. 907-928.
- [16] N. Aussenac-Gilles and J. Mothe, "Ontologies as background knowledge to explore document collections," Proc. *Coupling approaches, coupling media and coupling languages for information retrieval (RIAO 04)*, Avignon, France, 2004, pp. 129-142.
- [17] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Norwell: Kluwer Academic Publishers, 1981.
- [18] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Computational and applied mathematics*, vol. 20, 1987, pp. 53-65.
- [19] K. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, No. 1, 1972, pp. 11-21.
- [20] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, New York: AddisonWesley, 1999.
- [21] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition*, Pearson Prentice Hall, 2008.
- [22] G. Salton and C. Buckley, "Term weighting approaches in automatic retrieval," *Information Processing and Management*, vol. 24, No. 5, 1988, pp. 513-523.
- [23] ILO - International Labour Organization Web page: <http://www.ilo.org>.
- [24] O*Net - The Occupational Information Network Web page: <http://www.onetcenter.org>.
- [25] Sindice Web page: <http://sindice.com/>.
- [26] T. Berners-Lee, *Linked Data - design issues*, 2006, [Online]: <http://www.w3.org/DesignIssues/LinkedData.html>.
- [27] E. Prud'hommeaux and A. Seaborne, *SPARQL query language for RDF. W3C recommendation*, 2008. <http://www.w3.org/TR/rdf-sparql-query/>
- [28] Dbpedia Web page: <http://sindice.com/>
- [29] D. M. Herzig and B. Ell, "Semantic MediaWiki in operation: experiences with building a semantic portal," Proc. *The Semantic Web (ISWC 10)*, vol. 6497, 2010, pp. 114-128.
- [30] F. Darari and R. Manurung, "LinkedLab: a Linked Data platform for research communities," Proc. *Advanced Computer Science and Information System (ICACSIS)*, Jakarta, Indonesia, 2011, pp. 253-258.
- [31] R. Sint, S. Schaffert, S. Stroka, and R. Ferst, "Combining unstructured, fully structured and semi-structured information in semantic wikis," Proc. *4th Semantic Wiki Workshop (SemWiki 2009)*, Hersonissos, Greece, 2009.
- [32] S. Auer, S. Dietzold, T. Riechert, and T. Riechert, "OntoWiki - a tool for social, semantic collaboration," Proc. *5th International Semantic Web Conference (ISWC 06)*, Springer, 2006, pp. 736-749.
- [33] K. Englmeier, F. Murtagh, and J. Mothe, "Domain ontology: automatically extracting and structuring language community from texts," Proc. *IADIS International Conference Applied Computing*, Salamanca, Spain, 2007, pp. 59-66.
- [34] Y. Rezgui, "Text-based domain ontology building using tf-idf and metric clusters techniques," *The Knowledge Engineering Review*, vol. 22, No. 4, 2007, pp. 379-403.

Translating Natural Language Competency Questions into SPARQL Queries: A Case Study

Leila Zemmouchi-Ghomari

UMBB (M'hammed Bouguerra Boumerdès University)
Boumerdès, Algeria
l_ghomari@umbb.dz

Abdessamed Réda Ghomari

LMCS Laboratory
E.S.I (National Superior School of Computer Science)
Algiers, Algeria
a_ghomari@esi.dz

Abstract—Ontology validation is an important part of measuring the quality of an ontology, and the best way to assure the accuracy of the knowledge encoded in the ontology. One of the earliest approaches toward ontology evaluation was the introduction of competency questions, i.e., natural language questions that the ontology should be able to answer. Since the ontology is a machine readable representation of knowledge, end-users should be able to query it using a formal language, such as SPARQL; however, translating natural language competency questions into SPARQL queries is not a trivial task. In the scope of this paper, we consider competency questions of HERO (Higher Education Reference Ontology) ontology we have developed. We translated these competency questions into SPARQL queries using a variation of a known approach.

Keywords-Competency question; SPARQL query; Ontology validation; translation.

I. INTRODUCTION

Competency questions (CQs) [1] are the set of requirements or needs that the ontology should fulfill; they can be considered as a test collection, providing value during ontology analysis and validation [2].

According to Presutti et al. [3], CQs are used through the whole ontology development; the validation will be achieved by:

- Formalizing competency questions in the form of queries;
- Associating each query with the expected result;
- Running the queries against the ontology;
- And comparing actual with expected results.

So, in order to enable automatic evaluation with regard to competency questions, they need to be formalized in a query language. The query language has to be expressive enough to encode the competency questions appropriately.

We support the fact that SPARQL (Simple Protocol And RDF Query Language) [4] can represent a wide range of natural language questions, this language allows a high expressivity by representing and interrogating data as instances of concepts and relations defined in a reference ontology [5]. In addition SPARQL is the language proposed by W3C for querying RDF (Resource Description Framework) [6] data published on the Web.

Though translating natural language competency questions into SPARQL queries is not a trivial task [7][8].

To the best of our knowledge, automatic translation of competency questions into SPARQL queries, with the aim of validating an ontology, has not been tackled by researchers.

Although, in a more general perspective, there exist several approaches dedicated to web Question Answering (QA) area, which can potentially be exploited in addressing our specific issue. An overview of these approaches is presented in Section II. In Section III, we will describe our proposed approach. The translation process of HERO [9] competency questions into SPARQL queries is explained in Section IV and we will conclude our paper in Section V.

II. RELATED WORK

Several web QA approaches supported in most cases by platforms have been proposed to function as either natural language ontology editors, such as CNL editor [10] and OWLPATH [11], or natural language query systems like PANTO [12] and DEANNA [13]. Other approaches address this issue for a specific knowledge domain, such as: the medical domain in [14]. Table I summarizes the main features of each approach.

TABLE I. SOME WEB QA APPROACHES

Approach	Description
CNL editor	Formerly ontopath, it is composed of "OntoPath-Syntax", "OntoPath-Object" and "OntoPath-Semantic". After defining a set of concepts and relationships, the system returns the RDF ontology, and then natural language is expressed graphically by representing ontology elements, next the query is formed from the knowledge available in ontology and translated into RDF. CNL editor extends OntoPath in providing a context-free grammar with lexical dependence for defining grammars.
OWLPATH editor	It uses controlled language and grammar which are determined by question ontology. Defining the grammar using OWL ontology has two main advantages: the use of reasoners for consistency checking and the possible inclusion of restrictions in the properties of the question ontology. Thus, the grammar can take into account these restrictions while the sentence is being entered.

<p>PANTO</p> <p>Portable Natural Language Interface to Ontologies</p>	<p>WordNet and string metrics algorithms are integrated into PANTO system to help make sense of the words in the NL queries and map them to the entities in the ontology. Then nominal phrases are extracted in the parse trees as pairs to form <i>QueryTriples</i>. Next, by using knowledge in the ontology, PANTO maps <i>QueryTriples</i> to <i>OntoTriples</i> which are represented with entities in the ontology. Finally, together with <i>targets</i> and <i>modifiers</i> extracted from the parse trees, <i>OntoTriples</i> are interpreted as SPARQL queries.</p>
<p>DEANNA</p> <p>Deep Answers for Naturally Asked Questions</p>	<p>This method is composed of six phases: first, semantic items are extracted from natural language questions then they are mapped to potential knowledge bases entities. The next phase generates candidate triples which are disambiguated in order to form semantic triples. On the basis of these triples a SPARQL query is generated.</p>
<p>Ben Abacha& Zweigenbaum approach:</p> <p>Translating Medical Questions into SPARQL Queries</p>	<p>This approach is composed of six phases, that is: Identifying the question type (e.g., WH: What, Who, Why, etc., Yes/No, Definition) then Determining the Expected Answer(s) Type(s) for WH questions next Constructing the question's affirmative and simplified form (new form). The following phase is Medical Entity Recognition and Relation Extraction based on the new form of the question and finally, SPARQL Query Construction</p>

There is a limitation shared by all described approaches which is scalability, as the ontologies used for test purposes are relatively small.

Several approaches, such as DEANNA and PANTO, suppose that for every queried knowledge base, there exists a dictionary that maps questions' concepts to potential knowledge bases' semantic items, which is obviously tricky to carry out and to maintain, particularly for huge knowledge bases such as DPEDIA, Yago and Freebase. In addition, Vocabularies of the knowledge bases are controlled, so it is a challenge to correctly map users' words to vocabularies of the knowledge bases [12].

The lastly mentioned approach is limited to a particular set of questions: WH questions, except complex ones (why and when).

III. COMPETENCY QUESTIONS TRANSLATION APPROACH

Compared with web Question-Answering issue in general, our proposal tackles a specific issue. Actually, we target particular users, namely, knowledge/ontology engineers who are involved in an ontology building process. And in order to validate built ontology, they need to translate ontology specification in the form of natural language competency questions into SPARQL queries.

Expertise of these users leads us to make three assumptions before describing the proposed approach, i.e., our users are familiar with:

- Formal ontology languages (RDF or OWL) and web query languages used over ontologies/knowledge bases (SPARQL).
- Structure and vocabulary of the queried Ontology

The third assumption is related to extracted terms from competency questions which are similar to terms used to name ontology entities, according to NeOn methodology guidelines [15].

We investigated the related work (Section II) to find a methodological baseline in order to carry out a practical case study rather than a ready-to-use toolset, which has not yet been approved broadly by web QA community.

In our opinion, Ben Abacha & Zweigenbaum approach [14] fits to some extent to our needs. Actually, it is quite intuitive and relatively simple.

However, this approach is specific to the medical field, as explicitly mentioned in phase 4 of the approach; in addition, phase 2 shows that the approach focuses on a subset of WH questions which is not our intention.

Hence, we decided to slightly adapt it to our needs and the resultant approach can be summarized in five steps as illustrated by Figure 1:

- 1) *Identifying competency questions' categories according to expected answers' types* [14]: there are five sets of questions which are:
 - a) Definition Questions: that begins with “*What is/are*” or “*What does mean*”
 - b) Boolean or Yes/No Questions
 - c) Factual Questions: the answer is a fact or a precise information
 - d) List questions: the answer is a list of entities
 - e) Complex Questions: that begins with “*How*” and “*Why*”, in this case, obtaining a precise answer is almost improbable.
- 2) *Determining the expected (perfect or ideal) answer;*
- 3) *Extracting Entity or Entities from questions and their corresponding expected answers identified in 2;*
- 4) *Identifying answer entity type (class, data property, object property, annotation, axiom, instance) and entity location in the ontology;*
- 5) *Constructing the appropriate SPARQL query that gives the closest answer to the ideal answer:* based on question type identified in phase 1, question/answer entity extracted from phase 3 and its corresponding entity type/entity location in the ontology from phase 4 (as illustrated by input arrows pointing to “SPARQL Query Construction” in Figure 1).

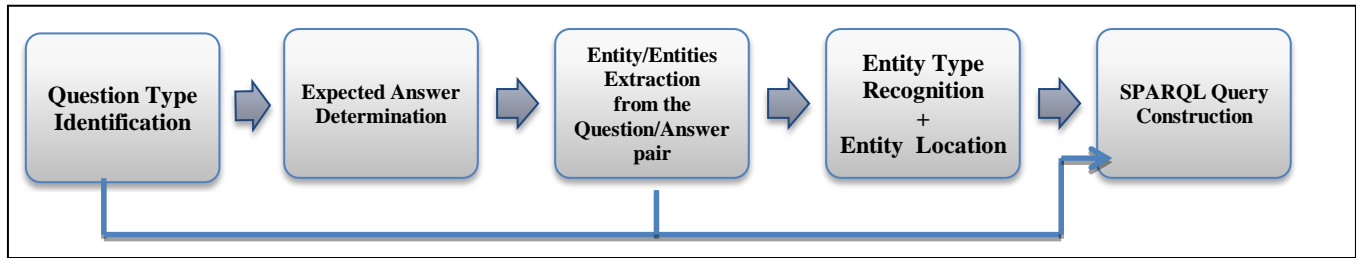


Figure 1. Competency Questions Translation Approach

IV. COMPETENCY QUESTIONS TRANSLATION PROCESS

Based on the approach description in the previous Section, we carry out the translation process of HERO ontology competency questions into SPARQL queries.

A. Identifying Question's Category

As a first attempt to detect HERO ontology requirements, we have identified eighty one (81) Competency questions (CQs) in the specification phase of HERO ontology development process; these CQs have been divided into six (6) sets according to sub-domains of higher education knowledge domain, namely: Faculty, appointments and research area, students and their life, administration, Degrees and curriculum, programs, finance, governance.

Another classification of these CQs is required according to answers types, as mentioned in the previous Section. An example of each question category is provided in Table II (CQs' numbering is similar to the one used in the full list of HERO CQs [16].

TABLE II. SOME EXAMPLES OF HERO CQs ACCORDING TO THEIR RESPECTIVE CATEGORIES

CQs' Categories	CQs' Examples
Definition questions	CQ59.What is a Credit?
Yes/No questions	CQ3. Must a university teacher be a researcher?
Factual questions	CQ44. What average size and duration have governing board?
List questions	CQ1. What are the possible academic ranks of a teacher?
Complex questions	CQ41.Why universities are organized into departments?

This sorting will facilitate the construction of the corresponding SPARQL queries, for example in the case of:

- *Definition question*, in our opinion, a combination of SPARQL queries can permit to provide as much information as required; we can divide this combination into five categories, to be precise:
 - 1) Ascendants or super classes
 - 2) Descendants or sub classes
 - 3) List of descriptive properties or data properties
 - 4) Relations or object properties
 - 5) And annotations (definition, comment, label).

The combination has not to be complete every time the definition question is met; it depends on the scope of the expected answer.

- *Yes/No questions*, in this case, ASK form of the SPARQL query will be preferred over the other forms, i.e., SELECT, CONSTRUCT and DESCRIBE, since it provides a Boolean response;
- *Factual questions versus List questions*: in the case of factual questions, we know that the query has to target one specific entity which might be a class, an instance or whatever, at the opposite of list questions where we have to obtain a number of entities as a single answer;
- *Complex questions*, often require a detailed answer, for example: in what manner things are done or causes of some phenomena. That is why we think that in most cases, corresponding ontology annotations are considered as best answers to this type of questions;

B. Determining the Expected Answer

HERO competency questions answers come from several information sources, such as: governmental academic websites, official higher education reports, experts' interviews, etc. Some examples of these answers are presented in Table III.

TABLE III. HERO COMPETENCY QUESTIONS' ANSWERS (EXCERPT)

CQs' Categories	CQs' Examples	Corresponding Answers
1	CQ59.What is a Credit?	Each course bears a specified number of credits. In general, the number of credits a course carries is determined by the number of class hours the course meets each week.
2	CQ3. Must a university teacher be a researcher?	Nearly all faculty members are expected to engage in research.

3	CQ44. What average size and duration have governing board?	The average size of public boards is approximately 10 people and the average size among independent (private) institutions is 30. The length of board members' terms varies from three years to as long as 12 years.
4	CQ1. What are the possible academic ranks of a teacher?	Assistant Professor, Associate Professor, Full Professor, Professor Emeritus.
5	CQ41. Why universities are organized into departments?	The basic unit of academic organization in most institutions is the department (e.g., chemistry, political science). Every department belongs to an academic field.

C. Entity Extraction from the Competency Question/Answer

From both competency questions and their expected answers, we carried out a manual extraction of relevant terms which preferably should be equivalent to some ontology entities; elsewhere the SPARQL query will not obtain an answer encoded in the ontology. This condition is valuable to warn the ontology evaluator, that it is necessary to update the ontology by adding the missing entity.

This extraction is based on a mapping between relevant terms in questions/answers pairs and their equivalent terms in ontology; it is limited to a syntactic mapping with regard to the third assumption mentioned in Section III. An excerpt of this mapping is shown in Table IV:

TABLE IV. ENTITIES' EXTRACTION FROM HERO COMPETENCY QUESTIONS AND THEIR ANSWERS (EXCERPT)

CQs' Relevant Terms	Answers' Relevant Terms	Corresponding ontology terms
CQ59...Credit?	...course ... number of credits.	Course, Credit Number
CQ3. ... teacher ... researcher ?	engage in research	Teacher, Researcher
CQ44. ...size ..duration... governing board ?	...10 ...30 people ...varies from three years to as long as 12 years	Size, Duration, Governing Board
CQ1. ...ranks...teacher ?	Assistant Professor, Associate Professor, Full Professor, Professor Emeritus	Rank, Teacher, Assistant Professor, Associate Professor, Full Professor, Professor Emeritus
CQ41... universities ... organized into departments?	... basic unit ... is the department... Every department belongs to an academic field.	Higher Education Organization, Department

D. Identifying Entity Type and Entity Location

Competency questions answers must be represented in one of the allowed forms of ontology entities like: classes, data properties, object properties, axioms, instances and annotations.

