

International Journal on Advances in Internet Technology



The *International Journal on Advances in Internet Technology* is published by IARIA.

ISSN: 1942-2652

journals site: <http://www.ariajournals.org>

contact: petre@aria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Internet Technology, issn 1942-2652
vol. 3, no. 3 & 4, year 2010, http://www.ariajournals.org/internet_technology/

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Internet Technology, issn 1942-2652
vol. 3, no. 3 & 4, year 2010, <start page>:<end page>, http://www.ariajournals.org/internet_technology/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA

www.aria.org

Copyright © 2010 IARIA

Editor-in-Chief

Andreas J Kassler, Karlstad University, Sweden

Editorial Advisory Board

- Lasse Berntzen, Vestfold University College - Tonsberg, Norway
- Michel Diaz, LAAS, France
- Evangelos Kranakis, Carleton University, Canada
- Bertrand Mathieu, Orange-ftgroup, France

Digital Society

- Gil Ariely, Interdisciplinary Center Herzliya (IDC), Israel
- Gilbert Babin, HEC Montreal, Canada
- Lasse Berntzen, Vestfold University College - Tonsberg, Norway
- Borka Jerman-Blazic, Jozef Stefan Institute, Slovenia
- Hai Jin, Huazhong University of Science and Technology - Wuhan, China
- Andrew Kusiak, University of Iowa, USA
- Francis Rousseaux, University of Reims - Champagne Ardenne, France
- Rainer Schmidt, University of Applied Sciences – Aalen, Denmark
- Asa Smedberg, DSV, Stockholm University/KTH, Sweden
- Yutaka Takahashi, Kyoto University, Japan

Internet and Web Services

- Serge Chaumette, LaBRI, University Bordeaux 1, France
- Dickson K.W. Chiu, Dickson Computer Systems, Hong Kong
- Matthias Ehmann, University of Bayreuth, Germany
- Christian Emig, University of Karlsruhe, Germany
- Mario Freire, University of Beira Interior, Portugal
- Thomas Y Kwok, IBM T.J. Watson Research Center, USA
- Zoubir Mammeri, IRIT – Toulouse, France
- Bertrand Mathieu, Orange-ftgroup, France
- Mihhail Matskin, NTNU, Norway
- Guadalupe Ortiz Bellot, University of Extremadura Spain
- Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science – London, Canada
- Dumitru Roman, STI, Austria
- Pierre F. Tiako, Langston University, USA
- Ioan Toma, STI Innsbruck/University Innsbruck, Austria

Communication Theory, QoS and Reliability

- Adrian Andronache, University of Luxembourg, Luxembourg
- Shingo Ata, Osaka City University, Japan
- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Michel Diaz, LAAS, France
- Michael Menth, University of Wuerzburg, Germany
- Michal Pioro, University of Warsaw, Poland
- Joel Rodrigues, University of Beira Interior, Portugal
- Zary Segall, University of Maryland, USA

Ubiquitous Systems and Technologies

- Sergey Balandin, Nokia, Finland
- Matthias Bohmer, Munster University of Applied Sciences, Germany
- David Esteban Ines, Nara Institute of Science and Technology, Japan
- Dominic Greenwood, Whitestein Technologies AG, Switzerland
- Arthur Herzog, Technische Universitat Darmstadt, Germany
- Malohat Ibrohimova, Delft University of Technology, The Netherlands
- Reinhard Klemm, Avaya Labs Research-Basking Ridge, USA
- Joseph A. Meloche, University of Wollongong, Australia
- Ali Miri, University of Ottawa, Canada
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Said Tazi, LAAS-CNRS, Universite Toulouse 1, France

Systems and Network Communications

- Eugen Borcoci, University 'Politehncia' Bucharest, Romania
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland
- Jan de Meer, smartspace®lab.eu GmbH, Germany
- Michel Diaz, LAAS, France
- Tarek El-Bawab, Jackson State University, USA
- Mario Freire, University of Beria Interior, Portugal / IEEE Portugal Chapter
- Sorin Georgescu, Ericsson Research - Montreal, Canada
- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, INRIA, France
- Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
- Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Reijo Savola, VTT, Finland

Future Internet

- Thomas Michal Bohnert, SAP Research, Switzerland
- Fernando Boronat, Integrated Management Coastal Research Institute, Spain

- Chin-Chen Chang, Feng Chia University - Chiayi, Taiwan
- Bill Grosky, University of Michigan-Dearborn, USA
- Sethuraman (Panch) Panchanathan, Arizona State University - Tempe, USA
- Wei Qu, Siemens Medical Solutions - Hoffman Estates, USA
- Thomas C. Schmidt, University of Applied Sciences – Hamburg, Germany

Challenges in Internet

- Olivier Audouin, Alcatel-Lucent Bell Labs - Nozay, France
- Eugen Borcoci, University “Politehnica” Bucharest, Romania
- Evangelos Kranakis, Carleton University, Canada
- Shawn McKee, University of Michigan, USA
- Yong Man Ro, Information and Communication University - Daejeon, South Korea
- Francis Rousseaux, IRCAM, France
- Zhichen Xu, Yahoo! Inc., USA

Advanced P2P Systems

- Nikos Antonopoulos, University of Surrey, UK
- Filip De Turck, Ghent University – IBBT, Belgium
- Anders Fongen, Norwegian Defence Research Establishment, Norway
- Stephen Jarvis, University of Warwick, UK
- Yevgeni Koucheryavy, Tampere University of Technology, Finland
- Maozhen Li, Brunel University, UK
- Jorge Sa Silva, University of Coimbra, Portugal
- Lisandro Zambenedetti Granville, Federal University of Rio Grande do Sul, Brazil

CONTENTS

Browsing Through OWL Domain Ontologies	184 - 194
Christian Kop, Alpen-Adria Universitaet Klagenfurt, Austria	
The Future of the Internet: Scenarios and Challenges in the Evolution Path as Seen by EIFFEL think-tank	195 - 202
Borka Jerman Blažič, Institute Jožef Stefan, University of Ljubljana, and University of Stockholm, Slovenia and Sweden	
The Impacts of the Digital Divide on Citizens' Intentions to Use Internet Voting	203 - 211
France Belanger, Virginia Tech, USA Lemuria Carter, North Carolina A&T, USA	
Free-Libre Open Source Software as a public policy choice	212 - 222
Thomas Margoni, University of Western Ontario, Canada Mark Perry, University of Western Ontario, Canada	
Lessons Learned on Enhancing Performance of Networking Applications by IP Tunneling through Active Networks	223 - 233
Tomas Koutny, University of West Bohemia, Czech Republic Jakub Sykora, University of West Bohemia, Czech Republic	
An Automated Framework for Mining Reviews from Blogosphere	234 - 244
Mehmet Aktas, Tubitak, Turkey	
Towards Effort Estimation for Web Service Compositions using Classification Matrix	245 - 260
Zheng Li, NICTA and UNSW, Australia Liam O'Brien, CSIRO and ANU, Australia	
Supporting Mobile Web Service Provisioning with Cloud Computing	261 - 273
Satish Srirama, University of Tartu, Estonia Vladimir Šor, University of Tartu, Estonia Eero Vainikko, University of Tartu, Estonia Matthias Jarke, RWTH Aachen University, Germany	
Developing Personalized Information Services for Mobile Commerce Location-Aware Applications	274 - 283
Christos Georgiadis, University of Macedonia, Greece	

Browsing Through OWL Domain Ontologies

Christian Kop

Applied Informatics

Alpen-Adria-Universitaet Klagenfurt

Klagenfurt, Austria

chris@ifit.uni-klu.ac.at

Abstract— There is a trend to see ontologies not as one big specification of a certain domain but to see it as a network of interrelated units. This paper will propose an approach and a tool, which allows the user to navigate through a network of ontology documents. The user will be supported by two different presentation techniques (graphical visualization and verbalization). Furthermore, s/he will be provided with information about relevant (key) classes and with a textual summary report of the ontology. Apart from the visualization of specific OWL domain ontologies, s/he will also get support for the visualization of the ontology network in which a domain ontology specification is just a unit, which is related to other units. Hence, a tool is introduced, which allows to present ontologies in different ways.

Keywords- semantic web; OWL; verbalization; summary; relevant classes; key classes; ontologies; visualization.

I. INTRODUCTION

With the Semantic Web Initiative [14], the Internet more and more becomes an Internet, which contains ontologies. Furthermore,, ontologies do not appear solely but are connected to each other. Such ontologies were mainly generated to be machine interpretable. But also a human reader must be able to browse through this network of ontologies. The tool, which is described here, does not provide mechanism for automatic reasoning but techniques to present the ontology to human readers. Especially within a network of ontologies it is necessary to give an overview of the ontology network and to provide a combination of visualization techniques for a node (ontology document) within this network. Therefore, one of the first changes to traditional tools is the visualization of an overview of the document network itself before visualizing the ontology inside a certain document. Furthermore,, it must be guaranteed that a user gets information, which node (document) s/he has successfully visited and which document is currently opened. To summarize, in this paper the following presentation strategies are proposed:

- Graphical representation of the import relationships between ontology documents

- Graphical representation of the user behavior (navigation visualization) within the document network.
- Graphical representation of the ontology specification for a specific ontology document.
- Verbalization of this specification
- A list of relevant classes
- Textual summary of the formal ontology specification.