SPARQL query syntax is highly dependent on the entity type of the expected answer, for example:

1) when the answer is an *INSTANCE*, the SPARQL query will then be:

```
SELECT * WHERE
{?Teacher rdfs:type HERO:Teacher . }
```

2) when the answer is a *CLASS*, the SPARQL query will then be:

```
SELECT * WHERE
{ ?subclass rdfs:subClassOf HERO:Student . }
```

Another indispensable parameter to construct an efficient SPARQL query is the location of the expected

answer in the ontology. This parameter can directly target the required information, for example: when the expected answer is located in an annotation (definition, label, comment) of a class, the SPARQL query (CQ62. What are articulation agreements?) will then be:

```
SELECT ?definition WHERE
{HERO:ArticulationAgreement rdfs:isDefinedBy
?definition . }
```

We pursue the translation process of competency questions by identifying entity types of each extracted entity from the question/answer pair and locate it in the ontology using ontology editor search function, on one hand, and on the other hand, the support of ontology engineer who knows the vocabulary and the syntax of the ontology (second assumption, Section III). The result of this identification is presented in Table V:

TABLE V. ENTITIES' TYPES AND LOCATIONS IDENTIFICATION (EXCERPT)

CQs	Entities' Types	Entities' Locations in the ontology
CQ 59	Class: Course Data Property: CourseCreditsNumber	• CourseCreditsNumber Domain Course
CQ 3	Classes: Teacher, Researcher	• Teacher SubClassOf Researcher
CQ 44	Class: Governing Board Data Properties: Size, Duration	• GoverningBoardSize Domain GoverningBoard • GoverningBoardDuration Domain GoverningBoard

CQ 1	<i>Class:</i> Teacher <i>Data Property:</i> Rank, Assistant Professor, Associate Professor, Full Professor, Professor Emeritus	<ul style="list-style-type: none"> • TeacherRank Domain Teacher • AssistantProfessor <i>SubPropertyOf</i> TeacherRank • AssociateProfessor <i>SubPropertyOf</i> TeacherRank • FullProfessor <i>SubPropertyOf</i> TeacherRank • ProfessorEmeritus <i>SubPropertyOf</i> TeacherRank
CQ 41	<i>Classes:</i> Higher Education Organization, Department	<ul style="list-style-type: none"> • Department <i>SubClassOf</i> Faculty • Faculty <i>SubClassOf</i> Role • Role <i>SubClassOf</i> HigherEducationOrganization • Department <i>Definition</i>

E. Constructing SPARQL query

Once the ideal answer identified, the equivalent entity type recognized and the localization in the ontology has

been achieved; the construction of the corresponding SPARQL query can be written (facilitated by first assumption, Section III), as displayed in Table VI:

TABLE VI. SPARQL QUERIES

Competency Questions	SPARQL Queries
CQ59.What is a Credit?	SELECT ?comment WHERE { HERO:CourseCreditsNumber rdfs:comment ?comment }
CQ3. Must a university teacher be a researcher?	ASK {HERO:Teacher rdfs:subClassOf HERO:Researcher . }
CQ44. What average size and duration have governing board?	SELECT ?university ?size WHERE { ?university rdf:type HERO:HigherEducationOrganization; ?y rdfs:subClassOf ?university ; ?y HERO:GoverningBoardSize ?size }
	SELECT ?university ?duration WHERE { ?university rdf:type HERO:HigherEducationOrganization ; ?y rdfs:subClassOf ?university ; ?y HERO:GoverningBoardDuration?duration }
CQ1. What are the possible academic ranks of a teacher?	SELECT ?a ?b ?c ?d WHERE { ?a rdfs:subPropertyOf HERO:TeacherRank. ?b rdfs:subPropertyOf ?a . ?c rdfs:subPropertyOf ?b . ?d rdfs:subPropertyOf ?c . }
CQ41.Why universities are organized into departments?	SELECT * WHERE { HERO:Department rdfs:subClassOf ?x ; OPTIONAL { ?x rdfs:subClassOf ?y ; OPTIONAL { ?y rdfs:subClassOf HERO:HigherEducationOrganization } } }
	SELECT ?definition WHERE { HERO:Department rdfs:isDefinedBy ?definition . }

Notice that when a single SPARQL cannot provide all identified entities, it is possible to translate a competency question into several SPARQL queries, e.g., CQ41, CQ44. Another alternative could be to make an UNION between all necessary sub queries.

V. CONCLUSION AND FUTURE WORK

Natural language competency questions translation into SPARQL queries is a *sine qua non* condition for automatic evaluation of ontology requirements satisfaction.

A well defined approach of this translation process is critical for ontology evaluation area in particular and for machine readable question answering field in a more general perspective.

Based on our intuition and on some precious guidelines provided by Ben Abacha & Zweigenbaum approach [14], we achieved the translation of Higher Education Reference Ontology (HERO) competency questions into SPARQL queries.

We are conscious that our work encompasses several limitations, such as:

- Two crucial phases of our approach are entirely manual and totally dependent of user knowledge background, namely: Entity extraction from questions/answers pairs and mapping between questions/answers relevant terms and ontology entities; semi automatic support of these phases should be considered. We suggest the use of a natural language processing tool like GATE [17] in terms extraction phase and automatic matching systems such as COMA 3.0 [18] to carry out the mapping phase.
- Weak treatment of complex questions, more precise answers are preferred to ontology annotations.

Obviously, more empirical evaluation on the approach is required to assess its performance and its effectiveness on one hand and to test HERO ontology with broader benchmark of competency questions provided by domain experts or end-users for example.

Despite these limitations, we are convinced that sharing experiences can significantly help research progress in web question answering processing.

VI. REFERENCES

- [1] M. Gruninger and M. S. Fox, "Methodology for the design and evaluation of ontologies", IJCAI95, Workshop on Basic Ontological Issues in Knowledge Sharing. Montreal, 1995, pp. 6.1–6.10.
- [2] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith, The evaluation of ontologies, in *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, C.J.O. Baker and K.-H. Cheung, Eds. New York: Springer-Verlag, 2007, pp. 139–158.
- [3] V. Presutti, E. Daga, A. Gangemi, and E. Blomqvist, "Extreme design with content ontology design patterns", Proceedings of the workshop on ontology patterns (WOP 2009), collocated with ISWC 2009, Washington D.C, USA, Vol.516, October 2009, pp. 83-97.
- [4] <http://www.w3.org/TR/rdf-sparql-query/>, [retrieved: Dec, 2012].
- [5] L. Zemmouchi-Ghomari and A. R. Ghomari, "Reference Ontology", Fifth International Conference of Signal-Image Technology & Internet-Based Systems (SITIS), December 2009, Marrakech, Morocco, pp.485-491.
- [6] <http://www.w3.org/RDF/>, [retrieved: Dec, 2012].
- [7] D. Damjanovic, D. M. Agatonovic, and H. Cunningham, Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction, In *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science, Volume 6088, 2010, pp 106-120.
- [8] R. J. Mooney, "Using multiple clause constructors in inductive logic programming for semantic parsing", Proceedings of the 12th European Conference on Machine Learning, Freiburg, Germany September 2001, pp. 466–477.
- [9] <http://sourceforge.net/projects/heronto/?source=directory>, [retrieved: Dec, 2012].
- [10] H. Namgoong and H. G. Kim, "Ontology-based controlled natural language editor using CFG with lexical dependency," Proceedings ISWC/ASWC, Vol. 4825, Lecture Notes in Computer Science, Busan, Korea, 2007, pp. 353–366.
- [11] R. Valencia-García, F. García-Sánchez, D. Castellanos-Nieves, and J.T. Fernández-Breis, "OWLPath: An OWL ontology-guided query editor," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, Vol. 41, no.1, 2011, pp. 121–136.
- [12] W. Chong, X. Miao, Z. Qi, and Y. Yong, "PANTO: A Portable Natural Language Interface to Ontologies", In Proceedings of the European Semantic Web Conference, volume 4519 of Lecture Notes in Computer Science, Springer-Verlag, July 2007, pp. 473-487.
- [13] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Deep answers for naturally asked questions on the web of data". Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion, Lyon, France, April 2012, pp. 445-449.
- [14] A. Ben Abacha and P. Zweigenbaum, "Medical Question Answering: Translating Medical Questions into SPARQL Queries", Proceedings of the 2nd ACM SIGHT International Health Informatics Symposium, Miami, Florida, USA, 2012, pp. 41-50.
- [15] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, The NeOn Methodology for Ontology Engineering, Book Chapter in *Ontology Engineering in a Networked World*, 2012, Publisher: Springer Berlin Heidelberg, pp. 9-34.
- [16] <http://herontology.esi.dz/content/downloads>, accessed on 3 January 2012.
- [17] <http://gate.ac.uk/>, [retrieved: Dec, 2012].
- [18] <http://dbs.uni-leipzig.de/Research/coma.html>, [retrieved: Dec, 2012].

Extending Web Modeling Language to Exploit Stigmergy: Intentionally Recording Unintentional Trails

Aiden Dipple, Kerry Raymond, Michael Docherty

Faculty of Science and Technology
Queensland University of Technology
Brisbane, Australia
aiden.dipple@student.qut.edu.au
k.raymond@qut.edu.au
m.docherty@qut.edu.au

Abstract—Software development and Web site development techniques have evolved significantly over the past 20 years. The relatively young Web Application development area has borrowed heavily from traditional software development methodologies primarily due to the similarities in areas of data persistence and User Interface (UI) design. Recent developments in this area propose a new Web Modeling Language (WebML) to facilitate the nuances specific to Web development. WebML is one of a number of implementations designed to enable modeling of web site interaction flows while being extendable to accommodate new features in Web site development into the future. Our research aims to extend WebML with a focus on stigmergy which is a biological term originally used to describe coordination between insects. We see design features in existing Web sites that mimic stigmergic mechanisms as part of the UI. We believe that we can synthesize and embed stigmergy in Web 2.0 sites. This paper focuses on the sub-topic of site UI design and stigmergic mechanism designs required to achieve this.

Web Collaboration; virtual pheromones; stigmergy;

I. INTRODUCTION

Our research analyses a number of User Interface (UI) designs within popular Web 2.0 sites. The UI designs observed provide representations of user feedback along with representations of behavior trends from unintentional interactions which have been recorded. Examples of these UI designs can be seen in Facebook where users “Like” other user contributions causing an area of focused interest. Another example can be seen where Facebook has introduced a new “Seen By” representation of feedback where the number of users navigating to a specific article that has been broadcast is presented as a trail of evidence or virtual footsteps. This mechanism of users indicating content of interest and trail forming through unintentional footsteps closely matches a phenomenon called stigmergy. Stigmergy is a term originally used to describe the apparent decentralized coordination amongst certain insects when performing tasks such as food foraging and nest building [1]. While we observe that popular Web 2.0 sites contain these features there is no evidence to suggest that these designs were influenced by stigmergy. The UI designs observed in Facebook might have been introduced without initially understanding their similarity to stigmergy but we see an increasing number of web sites introducing similar

mechanisms trying to emulate the same success that has been achieved in Facebook. The primary research question we are working on is whether these mechanisms can be synthesized into a generic design pattern that can be introduced as standard Web site UI elements to enhance coordination.

Web Modeling Language (WebML) is a method of modeling data content, user interaction and navigation flow for various Web 2.0 applications. WebML provides a way to design the mapping of a data model to different UI views and the navigation paths between those views. Given the unique requirements of web site development compared to traditional software development the Object Model Group (OMG) is establishing a standard in the area. The OMG has released a current Request for Proposal (RFP) [2] to formalize syntax, metamodel, UML profile and associated interchange format for languages used to model interaction flow. WebML is one modeling language implementation currently being considered for inclusion in the standard. The most pertinent aspect of the WebML framework to our research question is that WebML is designed to be extensible to facilitate new concepts, interface types and event types. Given the Web 2.0 UI designs which we have observed and a thorough analysis of how they correlate to stigmergy we believe that we can introduce the UI mechanisms as standard elements during web site implementations.

II. STIGMERGY

Stigmergy is a biological term that was first introduced in 1959 by a French zoologist named Pierre-Paul Grasse [3]. The term was used to describe how insects appear to coordinate successfully despite having no centralized management structure or direct observable intercommunication [1]. Stigmergy specifically refers to an indirect communication where the insects use signs mediated within the environment to aid their coordination. An example of stigmergy can be seen in the way that ants leave a pheromone trail during food foraging activities. The trail provides a signal to other ants as to which direction a food source can be found while the strength of the pheromones indicate the relevancy of any specific trail as being the current trail to follow. A positive feedback system is created where the trail strength will increase as more ants follow the trail and successfully return with food. Furthermore, the environment enacts upon the sign causing atrophy and entropy to diminish the signal strength. This

decay provides the negative feedback to ensure only the most current trails can be sensed thereby ensuring that old trails don't interfere with the food foraging activities after the associated food supply has been depleted.

Previously [4] we have introduced a model of stigmergy including the concept of a stigmergy *grand purpose* and the core components of stigmergy: the *agent*, the *environment*, and the *sign*. The model is illustrated in Figure 1.

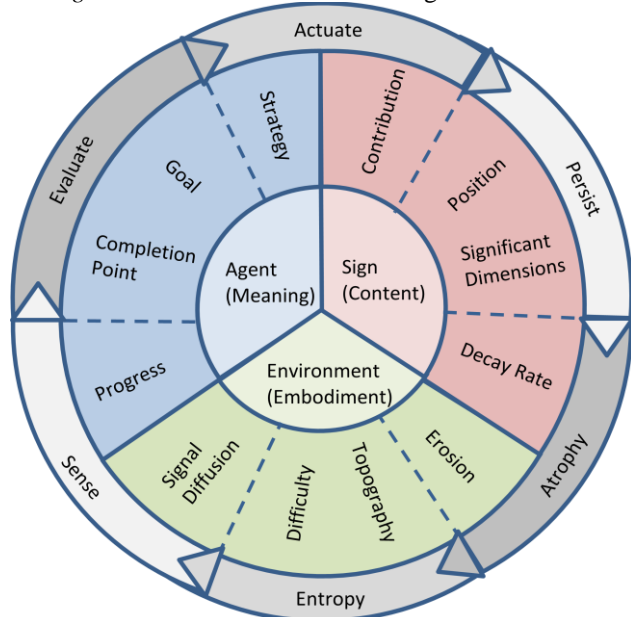


Figure 1. The Stigmergy Cycle.

The model as illustrated in Figure 1. ties together the core components of stigmergy, an inner band representing the *attributes* of the components, and an outer band representing the *dynamics* acting on those attributes. Furthermore, the outer band dynamics are either internal to each component, or defining the interface between components. Our model describes the dynamics of equilibrium between positive feedback (contributions) and negative feedback (decay) illustrating how the positive feedback is contributed by the agent, where the negative feedback is applied by the environment. The paper where we presented the model was focused on a holistic model of stigmergy which applies for the world of entomology, the human world and the virtual world. This paper is focused specifically on how the varieties of stigmergy manifest as Web environment UI elements. In context to this paper, these three components correlate to the users of Web environments, and the contributions that the users make.

There has been a significant amount of research focused on stigmergy in robotics and Web environments [5, 6]. Web environments provide a close facsimile to stigmergy in physical environments where a large number of users coordinate in a highly organized manner, specifically based on indirect communication through the contributions they make within the Web sites. In Facebook (see Figure 2.) there are similarities to the pheromone marker already observable in the Web.



Figure 2. Example of a Facebook LIKE mechanism.

Another variety of stigmergy describes the development of unintentional trails within the environment. An example of this is best shown by people wearing a path into a lawn when a short-cut is taken across the lawn. This unintentional trail is similar to another type of UI mechanism found within Facebook (see Figure 3.).

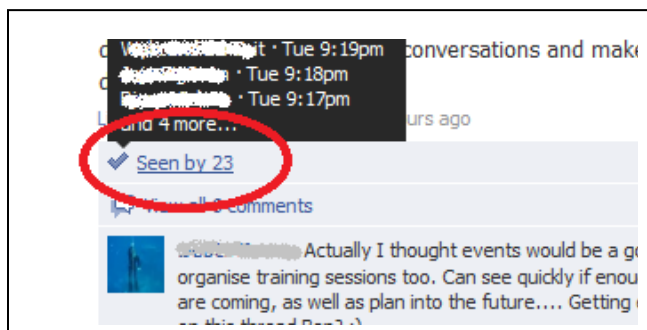


Figure 3. Example of a Facebook SEEN-BY mechanism..

Stigmergy provides a model of both *active contributions* and *passive interaction* with both varieties having numerous examples within the Web. The examples above for the two varieties of stigmergy have been categorized as *marker-based* [1] and *sematectonic* [7]. Marker-based stigmergy describes an explicit modification of the environment by leaving a sign with the intention of signaling other agents. Furthermore, Marker-Based stigmergy is broken into two sub-types: *qualitative* and *quantitative* [1]. This sub-type categorization is to clarify the difference between single contributions being sufficient to elicit a response as opposed to an accumulation of responses being required.

In contrast to the explicit method of leaving contributions, sematectonic stigmergy is defined as a modification to the environment as a by-product of actions being performed. These by-products are occurring inadvertently and unintentionally to the primary task being performed. For example, when considering a path being left in a lawn when people take a short-cut across it they have no intention of signaling to others that they have taken a short-cut. The short-cut is the purpose of the action, but the environment will retain the footstep impact as an alteration of the environment. There is no explicit foot-step left in the environment (obviously excluding cases such as wet feet leaving wet foot prints) however the action has altered the

environment and the cumulative foot-step action manifests in the format of a path rather than something recognizable as an aggregation of individual feet traces.

If we consider the two different varieties of stigmergy we can divide the notion of intentionality of communication as being either explicit or implicit [8, 9]. Marker-based stigmergy can be considered as an explicit form of communication where the contribution made by the agent is intentional; it is explicitly left with the intention of the *sign* being interpreted as a *signal*. Sematectonic stigmergy can be considered as implicit communication where the primary activity being performed by a user leaves implicit modifications to the environment unintentionally just as with a trail. We would consider that an explicit sign left unintentionally would not constitute a signal, but could be interpreted as one. Similarly we would consider that an intentional generation of an implicit *trail* would be a *counterfeit* and also not be considered stigmergic, although it must be noted that it can trigger the same behavior in agents receiving the signal.

III. WEB MODELING LANGUAGE

Web Modeling Language (WebML) is a platform independent way to express the interaction design, data model and business rules of Web application development separately from the implementation platform [10]. WebML permits the formal specification of the data model, interface composition and navigation options and ultimately be supported by tools for the auto-generation of code. WebML describes a visual notation for designing Web applications which is intended to be exploited by the visual design tool WebRatio [11].

WebML specifications are based on four perspectives:

- 1) *Structural Model*: Data Model for dynamic content.
- 2) *Hypertext Model*: Site Views of the data model, which in turn are comprised of two sub-models:
 - a) *Composition Model*: Page composition and how the data model maps to a view.
 - b) *Navigation Model*: How pages are inter-linked (contextually and non-contextually via passed parameters).
- 3) *Presentation Model*: Layout and graphic appearance of pages independent to output device.
- 4) *Personalization Model*: Individualisation of pages based on User / Group categories, preferences, etc.

The four modeling perspectives describe the principal facets of data-driven Web sites and therefore provide an excellent experimentation lab for our research when attempting to test stigmergic mechanisms. WebML represents specifications using XML. Examples of XML for the data entities of the structural model, data within a page view and the navigation links (including parameters in the case of contextual links) are provided.

IV. INCORPORATING STIGMERGY WITHIN WEBML

At this stage we are determining whether our implementation would be an extension to the existing WebML classes or a design pattern within the modeling

language using the existing classes. The simplest outcome of our research would be to prescribe a design pattern to follow when creating web sites which incorporate stigmergic mechanisms but we anticipate that a more prescriptive approach would be to create explicit stigmergic extensions to WebML. Initial examination of the WebML XML elements suggests that they would be easily extended to include additional attributes which support runtime instantiations of the stigmergic mechanisms. Furthermore the WebML submission to the OMG RFP describes *ViewComponent* as objects that display data or accept input [12]. This verifies that WebML can be extended to provide customized visualization components to display the stigmergic signals and trails to Web users.