Whereas other tools focus especially on either visualizing or verbalizing an OWL ontology itself, this approach focuses on the combination of these techniques. Furthermore,, it provides the possibility to show an OWL ontology as a network of interrelated OWL documents if the ontology imports concepts from other OWL documents. Therefore, it is possible to navigate through the interrelated OWL documents.

The ontology specification itself is presented on two additional levels. It is presented graphically or verbalized as a whole. If a user wants to see details on an OWL class then they are shown graphically and they are verbalized.

In order to describe such a tool and approach, the paper is structured as follows. In the next section, an overview of related work is given. Section III continues with the description of the document network visualization. Section IV starts with the first detailed view for a specific document node. This view is a graphical one. In Section V, a specific view is introduced, which lists all OWL classes together with the possibility to provide the user an understanding of the relevant classes of the ontology. This is based on measures, which are directly derived from the ontology structure. In this section also traditional views are introduced. The other important presentation strategy (verbalization) will be presented in detail in Section VI. Section VII combines the verbalization strategy and measures for relevant classes to introduce a view of a textual summary for an ontology. Section VIII gives a final discussion of the features of the presented tool. Section IX summarizes this approach and gives an outlook to future work.

II. RELATED WORK

Since the tool covers different techniques (e.g., graphical visualization, verbalization), the related work section is divided into subsections, which cover these aspects.

A. Visualization of Ontologies and Networked Ontologies

Concerning the research field of the visualization of an ontology specification, a lot of work has been done so far. Lanzenberger, Sampson and Rester summarized this work [16]. Beside traditional two-dimensional graphical views there is a trend to introduce 3D visualization techniques for large ontologies. The tools OntoViz [17], OWLViz [18] and Ontobroker [21] visualize ontologies with two-dimensional graphs. Tools like Ontorama [19] and OntoSphere 3D [20] provide three-dimensional representation techniques. Furthermore, visualization is not only used to present the ontology content but is also used to view specific aspects of the content. For instance the tool AlViz introduced by Lanzenberger et al. in [22] and [23] visualizes the alignment of two ontologies. The approaches described by Falconer et al. [24] and Gilson et al. [25] visualize the mapping of ontologies. Garcia et al. proposes how the coupling of ontologies can be visualized [26].

Ontologies are nowadays not seen as one big specification of a certain domain but they are seen as a network of interrelated units. The NEON Approach (www.neon-project.org) is a research project, which motivates networked ontologies. Here it is even proposed that ontologies should be splitted into modules [27]. Currently however, an ontology document itself is still the most used container, which can be treated as a bigger unit. The approach described in this paper therefore follows this traditional view of modularization. OWL classes and object properties, which belong together are specified in the same document. If classes and object properties are needed from another OWL document, then this external document is imported. In this sense, an ontology documents network is established. In order to browse such a network, browsing must start at the level of OWL documents.

B. Verbalization

Strategies to verbalize ontologies are described in Fuchs et al. [5] and Hewlett et al. [6]. These approaches mainly focus on an optimal natural language verbalization of OWL constructs itself like *subClassOf*, *intersectionOf*, *unionOf*, cardinality restrictions etc. Although, this helps to read the whole ontology it still looks artificial since the labels of OWL classes and OWL object properties are not transcribed. In their work on verbalization of OWL 1.1., Karljurand and Fuchs still conclude that object property and class labels hopefully will become more English-like over time. Luckily an analysis of many OWL ontologies made by Mellish and Sun showed that in many cases nouns for classes and verbs for object properties are used [11]. Although object property labels contain a verb many name variations can exist. Therefore, in [4] linguistically

motivated guidelines and naming conventions were proposed, which must be used for object property labels and class labels. Information of individuals is verbalized in the research work of Bontcheva [3].