We endeavor to design User Interface (UI) mechanisms to record both intentional and unintentional web site interaction based on our model of stigmergy. To provide environment mediated communication our UI mechanisms will need to trigger events which record user activity within a persistence layer specifically for stigmergy data. To enable the environment to provide negative feedback the environment must be able to trigger its own events to modify the stigmergy data. As discussed in Section III, WebML provides a collection of standard UI components with associated user and system triggered events to facilitate this within the *Hypertext Model*. WebML also provides the *Structure Model* that can facilitate stigmergy data persistence and access in conjunction to site specific data.

This section describes specific UI mechanisms available within the WebML *Hypertext Model* (including the ability to create custom controls) and how they apply to *input* and *output* components, events and persistence observed in examples of web-based stigmergy. We generalize these specific components into conceptual mechanisms which facilitate that contribution (input) and representation (output). For example, UI components such as drop-down lists, sliders and radio buttons are all representations of a single option selection, but each are different in their visual presentation. Understanding the fundamental mechanisms can extend the collection to encompass new UI components.

To incorporate stigmergy into WebML we need to understand the general ways the system would receive input, and how it should display output. If we consider our model of stigmergy (see Figure 1.) we understand that this will correlate with the environment having the facility to record contributions, process them and represent them back to users.

A. Qualitative, Marker-Based Stigmergy

Qualitative stigmergy is defined as discrete stimuli that can trigger a response in agents that encounter it. An example of this can be seen in Facebook with the "Share" functionality or within text messaging using emoticons [13]. Emoticons are icons included within the body of text messages to indicate to recipients the feeling or mood associated with the text (e.g.: "smiley faces"). The senders and recipients of text messages understand the associated meaning of the icons and as a result the emoticons add significant meaning to a text message without the use of language. The marker-based variety of stigmergy requires

an intentional contribution by the user, and must incorporate a UI control enabling the user to create the contribution.

In the example of the Facebook “Share” we observe that a single user can broadcast content discovered by the user performing the “Share” so that it is visible to that user’s group of friends (or level of privacy as selected). This represents the simplest input mechanism where there is only the requirement to intentionally trigger an interface element and that the user is aware of what signal it will contribute to.

The example of text messaging emoticons requires users to inherently know what emotions specific icons correlate to. There needs to be some UI mechanism for informing new users of the possible icons and their definition (e.g.: a context-sensitive, online help web function). Depending on how esoteric the required knowledge of available icons and their meaning are there might need to be instructions on appropriate reactions. The UI requires an administration function to define the possible icons in a given context. This administration function is considered outside the scope of this paper and is listed as an outstanding issue in Section V.

Our examples all generalize to one case: existing Web UI components which trigger an event (e.g.: radio buttons, etc). Each mechanism in the examples above equate to the user being presented with single or multiple options, however only a single option can be selected. We consider the generalized input mechanisms observed in the examples in this sub-section are:

- Single option intentionally triggering event.
- Single selection from predefined options using controls restricting selection and intentionally triggering event.
- User generated/input of content with predefined meaning (e.g.: using ascii text such as “:)” to be transformed to an icon representing a smiling face).

If we consider the output representations in the Facebook “Share” example and the text messaging emoticon example, the pattern can manifest in virtually any form. There are some simple and reusable patterns observed where verbatim representation or image substitution per associated option is provided. E.g.: smiling face emoticon entered as text and displayed either verbatim as “:)” or transformed as “☺”. Conversely there might be customized, proprietary implementations required such as colour coding of warning types using the green for safe, red for danger representations. Again these UI components map to existing (or extendable) components within the WebML *Hypertext Model* which use event listeners to process data from the *Structure Model*.

Within Facebook we see the representation mechanism as an embedded link to the shared content providing a preview as a teaser to get other people to “View”, “Like” or “Share” further. NOTE: The sharing of content by a user doesn’t necessarily mean that the following user whose behavior is triggered by the signal will choose the same, single-option response. The emoticon example represents the user entered text within a text editor and requires the user to only enter tags with predefined meanings (and understand the connotation of those representations).

Our output examples clearly show that there are business rule requirements to access at stigmergy data layer and to

present that data to the user with specific representation. The output mechanisms seen in our examples for this variety of stigmergy are:

- The (hyper-linking or verbatim) display of a UI component, whether contextually driven (e.g.: Display of the “Shared” content whether textual, image, html, multimedia, etc) or non-contextually (e.g.: “Terms and Conditions of Service” links).
- The input which was entered textually, but that requires recognition and interpretation by users who understand the signal representation (or the transformation where images are substituted).

We must remember that these abstractions are solely for our examples and is not intended to be an extensive list of all UI manifestations possible. An extendible WebML standard will enable future additions as required.

B. Quantitative, Marker-Based Stigmergy

Quantitative stigmergy is based on an accumulation of stimuli that do not differ qualitatively but that will reach a threshold and increase the probability of triggering a response. An example of this is demonstrated within the Facebook “Like” functionality which is a type of endorsement/acknowledgement system. A user who has made a contribution is hoping to provoke the response from their friends to “Like” it. The more people who “Like” the contribution increase the attention that the contribution receives. As the SUM total of people increases the probability of more people responding to the page increases, creating a positive feedback loop.

A more complex example of quantitative marker-based stigmergy is where there is more than just the single-option, signal contribution possible. In the online auction site eBay we have an example illustrated with the reputation feedback (see Figure 4.). There are a number of different criteria to answer, and each criteria has a 0 – 5 choice options where 0 represents a very negative feedback and a 5 represents a very positive feedback. As with the example in Section A if we generalize the mechanisms then we observe a single-choice selection from a group of possible options, however in this example there are multiple categories aggregated into a single contribution. The eBay example uses a non-standard UI component of a 5 star rating system. However any single selection UI controls (e.g.: drop-down list, radio button group, slider, etc) could be effectively used.

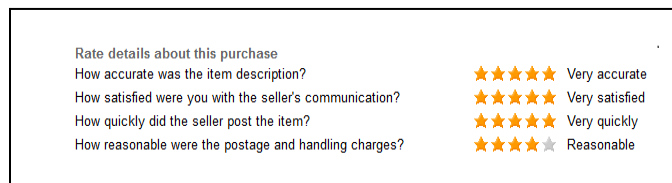


Figure 4. Example of an eBay feedback mechanism.

Just as with qualitative contributions, the WebML *Structural Model* will store the contribution triggered by events from the *Hypertext Model*. The UI has no other purpose than to facilitate the contribution of the stigmergic

signal. The generalized input mechanisms observed in the examples in this sub-section are:

- Single option intentionally triggered.
- Single selection from predefined options using controls restricting selection and intentionally triggering event.

When considering output the Facebook example the response of “Like” is signaled to all parties with access to the original contribution and displayed as a simple SUM total of the number of people who have intentionally chosen to respond with a “Like” to the same contribution, and re-ordering the original contribution higher within the news feed. All contributions are based on the same qualitative stimuli, with the environment providing an accumulation function as part of the output mechanism presented to the user. As with qualitative contributions, the UI components link the WebML *Hypertext Model* through event listeners to process and present data from the *Structural Model*.

The eBay reputation feedback signal appears to be a composite, aggregation function of all purchase history for the user via multiple criteria. We see that the input parameters of multiple-choice to multiple criteria are transformed through the aggregation function and result in an output value that is applied to the users’ current reputation status value (either positive or negative). We can see how the environment (which consists of other users) and the user provide the positive and negative feedbacks to the signal and how the WebML *Structural* and *Hypertext Models* can facilitate that. An interesting observation is that Wikipedia has adopted this quantitative, marker-based mechanism for soliciting user feedback on the quality of specific articles and pages [14].

We consider the generalized output mechanisms seen in our examples for this variety of stigmergy are:

- UI component display or modification representing the aggregate function of the contribution.

C. Sematectonic Stigmergy

Sematectonic stigmergy is defined as a modification of the environment as a by-product of actions being performed. These by-products are occurring inadvertently and unintentionally to the primary task being performed.

If we consider the “Seen By” example in Facebook we understand that the user is not making any intentional contribution by viewing a specific piece of broadcasted contribution, however their action of viewing the contributions has been recorded in the environment. The user has left a trail for others to see. In the Facebook example the user has only clicked on a hyper link in response to another users suggested interest point (see Figure 3.).

A more complex example of sematectonic stigmergy can be seen in the Amazon recommendation system (e.g.: “People who bought this also bought ...”), where product purchases made by a user are used as suggested items of interest to other users. In this example the input is virtually any potential sale item as the contextual input to the aggregation function. Irrespective of what manifestation the contextual input takes, we can observe that while the content type can take virtually any form the abstract mechanism

equates to hyperlinks and interaction flow points. These interaction flow points occur whether based on following hyperlinks or other action events (e.g.: selecting the “purchase now” button on Amazon).

In both examples we see unintentional user-triggered events which store that activity in the persistence layer. We consider the generalized input mechanisms observed in the sematectonic examples in this sub-section are:

- Unintentional trace logging via event-triggered interaction incidental to using primary functionality.

In the Facebook example the current state of “Seen By” is visible to all parties with access to the original contribution and displayed in two different ways: The first representation is a simple SUM total of the number of people who have intentionally chosen to view the contribution but who did *not* have the intention to let people know that. The second representation is the discrete list of the users who view the contribution and the date-time of viewing. This second representation can be seen in the top section of Figure 3. The instigating behavior in this example is the viewing or navigating to a user contribution with the unintentional by-product of leaving a trail.

The Amazon recommendation example shows how the contextual input can result in an aggregate function used to influence other site user purchasing behavior just as described in the qualitative, marker-based variety of stigmergy. Again we see that the output is represented through UI components linking the WebML *Hypertext Model* through event listeners triggering business rules to process and present data from the *Structural Model*.

A final and important example of output representation was demonstrated in a trial of Wikipedia article edit contributions [15]. In an attempt to display the verifiability of articles which had been edited (relatively) recently the article would have its page display a colour-tinted background. The background colour of the article page (or part thereof) would appear a pinkish-red colour to signal that article had previous un-validated modifications. This colour would slowly change through orange and yellow pastels until an undisclosed number of visitors to the article would presumably indicate that no ensuing modifications (indicating potential corrections) would imply that the original modification could be considered appropriate and correct. The importance of this example is that it relies on the user’s cultural understanding of colour association where warm colours such as red/orange imply caution and where green / white (default representing standard conformity) imply safe and reliable state. This illustrates the potential for new and insightful ways to provide implicit representation rather than using explicitly defined categories and numerical values.

We consider the generalized output mechanisms seen in our examples for this variety of stigmergy are:

- UI component display or modification representing the aggregate function of the contribution.

V. DISCUSSION

This paper has presented the varieties of stigmergy and briefly provided Web site examples of how the input and

output for each variety might be implemented. Our examples have explored simple implementations along with more sophisticated implementations. An initial analysis of User Interface (UI) components indicate that they might be independent to the variety of stigmergy they apply to. For example, we see that both qualitative and quantitative marker-based stigmergy can use a single selection type input element irrespective of the different types of output implementation. Similarly the output implementation of both quantitative marker-based stigmergy and sematectonic stigmergy appear compatible. If we are able to decouple the input/output implementations from the stigmergy varieties then we might enable a level of reuse of these mechanisms.

Our analysis has been focused on our UI observations which clearly correlate to the WebML *Hypertext Model* (both *Content* and *Navigation Models*). By definition stigmergy stipulates that communication occurs indirectly through environment mediated signs and indicates some form of data layer must exist.

Our model of stigmergy describes both positive feedback from users and negative feedback from the environment. The stigmergy data within the WebML *Structural Model* will therefore require data persistence accessible by the UI components as well as environment triggered events. This will be analyzed and included in the next phase of research when designing how stigmergy integrates with WebML. This metadata persistence is expected to also afford the administration of available qualitative marker-based options (e.g.: available text messaging emoticons and associated images/icons) and integrate into some form of tag disclosure through integrated online help. Our current work involves providing a model of how stigmergy as a design pattern integrates with WebML framework extending UI components and events within the *Hypertext Model* and how that maps to the *Structural Model* within WebML.

In the Facebook “Seen by” example we illustrate a single stigmergic mechanism resulting from a single input component (e.g.: The following of a link to a suggested content article) and how it can have two different output representations. The Facebook example showed that there can be a SUM aggregate function of the users who visited the suggested content, but also a chronological listing identifying the distinct database entry level data including the date/time of the event. This clearly highlights that when incorporating stigmergy into WebML we must accommodate multiple output visualizations (and different view locations where they are accessed) representing the single output state.

VI. CONCLUSION

What we hope we have clarified in this paper is how different stigmergic mechanisms might be implemented such that there is both an intentional and unintentional set of input elements. These input elements map to relevant output elements which include both visualization and representation (embodiment) of the contributions. This representation is determined by aggregation functions which are calculated using the business rules against persistent data for the specific stigmergic mechanism. These generate signals and

trails ranging from simple SUM aggregate functions to multi-criteria and multi-selection aggregate formulation into a final representation.

Stigmergy is not merely input and output mechanisms as presented within this paper; they are only part of the stigmergy phenomenon. For the mechanisms to work as intended a web site must be built analyzing the grand purpose of the site, and how stigmergic mechanisms can be employed to improve site coordination. The intra-site location of where the mechanisms are deployed (e.g.: specific users or role groups) ultimately depends on the development project sponsor. Similarly the WebML *Personalization Model* of a Web site might employ a fee paying structure where access to the mechanisms might be restricted. These issues are accommodated for within WebML and are ultimately expected to make the efficacy of introducing stigmergic mechanisms into Web application design significantly value-added.

REFERENCES

- [1] Theraulaz, G. and Bonabeau, E. A Brief History of Stigmergy. *Artificial Life*, 5, 2 (1999/04/01 1999), 97-116.
- [2] *Object Management Group IFML RFP*. City, 2012.
- [3] Grasse', P.-P. La reconstruction dun id et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp., La theorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs. *Insectes Sociaux*, 6, 1 1959), 41-80.
- [4] Dipple, A. *Stigmergy in Web 2.0: a Model for Site Dynamics*. ACM City, 2012.
- [5] Van Dyke Parunak, H. *A Survey of Environments and Mechanisms for Human-Human Stigmergy*. City, 2006.
- [6] den Besten, M., Gaio, L., Rossi, A. and Dalle, J.-M. *Using Metadata Signals to Support Stigmergy*. City, 2010.
- [7] Wilson, E. O. *Sociobiology: The New Synthesis*. Belknap Press of Harvard University Press, 2000.
- [8] Tummolini, L. and Castelfranchi, C. Trace signals: the meanings of stigmergy. In *Proceedings of the Proceedings of the 3rd international conference on Environments for multi-agent systems III* (Hakodate, Japan, 2007). Springer-Verlag, [insert City of Publication],[insert 2007 of Publication].
- [9] Tummolini, L., Castelfranchi, C., Ricci, A., Viroli, M. and Omicini, A. 'Exhibitionists' and 'Voyeurs' Do It Better: A Shared Environment for Flexible Coordination with Tacit Messages. City, 2005.
- [10] Brambilla, M. *Interaction Flow Modeling Language RFP: first cornerstone towards the standardization of WebML*. City, 2012.
- [11] *The Web Modeling Language*. City, 2009.
- [12] Brambilla, M. *Interaction Flow Modeling Language (IFML)*. City, 2012.
- [13] Corporation, M. *Emoticons*. City, 2012.
- [14] Rosenblatt, G. *Wikipedia Now Crowd-Sourcing Article Ratings*. City, 2011.
- [15] Harvey, M. *Wikipedia to highlight unreliable entries in colour*. City, 2009.

eRaUI: An Adaptive Web Interface for e-Research Tools

Farhi Marir, Sahithi Siva, Yango Jing
 Knowledge Management Research centre,
 School of Computing, London Metropolitan University, London N7 8DB
 f.marir@londonmet.ac.uk , s.siva@londonmet.ac.uk, y.jing@londonmet.ac.uk

Abstract— This paper presents the design and development of E-Research Adaptive User Interface (ERaUI) portable widget which uses case-based paradigm to learn web user profiles to enhance the usability and learnability of its host and adapt its content to that user profile. First, it uses click and mouse movement heat map techniques to track and record both user browsing behavior and the services provided by the host web site into text format which will be analysed by a text mining algorithm to form browsing patterns including changes in the web site content. Then, case-based reasoning paradigm and inductive learning algorithms analyse and index these browsing patterns into groups of web user profiles and store them into a case base memory from which most similar cases will be retrieved by ERaUI to identify and classify new users or discover new group of users. These web user profiles will be used by ERaUI to provide personalised services like content and collaborative search facilities useful for each web user profile, 'live help' box enabling the user to seek support and guidance from the admin feature of the host web site and display minimum content and a concise list of web site services that are most appropriate to individual web user profiles. This paper also discusses user evaluation of NaCTeM web site with and without ERaUI widget, and presents improvements identified through a series of usability tests.

Keywords— Web User Modelling; Adaptive user interface; Learnability; Usability; Heat map; Widget

I. INTRODUCTION

Adaptation refers to the notion of changing something to meet some specific requirements or purposes [1]. Adaptive systems are described as tools that generate new information about how to do the task better by analysing past experience and relating it to performance criteria set by humans [2]. It is also stated that an adaptive system adapts its behaviour to individual users based on information about them which can be either explicitly gathered or implicitly obtained during user-system interaction, and the adaptive system performs the adaptation using some form of learning, inference or decision making [3].

While it is recognised that adaptive interfaces improve usability, users' experiences and learnability and that they bring potential gains and a cost-benefit trade-off for usability, critics argue that autonomous user interface adaptation may disorient users and reduce its usability. Most critics relate these limitations with the unpredictable nature of the adaptive user interfaces and lack of accuracy [4, 5, and 6]. Learnability here refers to the system's support of the

user's efforts to learn how a system or an application has to be used

This paper looks at adaptive user interfaces from two viewpoints i.e. usability and learnability. The research considers two approaches in designing adaptive interfaces that address the above mentioned limitations: (i) those adaptive (static) user interfaces designed based on usability, learnability principles and methods, and user feedback are at the design level only and (ii) those adaptive (dynamic) user interfaces designed based on similar principles and methods but are different in that they keep on learning about users in dynamically adapting themselves to the needs of current and prospective users.

Case-based reasoning and inductive learning methods are used in profiling users and these user profiles are used in dynamically customising the web interface. The customisation comprises highlighting useful web page content and providing a list of selectable links to the content that is most appropriate to an individual user.

A number of methods have been used in the design of static adaptive user interfaces. For instance, analysis of the user interface using inspection methods such as heuristic evaluation, cognitive walkthroughs, GOMS (Goals, Operators, Methods, and Selection rules) analysis, and so on [7]. Also, empirical usability and learnability methods which involve user testing in a laboratory environment, could be either as formative or summative [8]. Formative studies are carried out during the product development process with the aim of fixing problems found during the product development process whereas summative studies are carried out after completion of the development and are used more as a basis for reflection and future work and base lining a product [9]. The activities undertaken include auto-recorded/measured user performed tasks, interviews, and analysis of experimental data.

However, dynamic adaptive user interfaces are enhanced with learning user profiles which could be either *Informative interfaces* that focus on filtering information the user finds interesting or useful, or *Generative interfaces* that generate some useful knowledge structures to support the user in their experience with the user interface [10].

In this paper, we present a dynamic e-Research adaptive user interface (ERaUI) widget. It is portable and is easy to add-on to any web site by adding a few lines of code to the header of the host web pages. It uses case-based reasoning paradigm and machine learning algorithms to learn from web user profiles that are generated whilst the users navigate through the website. ERaUI widget is an extension of a JISC

project commissioned to develop an adaptive interface to improve usability and learnability of NaCTeM e-Research tool [29]. The main focus of the eRaUI project is to customize the content of the host web site and highlight appropriate resources (e.g., menu options) according to the user's profile.

The paper is divided into five sections. Section II focuses on previous work pertaining to static and dynamic adaptive user interfaces. Section III describes usability and learnability methods used in the design and development of ERaUI widget. Section IV evaluates the usability and learnability features of ERaUI, and finally Section V presents the research findings and future work.

II. PREVIOUS WORK ON ADAPATIVE INTERFACES

Several recent projects have been sponsored by Joint Information Systems Committee (JISC) to develop usable and learnable user interfaces [30]. Most of these could be considered as static adaptive interfaces as usability and learnability are dealt with at the design level only. ALUIAR project [30] ranks the results of a user group feedback through interviews and "walk throughs" to improve the usability and learnability of the user interface of Synote [30], the open source web based video and audio annotation tool.

Rave in Context project [30] developed usable, accessible, learnable and adaptable W3C widget templates and widgets for MyExperiment, Simal and OpenDOAR web sites.

UseD project [30] developed a new user interface to improve the usability of Digimap data downloader, making it easier for a range of subscribers to use. It was designed by creating a number of stereotypical user Personas, which are based on the actual Digimap user requirements, their expectations of the service and their knowledge of spatial data.

ReScript Usability/Learnability Enhancement project [30] improved usability and learnability of the user interface of ReScript, a prototype of digital editing and research environment originally developed to support collaborative work on historical texts. Feedback compiled from a series of online surveys and interviews with editors and researchers was used to make changes to the ReScript interface to meet the needs of a variety of researchers working with very different texts, and with differing levels of expertise.

The Word Tree Corpuce Interface [30] had the goal of providing an alternative interactive user interface to traditional text-analytical tools like KeyWords In Context (KWICs). The website produced allows users to generate word trees for individual terms, starting from the searched term at the leftmost edge with branches of proceeding words extending to the right. To develop a usable and learnable user interface, the Word Tree Corpus team undertook a combination of quantitative and qualitative studies through surveys and video interviews of stakeholders. They also used Google analytics to analyse user behaviour and to further improve their new user interface to reflect the users' needs.

The literature review revealed a significant number of studies on dynamic adaptive interfaces. Pazzani and Billsus [11] reported the development of SySKILL & WEBERT

adaptive web user interface, which recommends web pages on a given topic that the user is likely to find interesting. The user marks suggested pages as desirable and undesirable and the system uses naive Bayesian classifier [32] for this task, and demonstrates that it can incrementally learn profiles from user feedback on how interesting web sites are. Furthermore, the Bayesian classifier may easily be extended to revise user provided profiles.

Another web user interface, NEWSWEEDER system [12], recommends stories to the user using each word in the story to predict whether the user finds the story interesting or not.

P-TIMS [13] is a commercial financial management system and was revised to add an adaptive and adaptable interface using a simple user model and rule set. As the user spends more time using the system and uses more complex functions, the system reveals a more extensive interface. The user model is explicitly exposed by providing a "preferences" dialog box, which the user can adjust at any time.

AVANTI [14] is a hypermedia information system about a metropolitan area and uses an initial interview to create the initial primary assumptions (i.e., user profile), draws inferences to generate additional assumptions, and uses stereotypes for certain subgroups of users (e.g., tourists, blind users). It then customises the web pages presented to the user accordingly.

Interbook [15] is an adaptive system which derives much of its data for the user model from the user interface component, and which can track user actions and report them in detail to the application user model. Interbook addresses these problems by tracking what the users have seen, rather than what they have done, and using that to infer what the users know.

Most recently, Lee et al. [16] developed a user interface prototype for the Android smartphone, which recommends a number of applications to best match the user's context based on five variables; time, location, weather, emotion, and activities. The developed system derives the best three recommended applications based on a probabilistic learning and inference algorithm named "Spatiotemporal Structure Learning" [33], which extends Naive Bayesian Classifier [32].

It can be noticed that most of the above dynamic adaptive interfaces require explicit input from the user to be able to model web user profiles and adapt web user interfaces to satisfy individual user profiles. ERaUI which is also a dynamic adaptive web user interface implements an intelligent and portable widget, which provides both explicit and implicit input from the user to build web user profiles. In the implicit input, ERaUI uses automatic tracking mechanisms to track and record the user behaviour whilst browsing the web site on which ERaUI widget is hosted. It also uses inductive learning algorithms to identify or discover new web user profiles and enhances the host web site's usability and learnability capabilities. Furthermore, it detects changes in hosted web site content in terms of functionalities and resources while tracking the user

browsing and reflecting that in the adaptation of the web user interface and also the services presented to the user.

III. DEVELOPMENT PROCESS OF ERAUI ADAPTIVE USER INTERFACE

A. ERaUI Web User Interface Design

In ERaUI project, adaptive web user interface is seen as an interactive software system that improves its usability and learnability in its interaction with a user based on partial experience with that user. The formal definition of usability by the International Standards for HCI and Usability [31] is that usability is concerned with the “effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.” Effectiveness means how accurately and completely users can achieve their goal. Efficiency means the effort required to achieve a goal. Satisfaction means the comfort and acceptability of using the system to achieve the goal. These definitions are in agreement with [7] who also stressed the importance of learnability as an important aspect of usability.

According to the Usability First glossary and reference [17] learnability is a measure of the degree to which a user interface design can be learned quickly and effectively. Learning time is the typical measure. User interfaces are usually easier to learn when they are familiar to the user and/or designed to be easy to use based on core psychological properties. Through literature reviews the learnability of a user interface design can be broken down into five types: Familiarity, Consistency, Generalizability, Predictability, and Simplicity. It is also stressed that although learnability could be part of usability, little is shown that an increase of ease of use (usability) can be realised without actually improving the user’s mental model (learnability) of adaptive systems.

Several methods have been used in the design of usable and learnable user interfaces. Analytical usability studies involve analysis of the system using inspection methods such as heuristic evaluation, cognitive walkthroughs, GOMS analysis, and so on [7]. Empirical usability involves people using test methods and traditionally conducted as either formative or summative studies [8]. Formative studies are carried out during the product development process with the aim of fixing problems found during the product development process whereas summative studies are carried out after completion of the development and are used more as a basis of deciding lessons learned and base lining a product [7]. The activities undertaken generally include recorded/measured usability evaluation, interviews, and laboratory-based experiments and so on. Nielsen and Phillips [18] research on “estimating the relative usability of two interfaces” concluded that the most reliable way of determining the relative performance was through the use of empirical usability studies rather than analytical usability studies although those empirical studies are more expensive to perform.

ERaUI was designed based on a methodology commonly found in usability studies and recommended by [7]. It uses personas and scenarios as a way of

contextualising the what, where, how, when and why of the use of an application so that in essence it shows to the targeted users. Power of personas is that all stake holders involved in the design process are much more likely to engage with other people, real or fictional, than they are with statistical information [19].

We conducted an experiment within the university research community on the impact of usability methodology of personas and scenarios that led to the design of the ERaUI interface as an intelligent and portable widget that we deployed on the left hand side of NaCTeM web interface as shown in Figure 1.

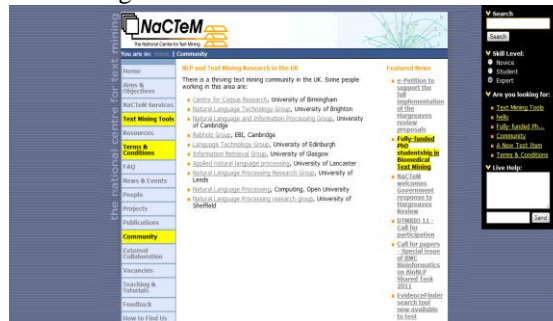


Figure 1. ERaUI Widget hosted by NaCTeM Web Site.

As shown in Figure 1, ERaUI can be hosted on any web site by adding few lines of code to the host website header files. The widget is comprised of a free-text / autocomplete search box which offers enhanced search capabilities for host websites. The learners can also specify explicitly their profile (e.g., skill level) for NaCTeM host web site, and can additionally choose from a variety of links which are recommended according to their user level. At the bottom of the widget, a ‘live help’ for users browsing the host website to communicate with the website administrator in real-time.

B. ERaUI Learning Web User Profile

ERaUI project uses case-based reasoning (CBR) paradigm [20, 21] to learn about the user’s profile and in developing the ERaUI adaptive web user interface. CBR uses machine algorithms to solve new problems by adapting solutions of previous similar problems following the cycle Retrieve, Reuse, Revise and Retain, as shown in Figure 2.

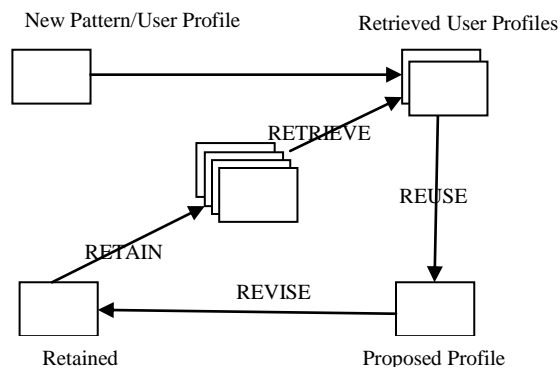


Figure 2. CBR Cycle [21]

For example, CBR systems CLAVIER [22], which are developed as advisory systems to recommend loads and layout for aircraft parts to be cured in an autoclave, and CBRefurb [23], which retrieves previous similar refurbished buildings to estimate the cost of new refurbishment. Both use inductive learning algorithms within CBR to index and retrieve and revise/adapt most similar past cases from their case libraries.

1) ERaUI Case Representation

ERaUI case is represented by two data structures which will be initially set by the administrator of the host web site: a user categories structure in which the admin can set initial known categories of web user profiles like for instance Novice, Student and Expert category of users of NaCTeM e-Research tool. A second keyword structure of the host web site and links in which the admin can set keywords, web page links and external web sites and assign them to the categories already set. Once set, these two structures are automatically managed by ERaUI learning system. That means ERaUI case-based learning system can automatically make changes to the content of either structure by adding or removing categories of users in the category structures or adding or removing keywords in the keyword structures to reflect the changes in both the user behaviour and the changes in the content or functionalities in the host web site itself.

2) ERaUI Case Indexing and Populating

ERaUI use click / mouse movement heat maps techniques to tracks the user interaction with the host website as shown in Figure 3.



Figure 3. ERaUI Click & Mouse Movement Heat Map.

The results of click heat maps tracking are translated by ERaUI into text format and stored as a personal user record in ERaUI database as shown in Figure 4.

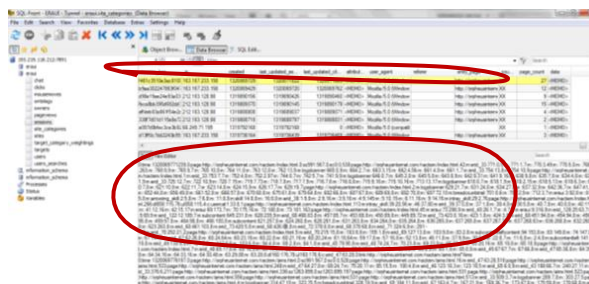


Figure 4. User Records and browsing patterns

The ERaUI text mining tool analyses the collected user text records to derive user patterns (clicked key words, functions, webs site links, etc.) which will be represented as new case (vector) of the current user updating the case base. This new case will be further analysed by the ERaUI case-based inductive algorithm to determine whether it is a new or an existing web user profile and therefore updating if necessary the content of the category structure

3) ERaUI Machine Learning Algorithm

Given a description of a problem, a retrieval algorithm, using the indices in the ERaUI case-memory, should retrieve the most similar cases to the current problem or situation. The retrieval algorithm relies on the indices and the organisation of the memory to direct the search to potentially useful cases.

The issue of choosing the best matching case has been addressed by research into analogy [24]. This approach involves using heuristics to constrain and direct the search.

Case-based reasoning will be ready for large scale problems only when retrieval algorithms are efficient at handling thousands of cases. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case.

In order to retrieve or generate suggestions on the web user profile which are relevant to the given browsing pattern we had to consider amongst well-known methods for case retrieval like nearest neighbour, induction, knowledge guided induction and template retrieval. We found that the most effective algorithm to match users with results according for instance to their skill level in NaCTeM was Nearest Neighbour Algorithm (NNA) because it is more effective when the case base is not huge.

NNA involves the assessment of similarity between stored cases and the new input case, based on matching a weighted sum of features. A typical algorithm for calculating nearest neighbour matching is the one reported in [25] where w is the importance weighting of a feature (key word, web link, etc...), sim is the similarity function, and fI and fR are the values for feature i in the input and retrieved cases respectively.

$$\frac{\sum_{i=1}^n w_i \times sim(f_i^T, f_i^R)}{\sum_{i=1}^n w_i}$$

Figure 5. Nearest Neighbour Algorithm

Furthermore, Induction algorithms ID3 [26], which determine the dominant, are used to discriminate cases based on these features; they generate a decision tree type structure to organise and categorise ERaUI cases of web user profiles in memory.

IV. ERAUI USABILITY AND LEARNABILITY FEATURES

In addition to its design using an iterative design methodology that focuses on usability, learnability, participatory design suitable for service-based implementations, ERaUI widget uses case-based reasoning and inductive learning methods to learn about both the user profile and the host web user interfaces in deducing and displaying needful information for the user.

ERaUI has addressed most of the common issues reported in the literature that arise in developing adaptive or advisory interfaces including information filtering, supporting the user in his/her experience with the web site and also visual changes to the user interface itself.

1) ERaUI Filtering Information

ERaUI implements advanced filtering algorithm(s) for extracting digital content of the host web site like NaCTeM to match user needs. Content and collaborative based filtering methods [27, 23] have been used as the basis for selection and learning about the content of NaCTeM web site. Content methods suggest topics similar to the ones a user group with similar profile has liked in the past as shown in Figure 6:

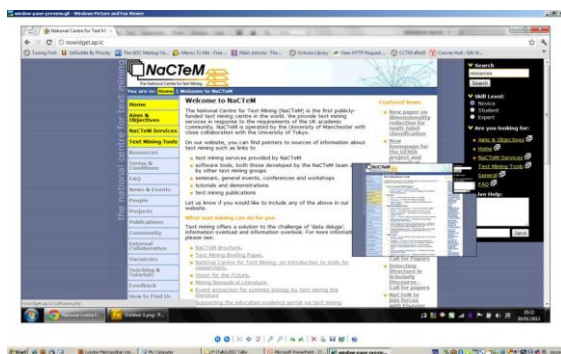


Figure 6. Results of content filtering method

However, as shown in Figure 7, collaborative filtering methods suggest items outside the user's normal area that the user will still find interesting, as the basis for selection and learning.

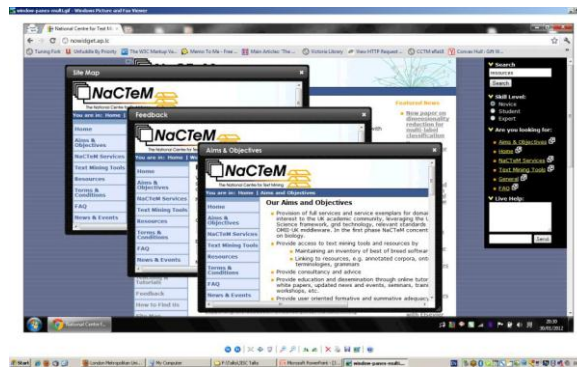


Figure 7. Panes of similar pages from collaborative filtering

2) ERaUI Web User Interface Adaptation

Also, based on the identified web user profile, ERaUI recommends choices on some aspects of the research processes as desirable or undesirable, rating them on a scale and giving some similar form of evaluation to help the user in his/her selection of retrieved information. An instance of ERaUI user interface adapting itself to a category of users is shown in Figure 8. This screenshot suggests to the user currently browsing the host web site a few links in yellow colour, normally used by users with similar web user profiles.

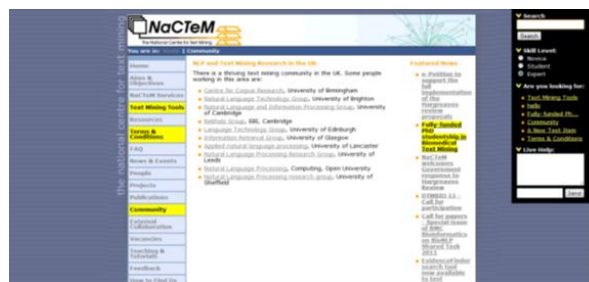


Figure 8. Suggesting useful links for a category of users

3) ERaUI Live User Support

As shown in Figure 9, ERaUI interface also provides live communication facilities through 'live help' box enabling the user to seek support and guidance from the NaCTeM host web site administrator.



Figure 9. ERaUI live help box

4) ERaUI Widget Evaluation

In a recent study, the investigation undertaken in Lazar et al. [28] found that users lose up to 40% of their time due to “frustrating experiences” with the application interfaces, with one of the most common causes of these frustrations being missing, hard to find, and unusable features of the software application.

In order to assess the frustration of the users and evaluate the ERaUI widget based on some usability and learnability criteria, we invited around fifty researchers and students with different skills to run NaCTeM web user interface with and without ERaUI widget. The users are divided into two groups. Each user has been given identical tasks to complete in a given time. The measure we used for evaluating ERaUI widget usability and learnability features were based on completion of tasks on time, accuracy in providing information to the user and also predictability of user interface.

They were asked to complete a series of simple tasks using the widget. Different researchers were asked to complete a series of tasks without the widget. We compared the results of the two groups to get an idea of how effective ERaUI is at enhancing the experience of users. We set the following tasks:

- Write down the postcode of the National Centre for Text Mining.
- Write down the name of one member of Core Staff working at NaCTeM
- Write down the name of the only listed Visiting Researcher at NaCTeM.
- Write down the closing date for applying for PhD studentship advertised on NaCTeM website (3 year studentship based at the School of Computer Science, University of Manchester).
- Post some simple feedback to NaCTeM. Write down the keyword which appears when you do this.

The results of this evaluation have shown that all users who used NaCTeM with ERaUI widget completed all the tasks on time compared to 80% when using NaCTeM without ERaUI widget.

Another evaluation carried out in this research is the accuracy of the actions predicted by the ERaUI widgets. For this, we conducted a second evaluation test with 45 researchers and students to assess predictability and accuracy of ERaUI through the following predictability and accuracy tests:

- Set your user profile (skills) in the ERaUI Interface for NaCTeM and check if the web links list in the widget and those highlighted in yellow in NaCTeM resources predict and match the profile you selected
- Browse through NaCTeM web site for five minutes without setting your user profile and check if ERaUI is predicting and highlighting in yellow colour a number of web page links and/or functionalities

within NaCTeM web content which are useful to you

- Make a search using ERaUI Interface and another search using original NaCTeM and compare which one is more accurate and useful for the user,
- Check the accuracy of results by comparing the results of the same functions of the original NaCTeM (like search and browse through some of NaCTeM functionalities) with the same functions and searches provided by ERaUI

Results on predictability and accuracy test have shown that 75% reported that the ERaUI widget predicted what they are looking for and 85% reported that ERaUI provided accurate information they asked for. This is compared to around 15% predictability and 75% accuracy when using NaCTeM interface without ERaUI respectively.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an initial design and development of an adaptive web user interface in the form of an intelligent and portable widget named ERaUI that could be hosted on top of any web site user interface to enhance its usability and learnability features.

The research proposed two approaches to the design and development of adaptive user interfaces. In both the cases the adaptive features are set either by using usability and learnability methods and elicitation of the users’ needs at the design level only or by requiring explicit input from the user whilst browsing through the web site interface to be able to build user profiles and use it to adapt the web user interface.

ERaUI uses usability and learnability methods like persona and scenarios customising web user interface with or without explicit input from the user. This is achieved by learning user’s profile through the following methods: (i) observe mouse click or mouse movement to track and collect user browsing behaviour, (ii) text mining to analyse the tracks and identify user browsing patterns/features, and (iii) case-based reasoning paradigm to index and store user profiles in case-based memory and nearest neighbour and ID3 inductive learning algorithm to retrieve or discover and also to classify web user profiles in the case-base memory. Each user profile will then be used to customise suitable web content.

ERaUI has used user profiles to implement some of the usability and learnability features like advanced search and filtering algorithms to provide useful information to user, adapting the content of the host web site to show only appropriate resources and functional features that meet the user profile and also support the user through communication tools.

One of the key approaches to assessing usability and learnability of a user interface will be via real users’ evaluation and testing of the interface. For this, we conducted an initial evaluation and testing of ERaUI by inviting around fifty researchers and students to experiment with NaCTeM web site, with and without ERaUI widget. Although these

evaluation tests are not comprehensive in that they do not include comprehensive usability and learnability measures, they have shown that there are significant improvements of the users' experience when using ERAUI widget based on testing usability and learnability measures like completing tasks on time, accuracy of returned information and prediction of users' expectations. Further work will be undertaken to experiment and improve the ERAUI widget.

ACKNOWLEDGEMENT

The authors would like to thank JISC for their funding of this research and the JISC main committee members and the reviewers for supporting the work. The authors acknowledge the work of Eamonn Ramsay, the Software Developer of this project.

REFERENCES

- [1] Victor M.G.B, "Personalization in Adaptive E-Learning Systems, A Service-Oriented Solution Approach for Multi-Purpose user Modelling Systems," Faculty of Computer Science, Graz University of Technology, Graz, Austria, May 2007
- [2] Werbos, P.J. , "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research". IEEE Transactions on Systems, Man, and Cybernetics, 1987, vol 17(1), pp 7-20.
- [3] Anthony J. and Krzysztof Z. G., "The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications" In Systems That Adapt to Their Users J. A. Jacko (Ed.) (3rd ed.), 2012 Boca Raton, FL: CRC Press.
- [4] Tim F. P., Jasper L. and Mark N. "Usability trade-offs for adaptive user interfaces: ease of use and learnability" Proceedings of the 9th International Conference on Intelligent User Interface IUI'04, Jan. 13–16, 2004, Madeira, Funchal, Portugal. ACM 1-58113-815-6/04/0001.
- [5] Krzysztof Z. Gajos, Katherine E., Desney S. Tan, Mary C., and Daniel S.W. "Predictability and accuracy in adaptive user interfaces." In CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pages 1271-1274, New York, NY, USA, 2008. ACM.
- [6] Talia L., Joachim M., "Benefits and costs of adaptive user interfaces" Int. J. Human-Computer Studies 68, pp. 508–524, 2010 Elsevier.
- [7] Nielsen, J. "Usability Engineering" Book published by Morgan Kaufman, San Francisco 1993, ISBN 0-12518406-9 1993
- [8] Holzinger A., "Application of Rapid Prototyping to the User Interface Development for a Virtual Medical Campus". IEEE Software, Volume 21, Issue 1, pp92-99, 2004 (ISSN 0740-7459) [Software Engineering, Rapid Design, Graphical User Interface (GUI)]
- [9] Barnum C., "Preparing for usability testing" In Usability testing essentials ready, set—test!" Burlington, 2010, pp 188-229, MA. Morgan Kaufmann Publishers.
- [10] Langley, P. "Machine learning for adaptive user interfaces". Proceedings of the 21st German Annual Conference on Artificial Intelligence (pp. 53-62), 1997, Freiburg, Germany: Springer.
- [11] Pazzani M., & Billsus D., "Learning and Revising User Profiles: The identification of the interesting web sites", Machine Learning, 27, 313 331, 1997.
- [12] Lang, K., "NewsWeeder: Learning to filter news". Proceeding of the twelfth International Conference on Machine Learning, pp331-339. Lake Tahoe, CA, Morgan Kaufmann (1995)
- [13] Strachan, L., Anderson, J., Sneesby, M., and Evans, M. "Pragmatic User Modelling in a Commercial Software System." In Anthony Jameson, Cécile Paris, and Carlo Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97. Vienna, New York: Springer Wien New York, 1997. Available from the World Wide Web: <http://www.um.org>
- [14] Fink, J., Kobsa, A., and Nill, A. "User-oriented Adaptivity and Adaptability in the AVANTI Project." Conference Designing for the Web: Empirical Studies. Microsoft Usability Group, Redmond, WA, 1996. Available from the World Wide Web: <http://fit.gmd.de/hci/projects/avanti/publications/ms96.html>. (last accessed on 2/1/13)
- [15] Fink, J., Alfred K., and Andreas N., "Adaptable and Adaptive Information Access for All Users, Including the Disabled and the Elderly." In Anthony Jameson, Cécile Paris, and Carlo Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97. Vienna, New York: Springer Wien New York, 1997, pp. 171-173.
- [16] Hosub L., Young S.C and Yeo-Jin K., "An adaptive user interface based on spatiotemporal structure learning" Communications Magazine, IEEE, June 2011, vol. 49 , Issue 6, pp. 118 – 124.
- [17] Louis T. "Usability 101: Learnability" published online June 2003 available in <http://www.tnl.net/blog/2003/06/17/usability-101-learnability/> (last accessed on 2/1/13)
- [18] Nielsen, J., and Phillips, V. L. "Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared" Proc. ACM INTERCHI'93 Conf. (Amsterdam, the Netherlands, 24-29 April), 214-221, 1993.
- [19] Grudin, J., 'Why personas work – the psychological evidence.' In: Pruitt, J., Adlin, T. (Eds.), The Persona Lifecycle, Keeping People in Mind Throughout Product Design, Elsevier, pp. 642–663, 2006.
- [20] Marir, F. and Watson, I.D. 'A Categorised Bibliography of Case-Based Reasoning', Knowledge Engineering Review Journal, 1994, Vol. 9:4, 355-381, 1994
- [21] Watson, I.D and Marir, F. (1994) 'Case-Based Reasoning: An Overview', Knowledge Engineering Review Journal, 1994, Vol. 9.4, pp. 32 7-354.
- [22] Hinkle, D. and Toomey, C.N. (1994). CLAVIER: Applied case-based Reasoning to composite part fabrication. Proceeding of the sixth Innovative Applications of Artificial Intelligence Conference, 1994, pp 55-62, WA: AAAI Press
- [23] Marir F. & Watson I.D. (1995) "Representing and indexing building refurbishment cases for multiple retrieval of adaptable pieces of cases". 1995, In, Manuela Voloso and Agnar Aamodt (Ed.). In, Lecture Notes in Artificial Intelligence: Case- Based Reasoning Research and Development. Berlin Heiderberg: Springer Verlag, LNAI 1010, pp. 55-66.
- [24] Falkeneheimer, B., Forbus, K.D. and Gentner, D. (1986). The structure mapping engine. In, Proceeding of the Sixth National Conference on Artificial Intelligence, Philadelphia, PA, US.
- [25] Kolodner, J.L., (1993). Case-Based Reasoning. Morgan Kaufmann.
- [26] Quinlan, J.R. (1979). Induction over large databases. Rep. No. HPP-79-14, Heuristic Programming Project, Computer Science Dept., Stanford University, US. [26, Quinlan, 79]

- [27] Haouam, K. and Marir, F. (2006) ‘A Dynamic Weight Assignment Approach For Index Terms and Rhetorical Relations”. *Journal of Computer Science* 2(3): 261-268.
- [28] Lazar, J., Jones, A. and Shneiderman, B., (2006). Workplace user frustration with computers: An exploratory investigation of the causes and severity. *Behaviour and Info. Technology*. 25(3):239-251.
- [29] NaCTeM, (2013). The National Centre for Text Mining (NaCTeM) Homepage, <http://www.nactem.ac.uk/> (last accessed on 2/1/13)
- [30] JISC (2013) <http://www.jisc.ac.uk/whatwedo/programmes/researchinfrastructure.aspx> (last accessed on 2/1/13)
- [31] International Standards for HCI and Usability (2013) http://www.usabilitynet.org/tools/r_international.htm (last accessed on 2/1/13)
- [32] Bayesian Classifier <http://www.autonlab.org/tutorials/naive.html> (last accessed on 2/1/13)
- [33] Walter F.B. and Terry C. *Learning Spatio-Temporal Relational Structures, Applied Artificial Intelligence*, 15:707-722, 2001, Edited by Taylor & Francis.

Using Semantic Indexing to Improve Searching Performance in Web Archives

Arshad Khan*, David Martin^ψ, and Thanassis Tiropanis^ψ

*National Centre for Research Methods (NCRM), University of Southampton
Southampton, UK

Email: a.khan@soton.ac.uk

^ψ Geography and Environment, University of Southampton, Southampton, UK

Email: D.J.Martin@soton.ac.uk

^ψ School of Electronics & Computer Sciences (ECS), University of Southampton, Southampton UK

Email: tt2@ecs.soton.ac.uk

Abstract—The sheer volume of electronic documents being published on the Web can be overwhelming for users if the searching aspect is not properly addressed. This problem is particularly acute inside archives and repositories containing large collections of web resources or, more precisely, web pages and other web objects. Using the existing search capabilities in web archives, results can be compromised because of the size of data, content heterogeneity and changes in scientific terminologies and meanings. During the course of this research, we will explore whether semantic web technologies, particularly ontology-based annotation and retrieval, could improve precision in search results in multi-disciplinary web archives.

Keywords—Semantic indexing; archive searching; multi-disciplinary web archives; semantic searching; linked data

I. INTRODUCTION

Information scientists have long been struggling to find a system that can help them organize disparate collections of web archives (or, in a general sense, web resource archives) so that users can have access to complete and coherent collections [1] in a much more meaningful way. Although both web archives and web repositories are sometimes used to refer to archives of web resources, the term web archive will be used through the remainder of this paper.

The prevalent access in web archives is based on the search over automatically extracted metadata from web documents [2] which have to be indexed for keyword searching. Providing broader access (unlike the current keyword search) to the collection of those web archives via an ontological framework structure could not only increase the utilization of these hard-earned resources but also make them more research-oriented, structured and flexible to cope with the changing needs of users [1], especially the research community.

The problem with conventional text based searching in web archives is that it only categorises the content in the archive on the basis of instance occurrence and query weighting with no attention paid to context, relevance, terminological coherence or relationship between web

pages, nor does it relate the web pages of the repository to external sources on the web of data.

Web resources archives contain complex collections of research materials in online domains that can serve distinct communities, for example social scientists or historians who desire to search information based on contextual and provenance information [3]. We understand that semantic web technologies will be of tremendous help in identifying and integrating such heterogeneous documents inside web archives and enabling context and meaning-based search in them by exploiting existing vocabularies and domain-specific ontologies.

To further investigate the above issues, a thorough review of the most relevant research was carried out under the umbrella of web archiving, searching strategies in web archives and the application of next generation semantic web technologies. Particular attention has been given to linked data to see if users' searching experience could be improved by locating more precise and relevant information in web archives and repositories. In almost all cases, past research has focused on individual components of a typical web page in a particular domain such as semantically annotating and linking research datasets in biology, e.g., [4] multimedia objects in a newspaper archive, or [5] research publications (usually in proprietary formats) in scientific publication archives, to generate semantic metadata and generate improved search results.

This paper will describe the application and extension of keyword searching in web archives of multi-disciplinary research data by annotating web documents using more specialized and archival domain-specific ontologies and subsequently searching over the annotated metadata and instances of data i.e. Knowledge Base (KB). We will also take into account the existing and widely used classification systems e.g., thesaurus, typologies and taxonomies of social science to see if they can be synthesized into building an annotation ontology for creating annotation metadata.

II. CORE AREA OF RESEARCH

The development of a single on-line resource built by subject experts (e.g., a site presenting the methods and results from a research project in a social science discipline) can be time-consuming and expensive (in part, because they are not IT experts) and the full value of the resource only comes into play close to the point at which project funding ends. Those online resources not only provide a valuable personal development resource for researchers wishing to benefit from the training or data provided, but also provide an important repository of social, economic, historical and human knowledge. They are frequently created as the end-product of particular projects by academics and researchers associated with particular disciplines.

Following the completion of a project, the materials contained in some of these resources are considered for archival purposes so that they remain available despite the end of funding and dispersal of teams. This is an increasingly familiar situation as funders and institutions seek to develop repositories to increase the impact and availability of their work. Such archives or repositories of web resources enable users to search for information in the collection using basic keyword searching which proves to be ineffective simply because they retrieve web pages, datasets, and research papers merely on the basis of incidental mentions of a term in users' queries. Such a searching strategy misses the context, relationship, historical details and other attributes of a particular resource which would have enabled users to extend their exploration to other related records in the archive. One such web resources repository is the ReStore repository [6] which will be used as a test-bed for the experimentation and search performance evaluation described here.

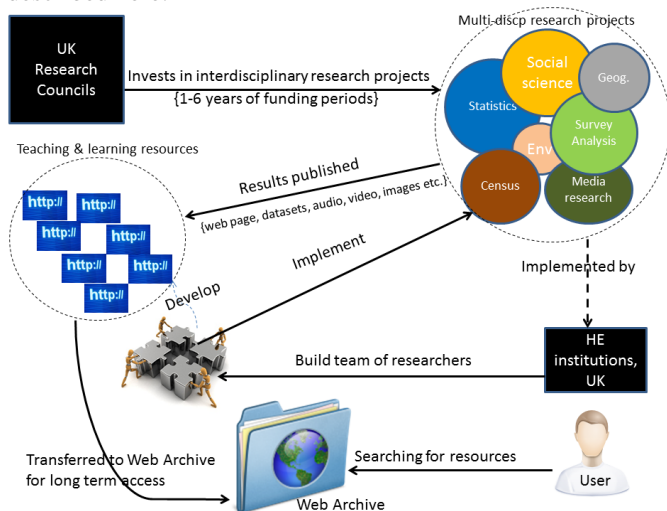


Figure 1: An overview of web resource, creation, development and its archival into a web resource archive, e.g., ReStore repository

Figure 1 shows a typical web resource development and archival process involving funding of a research project, development of a web resource by the project team based in higher education institutions, and the archival of the resource in a web archive such as ReStore repository.

Figure 1 also presents an opportunity to think about the problem which may arise at the time of searching the respective archive having different online materials created as part of various social science research projects.

III. THE PROBLEM

Some of the frequently observed problems stemming from the current searching methodologies include, but are not limited to:

- The typical keyword-search, usually based on keyword-matching or phrase-matching algorithms, hardly takes into account context, relevance and relationship between web documents, data, people, organizations, projects and other artefacts.
- Too many web pages make it almost impossible for users to go through and look for relevant information as a result of searching.
- The outputs of the same research project may be spread out over several repository/archive services, e.g., published research papers uploaded to an institutional repository, core research data uploaded to another data management service such as Economic & Social Data Service (ESDS) and videos/audios uploaded to video sharing services such as YouTube, Google videos etc.

Essentially, the content of valuable online resource collections becomes disintegrated and spread across different online archives. These are in most cases findable by standard search engines but with no classification hierarchy or associated relationships that links to their siblings (parts of the same web resource) or external, related objects that could be made available to users. An example would be research on a similar topic undertaken either in the past or currently. Similarly, a research funding organization may be funding another project in the same field or related workshops may have been offered as part of different projects. These types of relationship are well understood by the academic researcher but not readily amenable to discovery via existing searchable metadata.

IV. EXPERIMENTAL SETUP

The overall purpose of this research is to establish whether accuracy and relevance in search results are improved both at system level and in terms of user experience when searching for information in a repository of web resources having a meaningful semantic index.

We have outlined the current state of searching in website archives which largely employ only keyword-based searching techniques to retrieve information. We

are now able to formulate the following research questions which will act as a driving force for this particular research.

1. Can keyword-based searching be applied to semantically indexed metadata created by the semantic annotation process to enhance information retrieval in web resource archives?
2. Can domain-specific terms in an Ontology and their expressions in KB improve precision and recall in search results when added to the keywords search?
3. Does semantic indexing of ontology and KB along with inferencing cater to the heterogeneous types of data contained in web archives in terms of relationships, relevance?
4. Does the scale of searching over large multi-disciplinary web resources in web archives effect system performance in terms of the number of queries submitted at a particular point of time and the time it takes for the system to retrieve relevant information from multiple data sources?

It is important to remember that in all of the above, users of such archives have to be kept in the forefront of the research as they will be the ultimate beneficiaries of any new systems based on such research in the future.

V. PROPOSED ONTOLOGY-BASED SEARCHING SYSTEM

Figure 2 shows the building blocks of the experimentation environment which will enable us to conduct various experiments and evaluate search results performance in different scenarios.

The results have to be evaluated while keeping in view traditional RDBMS keyword-based searching techniques which are usually applied in many web resources archives including the ReStore repository.

Figure 2 shows basic components of the experimental setup including KIM (Knowledge & Information Management) platform with inherent support of GATE (General Architecture for Text Engineering), Open source full-text search engine Lucene, ReStore repository domain ontology fully mapped to KIM ontology and the resulting semantic metadata or RDF triples store with searching interface on top of it. It also follows that by incorporating the existing semantic web technologies and linked data concepts, we can enhance searching performance in web archives. A quantitative approach towards measuring performance has been adopted to evaluate accuracy of search results.

Sesame RDF repository will be an OWLIM semantic repository for managing RDF triples in addition to inferencing and SPARQL query evaluation during the course of various experimentation.

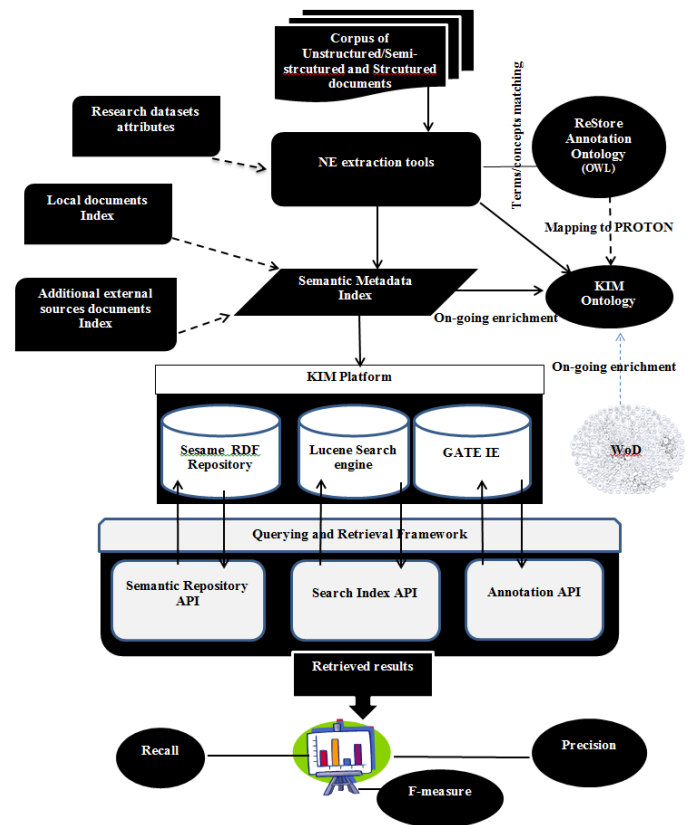


Figure 2: Proposed Semantic indexing based searching architecture built on KIM platform incorporating RDF(s) Sesame semantic repository and distributed (shared) ontological framework.

VI. THE CHALLENGE OF SCALE

Annotation of web documents containing different types of data on a large scale is certainly an issue, but the presence of a well-modelled ontology representing the domain of interest could address that issue to a large extent. Also the trend of submitting lengthy search queries [7] is making it increasingly difficult for keyword-based search engines to perform well as users are becoming more and more interested in the context precision, related information and the source and provenance of what is being searched.

The challenge is therefore of improving performance of the keyword-based search using semantics without losing search scalability [8] and users' interaction experience, to which they have become accustomed as the web has become an essential component of their lives. However it still remains to be seen whether the different environments in which web resources have originally been created (e.g., web served, harvested or manually collected), archived and indexed could influence the accuracy of results and performance of systems.

During our experimental setup we will analyse the above from different perspectives in order to support our research and solidify the basis of our findings. We have

already started the actual annotation work along with indexing and retrieval of information. We have already made progress in formulating the research framework and will proceed to undertake it in order to be able to evaluate performance in different scenarios, e.g., scale, heterogeneity of documents and retrieval of related information.

VII. ONTOLOGY ARCHITECTURE OF ReSTORE REPOSITORY

In Figure 3, we have put together very basic terms and concepts (classes, properties) in a container with a view to mapping and extending them to other ontologies and storing the resulting annotation metadata (instances of classes or KB together in the form of integrated ontologies. The KB and mapped network of ontologies are assumed to be searched at a time to produce relevant and accurate results with a view to evolving it iteratively as new terms are added. GATE will process structured/semi-structured content in documents and OWLIM represents RDF triple store containing inference, querying engine for information retrieval.

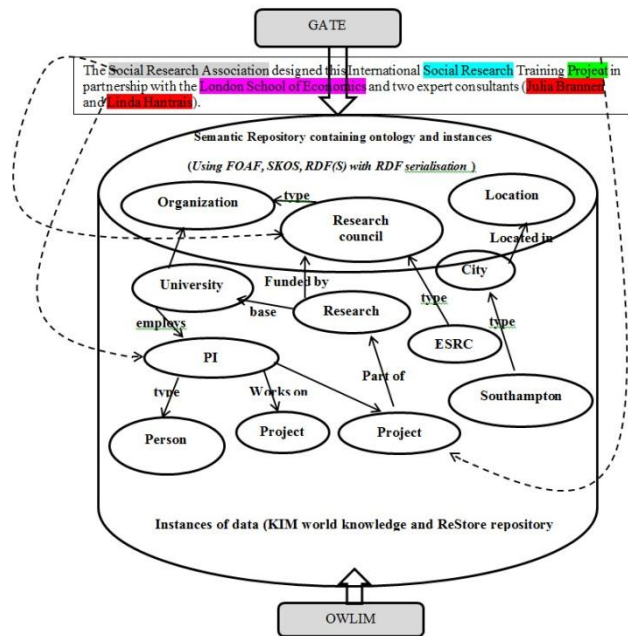


Figure 3: ReStore synthetic ontology architecture to be integrated into the KIM’s bootstrap ontology KIMO to aid in (a) annotating domain-specific (Social Science Research Methods) resources in ReStore repository (b) searching relevant information in integrated ontologies.

VIII. CONCLUSION AND FUTURE WORK

The core purpose of conducting this research is to find out if searching over the content of archived web resources could produce more accurate, meaningful and trust-worthy results by using semantic web technologies and linked data techniques. In today’s world of blogs, wikis, CMSs and social networking, the speed of publishing information on the web has greatly increased thereby affecting information consumption due to information overload. The problem is particularly acute in relation to web resources developed as part of specific

research projects. Despite their value, in most cases these disappear from the effective searching spectrum due to non-availability of meaningful metadata (except basic bibliographic page-specific metadata), lack of linkage to contemporary research and lack of full exposure to the mainstream web. Such highly specialised web resources become almost un-searchable, or in some cases misrepresented in search results in web archives where searching is performed using keywords matching algorithms.

This paper describes a research agenda which is focused on the addition of extra semantic meaning to the existing content of a web resources archive (in our case the Restore repository so as to permit more effective searching which takes account of similar content within and beyond their parental domain in order to make them part of the structured and meaningful web of data.

ACKNOWLEDGMENT

The authors acknowledge the support of ESRC Award No. RES-576-25-0023.

REFERENCES

- [1] P. Wu, A. Heok, and I. Tamsir, “Annotating the Web Archives—An Exploration of Web Archives Cataloging and Semantic Web,” *Digital Libraries: Achievements, Challenges and Opportunities*, pp. 12-21, 2006.
- [2] M. Costa, and M. Silva, "Towards information retrieval evaluation over web archives." pp. 37-38.
- [3] P. H. J. Wu, A. K. H. Heok, and I. P. Tamsir, “Annotating Web archives—structure, provenance, and context through archival cataloguing,” *New Review of Hypermedia and Multimedia*, vol. 13, no. 1, pp. 55-75, 2007/07/01, 2007.
- [4] C. Berkley, S. Bowers, M. B. Jones, J. S. Madin, and M. Schildhauer, "Improving data discovery in metadata repositories through semantic search," *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2009*. pp. 1152-1159.
- [5] P. Castells, F. Perdrix, E. Pulido, M. Rico, R. Benjamins, J. Contreras, and J. Lorés, "Neptuno: Semantic Web Technologies for a Digital Newspaper Archive," *Lecture Notes in Computer Science* C. Bussler, J. Davies, D. Fensel *et al.*, eds., pp. 445-458: Springer Berlin / Heidelberg, 2004.
- [6] "ReStore repository: A sustainable web resources repository," 14/01/2013, 2013; <http://www.restore.ac.uk>.
- [7] M. A. Hearst, “Natural’ Search User Interfaces,” *Communications of the ACM*, vol. 54, no. 11, pp. 60-6767, Nov., 2011.
- [8] S. Kara, Ö. Alan, O. Sabuncu, S. Akpınar, N. K. Cicekli, and F. N. Alpaslan, “An ontology-based retrieval system using semantic indexing,” *Information Systems*, 2011.

User Profiles in Information Web Portals

Carmen Moraga, M^a
 Ángeles Moraga
 Alarcos Research Group
 University of Castilla-La
 Mancha,
 Ciudad-Real, Spain
 Carmen.Moraga@alu.uclm.es
 MariaAngeles.Moraga@uclm.es

Angélica Caro
 Department of Computer
 Science and Information
 Technologies, University of
 Bio Bio,
 Chillán, Chile
 mcaro@ubiobio.cl

Coral Calero
 Alarcos Research Group
 University of Castilla-La
 Mancha,
 Ciudad-Real, Spain
 Coral.Calero@uclm.es

Rodrigo Romo Muñoz
 Department of Business
 Management, University of
 Bio Bio,
 Chillán, Chile
 rromo@ubiobio.cl

Abstract—The number of Web portals is increasing daily. These Web portals can be grouped into different types according to their purpose. One of these types is ‘Information Web Portals’ in which data quality is particularly important to users. This paper uses a survey to study the relevance that users place on a series of data quality characteristics in this type of Web portal. To do this, we determined various user profiles based on demographic aspects (gender, age range, level of studies and type of organization). We also analysed whether each profile prefers some characteristics to others. The results obtained will allow designers and developers to know which data quality characteristics they should place most emphasis on depending on the users of a particular Web portal.

Keywords-Data Quality; Web portal; Statistical Method.

I. INTRODUCTION

The Internet has become a powerful tool for communication purposes, both for the exchange of information and ideas and for learning and gaining knowledge, and it is even used for participation in local, national and international networking. The Internet has also come to be used in all aspects of our life in recent years [1]. One means to access information on the Internet is through Web portals. A Web portal is an entry point to the Internet [2-4]. Web portals select, organize, integrate distributed contents and enable viewers to access organizational services via the Internet [5-7]. Web portals play an increasingly specialized role in the online world [3] and they also allow the creation of a work and/or business environment in which users can navigate in a simple manner to rapidly obtain the information that they need [8], thus facilitating access to data on the Internet.

We have found several classifications of Web portal (i.e [9, 10]) depending on the type of purpose. In our work, we have divided Web portals into the following groups according to the principal type of activity that users wish to carry out:

- ‘The Search for and Reading of Information’: defined as those portals that the user uses solely to obtain information (e.g., a TV channel Portal to discover what programs are being shown, a cinema Portal to see what films are being shown, a newspaper Portal,

etc.) This type of portal is, therefore, merely informative.

- ‘Commercial Interaction’: determined by the fact that it is used to carry out some kind of on-line transaction, such as buying train or airline tickets, making downloads of a legal nature, transferring money, making payments, etc. This type of portal is, therefore, of a transactional nature.
- ‘Interaction with other People’: the important aspect here is the ability to relate to or get in contact with other people, known or otherwise. For example, social networks. This type of portal is, therefore, of the data-exchange type.

Although the same users may access each type of Web portal, their preferences may be different according to, amongst other things, their demographic aspects, for example, with regard to gender [11], [12]. Men are, according to [13], more analytical and therefore more objective, whilst women are more subjective and intuitive. The level of the user’s studies also influences Web portal use and, as the author states in [14], there is a significant difference between the Internet addiction scores of students and other professional groups.

Furthermore, Data Quality (DQ), which is often defined as the ability of a collection of data to meet user requirements [15, 16], is increasingly more important to Internet users [15, 17, 18]. This importance resides in the fact that users can use the data obtained to carry out everyday tasks and to make decisions both in their jobs and in their personal lives. With this, the area of Web portal data quality has consequently begun to emerge [19]. The Web portal owners are aware that DQ is important to increase user reliability, since users can clearly see its usefulness. Thus, when the degree of satisfaction increases, the number of customers that access the portal also increases.

All of the above led us to consider that it would be interesting to establish whether different user profiles exist as regards preferences towards the various characteristics of DQ in a Web portal. This was done by studying the following demographic aspects: the gender, age range, level of studies and type of organization to which Web portals users are linked.

In this paper, we particularly focus on the analysis of DQ characteristics for Web portals of the ‘Search for and Reading

of Information' type (which from here on will be referred to as 'Information Web Portals'). The DQ reference model used was SPDQM (SQuaRE-Aligned Portal Data Quality Model), a DQ model for Web portals which provides 42 DQ characteristics. These DQ characteristics are distributed in 4 categories: Intrinsic, Operational, Contextual and Representational. [20]. We shall use the set of DQ characteristics in the Contextual category, since it is the only category in which the importance that users place on certain DQ characteristics with regard to others may vary according to the type of Web portal.

Our study will allow us to determine, first, if all the DQ characteristics are important for users, and second, which are most relevant in comparison to the others according to the different user profiles established.

The results obtained will allow designers and developers to discover the most relevant DQ characteristics according to the aforementioned user profiles, in order to reinforce these DQ characteristics in the Web portal and thus satisfy their users.

The remainder of this paper is organized as follows: in Section 2, the DQ characteristics in the Contextual category are defined. Section 3 describes the data-collection method used. The various user profiles and their preferences as regards the DQ characteristics in the 'Information Web Portals' is determined in Section 4. In Section 5, guidelines for designers and developers are indicated. Finally, Section 6 presents our conclusions and future work.

II. THE CONTEXTUAL DQ IN WEB PORTALS

As already mentioned, in this work we shall focus on 'Information Web Portals' to determine the relevance of the DQ characteristics in the SPDQM Contextual category.

In SPDQM, the Contextual category highlights the requirement which states that data quality must be considered within the context of the task at hand. A certain type of context must therefore be considered in the Contextual category in order to establish the data quality in this environment, in our case, the 'Information Web Portals'.

This category contains 10 DQ characteristics and 6 DQ sub-characteristics (see Fig. 1). Further information on SPDQM can be found in [20].

Moreover, the importance that is placed on certain DQ characteristics in the Contextual category with regard to others may be different according to the various user profiles.

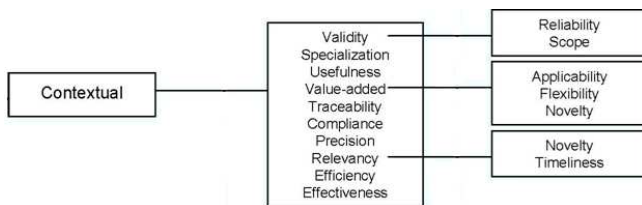


Figure 1. DQ characteristics in the contextual category

III. DATA COLLECTION SURVEY

In order to determine the user profiles and the DQ priorities for the different profiles of 'Information Web portal' users, we decided to carry out an unsupervised survey, in which questions related to this type of Web portals appeared. We used "the principles of survey research" proposed in [21, 22].

The questionnaire in this survey was formed of a total of 21 questions, 4 of which were general, related to demographic aspects (Table I), 16 of which were related to DQ characteristics in the Contextual category and 1 question concerning the definition of the term 'Contextual'. The questions concerning the DQ characteristics had to be easy to understand. Pre-test questionnaire was therefore first carried out with users who were experienced in the use of Web portals and whose feedback allowed us to modify the initial questions and obtain a definitive set of understandable questions for all types of Web portal users (Table II). In the questionnaire, only one response was possible because we used closed questions.

The questions were answered using an 11-point Likert-type interval scale, ranging from strongly disagree (0) to strongly agree (10).

The final questionnaire was distributed to a heterogeneous group of 200 Web portal users from Europe and Latin America, by e-mail or in printed format. The questionnaires were collected in the same manner, and 192 of them were returned, signifying that a response rate of 96% was obtained. However, 4 surveys had to be discarded because they were incomplete. We were therefore left with 188 surveys that could be used, thus obtaining a response rate of 94%.

Once the data obtained with the questionnaires had been collected, it was necessary to carry out a statistical analysis to investigate the results [23].

The starting point for this was the calculation of the Cronbach's alpha to estimate the reliability of the results. A value of 0.942 was obtained as a result of this, which indicated that the results had good internal consistence. The information was therefore reliable.

An analysis of the sample obtained is shown as follows.

IV. DETERMINING PREFERENCES FOR DQ CHARACTERISTICS ACCORDING TO THE USER PROFILE

In this section we shall determine whether all the DQ characteristics identified in the Contextual category in our SPDQM model are considered to be important by the users of this type of Web portals, and we shall also identify whether any of these DQ characteristics are more important than others depending on the various user profiles.

This will be done by carrying out a statistical analysis of the results obtained from the survey with the use of an SPSS statistical analysis tool and on the basis of the following steps:

TABLE I. QUESTIONS CONCERNING DEMOGRAPHIC ASPECTS

Gender: Male/Female
Level of studies COMPLETED: High School / Vocational Training/ University/ Post Graduate.
Type of organization with which you are linked (for study or work purposes). If there are various, please place them in the order in which most time is dedicated to them, from greatest to least: Education / Service Sector / Industrial - Commercial – Financial / Other (Please state which).
Age range: Under 25 / Between 25 and 35 / Between 35 and 45 / Between 45 and 55 / Over 55.

TABLE II. QUESTIONS CONCERNING THE CONTEXTUAL DQ CATEGORY

1.- The data should be sufficiently detailed to facilitate the task at hand.
2.- The data obtained from a Web portal should be true and reliable (believable).
3.- In general, the data in Web portals should be understandable for you to consider them valid. (This characteristic is related to those shown in questions 1 and 2).
4.- It should be possible to verify the data, and it would be appropriate to know their author and/or their source, and to be able to obtain a record of any modifications made to them.
5.- The data should be defined in accordance with a regulation, i.e., they should comply with pre-established standards (by, for example, showing dates, prices, etc.) thus avoiding situations in which doubts or different interpretations emerge.
6.- The data provided by a Web portal should contain the appropriate and specific information for the use to which they will be put.
7.- The data should adapt to user needs (e.g., they should be integrated into other applications or presented in different formats).
8.- The data should be useful and specially oriented towards the user community that will utilize them.
9.- The data should be innovative, thus allowing those who use them to obtain new knowledge.
10.- In general, the data should be suitable to allow the user to obtain advantages thanks to their use. (This characteristic is related to those shown in questions 7, 8 and 9).
11.- The data should be available in the shortest possible amount of time.
12.- In general, the data should be applicable and innovative, and should be available in a reasonable amount of time. (This characteristic is related to those shown in questions 9 and 11).
13.- The data offered by portals should be useful for their users, and should satisfy their needs.
14.- It should be possible to obtain the data by using the appropriate quantities and types of resources (by, for example, using the smallest possible number of links to locate the data desired).
15.- The data obtained from Web portals should be exact and concise, thus helping you to find relevant results.
16.- The data in Web portals should provide the information that users are seeking.
17.- The data provided by Web portals should have a level of quality that accords with the specific use to which you wish to put them, i.e., in the context of the specific area in which you wish to work with them.

A. Study of DQ Characteristics: Descriptive Statistical

A descriptive statistical analysis was used to show the minimum, maximum and mean values of all the DQ characteristics (including a question concerning the definition of the term ‘Contextual’) with the objective of verifying whether all these DQ characteristics are important in ‘Information Web Portals’. 11 characteristics with mean values over 8 were obtained, while the others have values of between 7.57 and 8, all of them being above 7.5. The mean values given by users are thus high (nearer to the maximum value of 10 than to the intermediate value of 5). This signifies that all the DQ characteristics in the Contextual category are, in effect, important to the users of ‘Information Web Portals’.

B. Creation of groups of DQ characteristics: Factor analysis

This sub-section was carried out with the use of a factorial analysis which allowed us to determine homogeneous groups of DQ characteristics (denominated as factors). These factors permitted a summary to be made of the relationship between the DQ characteristics considered. Each factor is independent of the others.

The results reveal the existence of three factors (Table III). Factor 1, Factor 2 and Factor 3 represent 53.48%, 7.47% and 5.83% of the total variance, respectively. These factors, when combined, can therefore explain 66.77% of the total variance, which can be interpreted as an acceptable percentage.

The Cronbach’s alpha was calculated for each of the factors obtained in order to estimate the reliability of the results. Factor 1 obtained a Cronbach alpha value of 0.874, Factor 2 obtained a Cronbach alpha value of 0.907 and Factor 3 obtained a Cronbach alpha value of 0.862. This signifies that the values obtained are good, and that the results are therefore reliable.

If we interpret Table III on the basis of the definitions of the DQ characteristics we obtain that the data in Factor 1 must be understandable to users if they are to consider them valid (validity), defined according to regulations and standards (compliance), true and reliable (reliability), contain detailed information to facilitate the task at hand (scope), contain the appropriate and specific information needed for the use to which they will be put (specialization), are verifiable, and the author or source from which they came is known (traceability) and can be adapted to user needs (flexibility). For Factor 2, the users wish to find what they are looking for (effectiveness), with concise data that will allow them to find relevant results (precision), that satisfy their needs (usefulness) and using an appropriate quantity of resources (efficiency). In Factor 3 the users are interested in the fact that the data are applicable, innovative (relevancy) and are oriented towards a destination community (applicability), that they allow them to acquire new knowledge (novelty), to obtain advantages from them (value-added) and that they are available in the shortest possible amount of time (timeliness).

TABLE III. FACTORIAL ANALYSIS

Factor 1	Factor 2	Factor 3
Validity	Effectiveness	Relevancy
Compliance	Efficiency	Timeliness
Reliability	Precision	Novelty
Scope	Usefulness	Value-added
Specialization		Applicability
Traceability		
Flexibility		

C. Creation of user profiles: Cluster analysis

The DQ characteristics have now been organized into factors. However, our intention was to discover whether any of the DQ characteristics are more relevant than others according to the different user profiles. To do this, we carried out a cluster analysis in order to group the previously identified factors by resemblance or similitude. Three groups are also obtained in this case. Each cluster may be formed of one or various factors, depending on the importance that each cluster gives to each factor. Cluster 1 contains the DQ characteristics of Factor 2. Cluster 2 contains the DQ characteristics of Factor 1 and Factor 3 and Cluster 3 contains the DQ characteristics of Factor 3 (see Table IV).

Our next objective was to determine the relationship between the demographic aspects and each of the clusters identified. This was done by using the contingency tables shown in Table V and the following steps:

1°.- We determined which variable had the greatest value for each demographic aspect and each cluster. For example, Cluster 1 and the demographic aspect 'gender' give us a value of '68' which corresponds with the variable 'male', while the demographic aspect 'level of studies' gives us a value of '54' which corresponds with the variable 'University'.

2°.- The value obtained was compared with the other values in this variable for the other clusters. In the example, the values are '55' and '44' for the variable 'male' in Clusters 2 and 3, respectively, and the values are '50' and '56' for the variable 'University' in Clusters 2 and 3, respectively.

3°.- We chose the greatest of the values obtained for this variable. In the example, we selected the value '68', which is in Cluster 1, for the variable 'male', and the value '56',

which is in Cluster 3, for the variable 'University', and this variable was discounted in Cluster 1 (the values shown in bold type in Table V).

4°.- For those variables which did not yet have a selected value, we chose the highest value in its row. For example, for the 'Vocational Training' variable, whose values are '7', '13', and '20' in Clusters 1, 2 and 3, respectively, we chose the value '20' which is in Cluster 3 (the values shown in italics in Table V). By following these steps we therefore obtain the values shown in bold type and in italics (highlighted values) in Table V.

These results allow each variable of each demographic aspect to be situated in one of the three clusters, which allows three user profiles to be determined, as is shown in Table VI.

The user profile 1 is composed of: men between 25 and 45 with a postgraduate level of studies who work in education belong to Cluster 1 and give priority to the DQ characteristics in Factor 2 (Effectiveness, Efficiency, Precision and Usefulness). In the user profile 2, there are users under 25 and over 55 years of age with High School who belong to an Industrial, Commercial or Financial organization, of which Cluster 2 is composed, give priority to the DQ characteristics that correspond with Factor 1 (Validity, Compliance, Reliability, Scope, Specialization, Traceability and Flexibility) and Factor 3 (Relevancy, Timeliness, Novelty, Value-added and Applicability). The user profile 3 consists of: women between the ages of 45 and 55 with vocational training or university studies from the service sector or another (i.e., not Education, Industrial, Commercial or Financial), belong to the cluster 3 and these users place more relevance on the DQ characteristics in Factor 3 (Relevancy, Timeliness, Novelty, Value-added and Applicability).

TABLE IV. CLUSTER ANALYSIS

Cluster		
1	2	3
Factor 2	Factor 1	Factor 3
	Factor 3	

TABLE V. RELATIONSHIP BETWEEN DEMOGRAPHIC ASPECTS AND CLUSTERS

Demographic Aspect	Variable	Cluster (%)		
		1	2	3
Gender	Male	68	55	44
	Female	32	45	56
Age	Under 25	13	36	16
	Between 25 and 35	32	23	28
	Between 35 and 45	39	7	20
	Between 45 and 55	13	23	30
	Over 55	3	<i>11</i>	6
Level of Studies	High School	7	32	16
	Vocational Training	7	13	<i>20</i>
	University	54	50	56
	Postgraduate	32	5	8
Type of Organization	Education	58	41	35
	Industrial-commercial-financial	7	<i>16</i>	4
	Service Sector	26	22	<i>34</i>
	Other	9	21	<i>27</i>

TABLE VI. SUMMARY OF THE RELATIONSHIP BETWEEN DEMOGRAPHIC ASPECTS AND CLUSTER

Profile/Cluster	Factor	DQ Characteristics	Gender	Age	Level of Studies	Type of Organization
1	Factor 2	Effectiveness, Efficiency, Precision, Usefulness	Male	Between 25 and 35 Between 35 and 45	Postgraduate	Education
2	Factor 1 Factor 3	Validity, Compliance, Reliability, Scope, Specialization, Traceability, Flexibility Relevancy, Timeliness, Novelty, Value-added, Applicability		< 25 > 55	High School	Industrial-commercial-financial
3	Factor 3	Relevancy, Timeliness, Novelty, Value-added, Applicability	Female	Between 45 and 55	Vocational Training University	Services Other

Limitations of the study. This work has been carried out in a systematic manner. Nevertheless, we are conscious that it has certain limitations. The first concerns the quality model used since we have limited the research to DQ characteristics in the Contextual category of the SPDQM model.

We have also limited the type of Web portal with which we have worked, and have focused solely upon ‘Information Web portals’ and have obtained those DQ characteristics which are most relevant according to the users’ different demographic aspects.

In future works we shall consider the other types of Web portals and we shall analyse the other categories in the model.

As will be noted, all these changes will make the results that will be obtained more global, which is our eventual objective.

V. GUIDELINES TO DESIGNERS AND DEVELOPERS

In this section, we show the method used in order to create guidelines for designers and developers so that they will know which DQ characteristics are most important according to the type of user to which the ‘Information Web portals’ are oriented and which they intend to develop or modify. The designers and developers will therefore be able to deal with and use them.

We used the following steps:

1.- The type of user towards which the Web portal is oriented is identified. The type of user will be determined by the demographic aspects: gender, age range, level of studies and type of organization with which they are linked (for study or work purposes). The following examples will allow us to analyse the results:

Example 1: A ‘Information Web portal’ oriented principally towards men between 35 and 45 with postgraduate studies and belonging to educational organisations.

Example 2: A ‘Information Web portal’ oriented principally towards women under 25 with ‘High School’ studies and belonging to organisations in the services sector.

2.- The cluster and factor belonging to each demographic aspect is obtained (see Table IV and Table VI).

For example 1:

- Men belong to Cluster 1 with Factor 2.
- People between 35 and 45 are in Cluster 1 with Factor 2.
- People with university studies are in Cluster 1 with Factor 2.
- People in educational organisations are in Cluster 1 with Factor 2.

For example 2:

- Women belong to Cluster 3 with Factor 3.

- People under 25 are in Cluster 2 with Factors 1 and 3.
- People with ‘High School’ studies are in Cluster 2 with Factors 1 and 3.
- People from organisations in the services sector belong to Cluster 3 with Factor 3.

3.- The designers and developers will put special emphasis on the DQ characteristics in the factor that appears most often. If none of the factors are repeated or various factors are repeated the same amount of times, they will consider the DQ characteristics of those factors.

From example 1: emphasis should be placed on the DQ characteristics in Factor 2. Therefore, in ‘Information Web portals’, their users are interested in data that are actually seeking (Effectiveness), are obtained using appropriate quantity and resources (Efficiency), are exact (Precision) and that satisfy their needs (Usefulness).

From example 2: most importance will be placed on the DQ characteristics from Factor 3. In this case, users are interested in the fact that the data are applicable and innovative (Relevancy), are obtained in the least possible amount of time (Timeliness), that they allow them to acquire new knowledge (Novelty), permit advantages to be attained (Value-added) and data that are oriented towards a destination community (Applicability) in ‘Information Web portals’.

We believe that with this type of guidelines, designers and developers will increase the data quality of the Web portals as perceived by their users, since they will be ‘tailor-made’.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a study whose intention is, on the one hand to establish the importance that Web portal users place on a group of DQ characteristics and, on the other to determine whether these DQ characteristics have a different level of relevance according to the various user profiles.

We determined the set of DQ characteristics to be analysed by focusing on those corresponding to the Contextual category in the DQ model for Web portals denominated as SPDQM [20]. Given that the context in which DQ characteristics are analysed has an influence on this Contextual category, we decided to focus our investigation on Web portals of the ‘Information Web Portals’ type.

Furthermore, the following demographic aspects: gender, age range, level of studies and type of organization to which the users were linked, were used to identify the various user profiles.

The entirety of this study was carried out by using a survey as a starting point and then analysing the results

obtained from the questionnaire by using the SPSS statistical analysis tool. The questionnaire in this survey contained a total of 21 questions, 4 of which were general, related to demographic aspects, 16 of which were related to DQ characteristics in the Contextual category and 1 question concerning the definition of the term 'Contextual'.

In the analysis that followed we identified three different user profiles and we grouped the DQ characteristics according to those user profiles such that we are now able to indicate the set of DQ characteristics to which more attention should be paid in the DQ of a Web portal according to a particular user profile.

All of this should enable designers and developers to be guided in the construction of Web portals, in order to make them more appropriate for the users at which they are aimed.

Our short-term future work will be to determine the importance of DQ characteristics in the Contextual category for Web portals of the types 'Commercial Interaction' and 'Interaction with other People'. We shall then compare our current results with the results obtained in the other types of portals. In this way, we will see whether all the types of user profiles place importance on the same DQ characteristics in all types of portals. Our eventual intention is to make our model available to users and developers through a free tool.

ACKNOWLEDGMENT

This research has been funded by the following projects: ORIGIN (CDTI-MICINN and FEDER IDI-2010043(1-5)), PEGASO/MAGO project (Ministerio de Ciencia e Innovacion MICINN and Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01), EECCOO (MICINN TRA2009_0074), VILMA (JCCM PEII 11-0316-2878) and GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

REFERENCES

- [1] M. Komathi and I. Maimunah, "Influence of gender role on Internet usage pattern at home among academicians," *The Journal of International Social Research*, vol. 2, 2009.
- [2] K. Laudon and C. Traver, *E-commerce 2009* (5th ed.): Upper Saddle River, NJ: Prentice Hall, 2008.
- [3] P. Sharma and J. Gupta, "A framework for enterprise-wide-e-commerce portal for evolving organizations," in A. Tatnall (Ed.), *Web portals: The new gateways to internet information and services*. Hershey, PA: Idea Group ed, 2005.
- [4] R. Stair and G. Reynolds, *Principles of information systems*. Boston, MA: Thomson Course Technology, 2008.
- [5] A. S. Al-Mudimigh, Z. Ullah and T. A. Alsubaie, "A framework for portal implementation: A case for Saudi organizations," *International Journal of Information Management*, vol. 31, pp. 38-43, 2011.
- [6] P. Yang, et al., "The emerging concepts and applications of the spatial web portal," *Photogrammetric Engineering & Remote Sensing*, vol. 73, pp. 691-698, 2007.
- [7] M. A. Domingues, C. Soares and A. M. Jorge, "A Web-Based System to Monitor the Quality of Meta-Data in Web Portals," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IATW'06)*, 2006, pp. 188-191.
- [8] H. Collins, "Corporate Portal Definition and Features," AMACOM, 2001.
- [9] M. Davydov, *Corporate Portals and e-Business Integration Emerging Business Technology Series*, 2001.
- [10] C.-T. Liu, T. C. Du and H-H. Tsai, "A study of the service quality of general portals," *Information & Management*, vol. 46, pp. 52-56, 2009.
- [11] A. Durdell and Z. Haag, "Computer self efficacy, computer anxiety, attitudes towards the Internet and reported experience with the Internet, by gender, in an East European sample," *Computer in Human Behavior*, vol. 18, pp. 521-535, 2002.
- [12] M. E. Hupfer and B. Detlor, "Gender and Web information seeking: A self-concept orientation model.," *Journal of the American Society for Information Science and Technology*, vol. 57, pp. 1105-1115, 2006.
- [13] D. Kim, X. Lehto and A. Morrison, "Gender differences in online travel information search: Implications for marketing Communications on the Internet," *Tourism Management Information Systems*; Armonk; Spring, vol. 28, pp. 423-433, 2007.
- [14] C. Şahin, "An analysis of Internet addiction levels of individuals according to various variables," *TOJET: The Turkish Online Journal of Educational Technology*, vol. 10, pp. 60-66, 2011.
- [15] C. Cappiello, C. Francalanci and B. Pernici, "Data quality assessment from the user's perspective," in *Proceeding on International Workshop on Information Quality in Information Systems (IQIS2004)*, Paris, France. ACM, 2004, pp. 68-73.
- [16] D. Strong, Y. Lee and R. Wang, "Data Quality in Context," *Communications of the ACM*, vol. 40, pp. 103-110, 1997.
- [17] L. Pipino, Y. Lee and R. Wang, "Data Quality Assessment," *Communications of the ACM*, vol. 45, pp. 211-218, 2002.
- [18] R. Wang and D. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of Management Information Systems*; Armonk; Spring, vol. 12, pp. 5-33, 1996.
- [19] M. Gertz, T. Ozsu, G. Saake and K-U. Sattler, "Report on the Dagstuhl Seminar "Data Quality on the Web"," *SIGMOD Record.*, vol. 33, pp. 127-132, 2004.
- [20] C. Moraga, M. Moraga, C. Calero and A. Caro, "SQuARE-Aligned Data Quality Model for Web Portals," in *9th International Conference on Quality Software (QSIC 2009)*, 2009, pp. 117-122.
- [21] B. Kitchenham and S. Pfleeger, "Principles of survey research part 2: designing a survey," *SIGSOFT: Software Engineering Note*, vol. 27, pp. 18-20, 2002.
- [22] B. Kitchenham and S. Pfleeger, "Principles of survey research part 3: Constructing a survey instrument," *SIGSOFT: Software Engineering Note*, vol. 27, pp. 20-24, 2002.
- [23] M. J. Norusis, *SPSS 11.0 Guide to Data Analysis*: Prentice Hall, 2002.

A Graph Model of Events Focusing on Granularity and Relations Towards Organization of Collective Intelligence on History

Minoru Naito
Graduate School of Informatics
Kyoto University
Kyoto, Japan
naito@db.soc.i.kyoto-u.ac.jp

Yasuhito Asano
Graduate School of Informatics
Kyoto University
Kyoto, Japan
asano@i.kyoto-u.ac.jp

Masatoshi Yoshikawa
Graduate School of Informatics
Kyoto University
Kyoto, Japan
yoshikawa@i.kyoto-u.ac.jp

Abstract—We propose Event Graph Model to organize collective intelligence on history and acquire useful knowledge of events. Our Event Graph Model represents a complicated event of various granularities as a graph composed of nodes corresponding to smaller-grained events and edges corresponding to relations between the events. This model is expected to be useful for finding events whose structures are similar to each other. In other words, our model would be able to distinguish such events from ones similar by vocabulary but different substantially, although previously proposed keyword-based event analyses could not.

Keywords-Collective intelligence; Model of Events.

I. INTRODUCTION

The investigation of history is important for the future of humankind. As it has been said that history repeats itself, we might be able to infer a solution of a current problem by drawing a lesson from knowledge of past events similar or related to the problem. However, historical knowledge is too massive for common people to find and understand such events.

Automatic analysis and organization of historical knowledge could be helpful for people to find and understand events. Several methods have been proposed to analyze events utilizing news articles [1], [2], [3]. Although these methods are useful for understanding events roughly, they are insufficient for our motivation that people draw a lesson from historical knowledge. For example, the two articles illustrated in Figure 1 represent events similar by vocabulary but different radically; in the left article a famine first occurred and consequently a reform was carried out; in the right article a famine wasted the reform. The methods above might not distinguish them because they regard an article as an event and use few keywords to represent an event. Therefore, a model of events should be able to analyze the granularity of events more flexibly, and organize various kinds of relations between events.

In this paper, we propose an event graph model, abbreviated to EGM, that represents a large event, named *composite event*, in a given article by a graph whose node corresponds to a small event, named *minimal event*, and

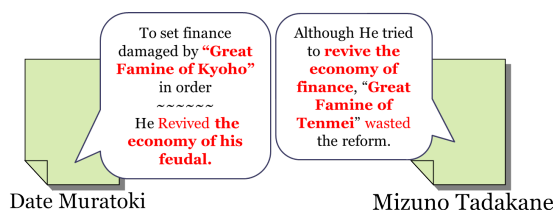


Figure 1. Events that are similar by vocabulary but different.

edge represents the relation between two minimal events. In order to establish EGM, we first define two types of a minimal event which can not be divided on a sentence in the given article: an *event noun phrase* and an *event predicate-argument structure*. An event noun phrase is a noun phrase which can represent an event by itself. If we divide the noun phrase into several elements, then no elements can represent an event. Therefore, it is adequate that we regard an event noun phrase as a minimal event. A predicate-argument structure in a sentence, abbreviated to PAS, is called an event predicate-argument structure, abbreviated to EPAS, if its predicate represents an action related to an event. For example, sentence “Date Muratoki revived the economy of his feudal domain” contains an EPAS because its predicate “revived” represents an action in the economic revival; on the other hand, the predicate of sentence “Tokugawa Ieyasu had been told as a descendant of Nitta Yoshishige” is not regarded as an action of a historical event. If a sentence has two predicates representing actions of events, then we regard the sentence contains two EPAS. We then introduce the following eight kinds of a relation between minimal events: deployment, illustration, parataxis, causality, paradox, progress, means, and situation. In contrast to models dealing with the causality relation between events [1], our eight kinds of relations are expected to find similar events more accurately. We also present an idea how to determine which events have a relation by utilizing the centering theory proposed by Grosz et al. [5], [6]. Figure 2 illustrates an example of composite events in

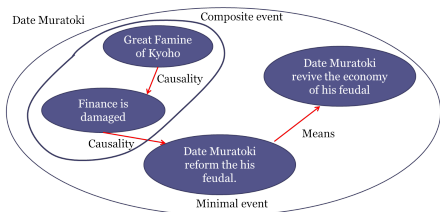


Figure 2. Examples of minimal and composite events.

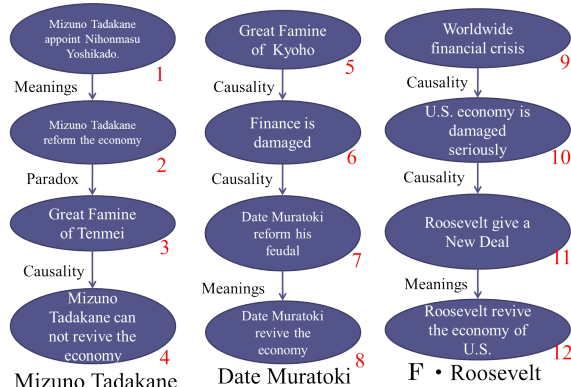


Figure 3. Composite events for Mizuno, Date, and Roosevelt.

our EGM constructed from several sentences in Wikipedia article “Date Muratoki.” Each solid circle represents a minimal event; an arrow represents the relation between two minimal events. Each open circle represents a composite event; note that an arbitrary connected subgraph can be regarded as a composite event despite only two composite events are shown in the figure.

Once we succeed in establishing EGM, then we might be able to find similar events accurately enough for our motivation. For example, Figure 3 portrays three composite events extracted from three Wikipedia articles “Mizuno Tadakane,” “Date Muratoki,” and “Franklin Roosevelt.” The composite events for Mizuno and Date contain similar minimal events: nodes 2 and 7 represent reforms, and nodes 3 and 5 represent famines. However, the orders of the minimal events are different: the reform is before the famine for Mizuno, while the famine is before the reform for Date. Therefore, these events should be regarded as different. On the other hand, the composite event for Date is similar to that for Roosevelt in both of the content of minimal events and their relations. We plan to propose a method for finding events having similar structures by applying techniques based on graph theory including graph isomorphism and topological minor.

As work in progress, we have implemented the extraction of minimal events and the determination which events have a relation. We also have a preliminary experiment on the extraction of minimal events. We use Wikipedia articles about people as a dataset because they contain a plenty knowledge of historical events and the following charac-

teristics useful for acquiring the information of events: the explanation of events are usually described in chronological order; specific expression can be normalized easily using wiki links; attributes of an article are readily identifiable by category; an article has fewer subjective opinions and has many descriptions of objective facts; entity information is included in the article because of the principle of “one entity – one article.” For example, we utilize wiki links to identify event noun phrase in the implementation of the extraction of minimal events.

II. RELATED WORK

Ishii et al. [1] proposed a method for constructing a network of events to illustrate causality relations between events.

Ikeda et al. [2] proposed a method for representation of an event by 5W1H(who, when, where, what, why, how) and predicate information that are extracted from a news article.

Chin-man et al. [3] proposed a method for acquisition of detecting knowledge from collection of past news. They show that analysis of references to the past in news articles allows us to gain a lot of insight into a historical event.

Murakami et al. [4] proposed a Statement Map showing the relation of statements described on the Web by creating a database of knowledge between events using data and a thesaurus of verbs. This system helps users with selection and aggregation of information.

In this way, many earlier studies have been conducted to extract relations between statements to help users. However, few studies have proposed an event-based comparison with consideration of the flow across multiple sentences and statements.

III. EVENT GRAPH MODEL

In this section, we explain the detail of EGM. Because events takes various granularity, we define two types of an event: a minimal event and a composite event.

A. Minimal Event

Minimal events are nodes of a graph in our EGM. A minimal event on a sentence consists of its elements representing an event; if the elements are divided into two or more parts, then any of the parts can not represent an event. We define two types of an minimal event: an event noun phrase and EPAS.

1) *Event Noun Phrase*: An event noun phrase is a noun phrase which can represent an event by itself. We regard a noun phrase as event noun phrase if it has a wiki link to an article of a category related to events. In this work we use “incident,” “disaster” and “battle” categories. Because a noun phrase itself does not contain a plenty of information about the event, we employ a technique for compensating for such information. It is known that the first sentence of the Wikipedia article of an entity can be regarded as a summary

of the entity. Therefore, we extract an EPAS explained in Section III-A2 from the first sentence of the article linked from the event noun phrase.

2) *Event Predicate-Argument Structure*: A complex sentence that includes plural predicates is frequently appear on Japanese Wikipedia articles. Therefore, we should not regard a sentence as an event. Instead, we focus on a PAS containing a predicate representing an event in a sentence. One PAS consists of a predicate and terms that have a case relation with the predicate. Let $p(c_1:a_1, c_2:a_2, \dots, c_\ell:a_\ell)$ be a PAS consists of predicate p and ℓ terms, each a_i having case relation c_i with p for $i = 1, 2, \dots, \ell$.

We extend the PAS in order to represent a minimal event, because a predicate often corresponds to an action related to an event. However, not all the predicates correspond to such actions. Therefore, we propose a rule-based definition of an EPAS to use only predicates corresponding to such actions. The proposed rules are based on Japanese language. We use a verb or a “sahen noun” as predicates in an EPAS. Here, a sahen noun is a Japanese-specific noun that functions as a verb if it is followed by word “suru”, which is the equivalent of “do” in English. We explain our four rules (a)-(d) below; because it is difficult to explain them completely in English, we only describe the summary of them. (a) We do not use a verb that represents hearsay and inference. For example, in sentence “X was said to be serious” of the article for entity X, “said” is not an action of events in which X took part in. (b) We do not use an auxiliary verb. For example, in sentence “He was made to win,” “win” would be preferable for representing an action to “made.” (c) We use a sahen noun if it is followed by a punctuation. In Japanese, such a representation is considered as an omission of word “suru” explained above. (d) We use a sahen noun if it is a part of an adverbial clause. For example, in adverbial clause “by the attack of enemy,” noun “attack” is an action of the enemy.

Our rules, especially (c) and (d), reflect characteristics of Japanese Wikipedia articles. A single sentence in them frequently contains a larger amount of information than that contained in a sentence in usual documents. Therefore, it frequently has omissions of “suru” explained in (c) and adverbial clauses explained in (d). Although our rules use Japanese-specific sahen nouns, we might be able to propose equivalent rules for English Wikipedia articles by utilizing well-known characteristics of English grammars.

Once we determine whether we use predicate p in a given sentence, then we determine the other elements in the EPAS which contains p . We use only two kinds of a case relation, nominative and accusative, because these are necessary and sufficient for determining whether two minimal events have a relation corresponding to an edge in our EGM. We also decide to use keywords which are nouns complement of the predicate but are not nominative and accusative. The keywords help us to compare and determine the connection of minimal events. Therefore, we denote an EPAS containing

Table I
EXAMPLES OF A PREDICATE STRUCTURE OF EVENTS.

Predicate	Nominative	Accusative	Keyword
get	finance	damage	-
reform	Date Muratoki	feudal	-
revive	Date Muratoki	economy	feudal

predicate p by

$$p(\text{nominative:}a_n, \text{accusative:}a_a, \text{keywords:}k_1, k_2, \dots),$$

where word a_n is the nominative of the predicate p , and word a_a is the accusative of p , and words k_1, k_2, \dots are the keywords. The nominative and accusative words can be extracted from a given sentence by using morphological analysis tools such as MeCab [7] and CaboCha [8]. At the same time, keywords can be detected. Table I depicts an example of EPASs corresponding to three minimal events illustrated in Figure 2.

B. Composite Event

A composite event represents a large event in a given article by a graph whose node corresponds to a minimal event and edge represents the relation between two minimal events. As illustrated in Figure 2, a composite event can be represented by a connected subgraph. Every edge has a label representing a kind of the corresponding relation. A causal relation exists between minimal events “Great Famine of Kyoho” and “finances damaged” in the Figure.

1) *Relation between Minimal Events*: There are studies about discourse structure analysis used to examine the relation between a sentence and another sentence [6]. Various kinds of relations are defined in each study. However, these kinds of relations are not sufficient for represent the relations between minimal events which might have smaller granularity than a sentence, that is, which can be appear more than twice in a single sentence. Therefore, we extend ideas used in the discourse structure analysis in order to capture the relations between minimal events.

We decide to use eight kinds of relations which are classified into three categories: deployment, illustration and parataxis in structural relation category; causality, progress, and paradox in semantic relation category; means and situation in complementary relation category. Structural relations are those necessary to capture the structure of a document. Wikipedia articles about historical figures often describe actions of a person in chronological order. Therefore, these relations are important to compare the chronological structures of events. We consider that the causality, progress, and paradox of events are interpretation of history by human subjective. Therefore, semantic relations would be the most important kind of relations for our motivation. A complementary relation is a particular relation between minimal

events in the same sentence; one event gives complementary information about the other.

2) *Determination of Unity between Minimal Events:* We here summarize our idea for determining which minimal events have a relation by utilizing centering theory proposed by Grosz [5]. The centering theory can be used for extracting the relation between adjacent sentences. Umezawa et al. [6] proposed a system named DIA that performs discourse structure analysis extend the centering theory so that not only the adjacent sentences but also neighbor sentences are used. We apply this extended centering theory to extraction of the relation between minimal events.

The centering theory uses the nominative words and accusative words mainly in sentences to determine the relation between them. Therefore, we can apply it to determine the relation between two minimal events, each of them is represented by an EPAS which includes nominative and accusative words. The extended centering theory assigns a score representing how much the probability there is a relation between the pair. If the nominative words of the events in a pair are the same, then the highest score is assigned. On the other hand, if there is no common words in them, the score is lowest. In addition to the scoring used in the extended centering theory, we assign a high score to the pair of minimal events extracted in the same sentence because they would have a relation with a high probability. We omit the details for space.

IV. EVALUATION

A. Selection of predicate

In order to make sure the effects of our rules defined in Section 3 that select verbs and sahen nouns as predicates, we apply the rules to 5 Wikipedia articles about historical Japanese people including “Date Muratoki” and “Takeda Motoshige.” Table II shows the ratio of predicates excluded by our rules. The “Precision” column indicates the ratio of predicates which are not EPASs actually to the ones excluded by our rules. The “Recall” column indicates the ratio of ones which are not EPASs and are excluded by the rules correctly to all the verbs and sahen nouns. Most of the predicates excluded by our rules are actually not EPASs. Therefore, our rules effective in precision. On the other hand, the recall is not high sufficiently. We consider that rules should be added for proper exclusion.

Table III shows the number of relations extracted by the extended centering theory (base line) and our method. Our method can find more relation than the baseline.

B. Predicate Search

In order to make sure the benefits of our approach with event graph model, we compare three simple methods of search from Wikipedia articles of Japanese people.

Queries of search are composite events that is organized by two minimal events. We use four queries: (a) He became

Table II
RESULTS OF APPLYING OUR RULES OF PREDICATES.

Predicate	Excluded predicates	precision	recall	example
Verbs	3.7 %	87.5 %	36.8 %	said estimated
Sahen nouns	80.7 %	98.6 %	89 %	attack recovery

Table III
THE NUMBER OF RELATIONS BETWEEN MINIMAL EVENTS.

Article Name	Minimal events	Base line	Our method
Date Muratoki	46	40	43
Nasu Takasuke	56	27	32
Takeda Motoshige	68	46	60

Table IV
PRECISION OF PREDICATE SEARCH BY EACH METHODS.

Query Name	Baseline	Use edge	Use edge and label
a	66.7 %	100 %	100 %
b	20 %	95.2 %	100 %
c	37.5 %	100 %	-
d	2.4 %	100 %	100 %

Table V
RECALL OF PREDICATE SEARCH BY EACH METHODS.

Query Name	Use edge	Use edge and label
a	40 %	40 %
b	66.7 %	43.3 %
c	66.7 %	-

a priest and renamed; (b) He succeed to a house because of retirement of his father; (c) He is helped to the lord, and returned to secular life; (d) He killed his men and he lost trust. We use three simple methods for search of these queries: AND search of two predicates; search with the information of predicates and the edge of the predicates in our event graph model; search with the information of predicates, the edge and its label in our event graph model. For this experiment, we use 500 Wikipedia articles of Japanese feudal lords. We also use the Japanese Wordnet [9] for comparison of two predicates.

Table IV, V shows the precision and recall of these methods. Because there is not result on query (c) by use edge and label, there is a blank column in tables. In Table V, because query (d) has an only 1 correct article, the line of query (d) is omitted.

Table IV shows there are many noisy results in baseline method because two predicates are not always related in articles. On the other hand, our methods show high scores in precision. It means our method is effective in getting knowledge of composite events.

However, Table V shows the methods show low scores in recall. In most cases, it is caused by different written forms, illegal connection of minimal events and labels. In addition,

there is a case that some minimal events exist between two minimal events of query. To resolve these problems, the method of comparison of graph models is needed.

V. CONCLUSION AND FUTURE WORK

We proposed an event graph model in order to organize collective knowledge of history and find useful knowledge from it. Focusing on the granularity and relation between events especially, we proposed a minimal event and a composite event which represents relations among minimal events. We actually found minimal events and determined whether each pair of minimal events has a relation using Wikipedia articles about people.

A candidate of future work is to propose a method for determining the kind of the relation of minimal events. We also plan to construct a method for comparing composite events by varying the granularity. We consider that graph theory, including graph isomorphism and topological minor, would be useful for our plan.

REFERENCES

- [1] Hiroshi Ishii, Qiang Ma, and Masatoshi Yoshikawa. Incremental Construction of Causal Network from News Articles, *Journal of Information Processing*, Vol. 20, No.1, pp. 207-215, 2012 [accessed December, 2012]
- [2] Takahiro Ikeda, Akitoshi Okumura, and Kazunori Muraki. Information Classification and Navigation Based on 5WIH of the Target Information, *Computational Linguistics and the Association for Computational Linguistics 1998*, Montreal, pp. 571-577, 1998 [accessed December, 2012]
- [3] Ching-man Au Yeung and Adam Jatowt. Studying how the past is remembered: towards computational history through large scale text mining, *Conference on Information and Knowledge Management Proc. of the 20th*, Glasgow, pp.1231-1240, 2011 [accessed December, 2012]
- [4] Koji Murakami, Eric Nichols, Junta Mizuno, Yotaro Watanabe, Shouko Masuda, Hayato Goto, Megumi Ohki, Chitose Sao, Suguru Matsuyoshi, Kentaro Inui, and Yuji Matsumoto. Statement map: reducing web information credibility noise through opinion classification, *Analytics for Noisy Unstructured Text Data 2010*, Toronto, pp. 59-66, 2010 [accessed December, 2012]
- [5] Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. Centering: A Framework for Modeling the Local Coherence of Discourse, *Association for Computational Linguistics*, 21, Cambridge, Massachusetts, pp. 203-225, 1995 [accessed December, 2012]
- [6] Toshiyuki Umezawa and Minoru Harada. Discourse Structure Analysis System DIA Based on Centering Theory and Object Knowledge, *Journal of Natural Language Processing* 18(1), pp. 31-56, 2011 [accessed August, 2011]
- [7] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, <http://mecab.sourceforge.net/> [accessed December, 2011]
- [8] CaboCha: Yet Another Japanese Dependency Structure Analyzer, <http://code.google.com/p/cabocha/> [accessed October, 2012]
- [9] Japanese Wordnet (v1.1) (c) NICT, 2009-2010, <http://nopwww.nict.go.jp/wn-ja/index.en.html> [accessed December, 2012]