C. Relevant Class Measures

Interesting related work for this case was done by Bezerra et al. [2] and Huang et al. [7]. They introduced key classes, which are similar to relevant classes in order to estimate the quality of an ontology. The weighted relationships of a class are counted in [7]. The weight is determined from the relationships that can be inherited from the super classes. Then these weights are forwarded to the involved classes of these relationships. In [2], the number of direct children is counted. Vrana and Mach propose to return a vector of terms instead of the whole ontology [30]. This vector can be seen as a summary, which is based on the input of a user who searches for keywords in ontologies. More detailed research on key concepts are provided by Zhang et al. [31] and Peroni et al. [29]. Zhang et al. summarizes RDF ontologies. It returns a graph of salient RDF sentences. The approach proposed by Peroni et al. integrates topological measures (density and coverage of concepts) with statistical measures (popularity of a concept) as well as cognitive criteria (natural categories). Topological measures rely on the structure of the ontology. The included cognitive criteria is based on the idea that simple single words represent more likely key concepts. However it turned out, that this criteria does not work so well. The popularity measure is estimated by counting the hits of several ontology concepts in Yahoo. Then the hits are compared. An ontology concept, which has more hits than another is seen as more popular. This measure was introduced to optimize the output with regard to human experts. The authors in [29] found out, that human experts tend to prefer more common knowledge terms rather than special terms. In cases where common knowledge terms and special terms are equal candidates for key concepts, they wanted to ensure that the common knowledge term is preferred in the selection process by the algorithm. Centered concepts play an important role for estimating clusters in conceptual models (see the work of Moody et al.) [12] [13]. Since conceptual models depend on relationships between entity types (classes), for each entity type its relationship is counted.

D. Differences to Related Work

Although there is a lot of work, which focuses on graphical representation of ontologies, a visualization of interrelated ontology documents was not found in literature. This approach therefore not only presents a visualization of the ontology itself but also focuses on the visualization of the import relationships between OWL documents. In fact, this is the starting point of the browser. In this feature, the user can enter the link to an ontology document. Then s/he is provided with the document itself and all the other OWL

documents, which are needed for “imports” in the chosen OWL document. From this first view s/he can start to navigate and explore the OWL document network. A more detailed description of how this can be done, will be presented in Section III. Of course s/he can let the tool present the content of each document. This is done graphically or by using verbalization. Actually, a two-dimensional graphical representation is provided for the graphical representation of the ontology content.

The verbalization strategy proposed here is a refinement of [4]. It allows some more labeling freedom, though some linguistic guidelines are still needed to achieve good verbalization results. Finally this approach differs from the approach described in [3] since this approach verbalizes OWL classes and not individuals.

The determination of relevant classes in this approach relies on the ontology structure itself. A class is only a relevant (key) class if this can be automatically determined from the content of the ontology itself. A popularity measure is not used in this approach since it was the aim that only the structure and no external resource should be used. Whereas in [29] only the best N key concepts are returned, this approach does not select only N concepts but categorizes concepts into relevant (= key) concepts and considerable concepts. Hence, beside an inner circle of concepts it also has an outer circle of considerable concepts, which could be candidates too. Furthermore, in one view of the tool it is also possible to order all the concepts according to its relevance. In this view nothing gets lost. As an alternative to the number of children [2], the number of all successors is calculated. This alternative achieves that the relevance is treated globally since the whole sub tree and not only the children are considered. The calculation of the weight for relationships as described in [7] is not taken as the only measure, since it only works if classes have object properties. However, OWL classes do not necessarily need such object property relationships. In fact, there are also ontologies, which only consist of a class/subclass taxonomy. If object properties are specified in an ontology, then it will be described that simpler measures are sufficient. The paper introduces several other statistical measures for classes (e.g., number of instances, number of restriction and disjoints). Instead of the RDF basis [31] it focuses on specific OWL features.

All verbalization strategies today verbalize the whole ontology content. On one hand it is an advantage since the human reader does not have to understand the formal and artificial constructs of the ontology. On the other hand however, it is still necessary to read the whole ontology. To use a metaphor, this can be compared with presenting the reader a full “news paper article”, though at the beginning, it might be better to present him only an abstract or the headline (textual summary).

Regarding textual summary generation, it can be said, no approach is known, which currently generates a textual summary out of the ontology structure. According to an

evaluation made in [1] a combination of two measures out of the measures for relevant class determination were used as a basis for textual summary generation.

Finally, in this paper it is proposed to combine all the above mentioned representation techniques together in one tool.

III. MAP OF INTERRELATED OWL DOCUMENTS

Each OWL document is related with other OWL documents from which it imports necessary other specifications. For instance, the wine ontology imports PotableLiquid from the food ontology. This resource is needed to specify that a Wine is a subclass of PotableLiquid. Another example: The climaticzone ontology at the designpatterns web site needs the ontologies aquatic resource, cpannotationschema and observation. The tool visualizes this information.

A. Visualization of the Document Network

Viewing such a network is done, using the header of the OWL specification in which the imports are specified. Instead of presenting the whole ontology it firstly scans the document for the section with the import specifications.

It then paints the document itself as a node (rectangle). If the document imports concepts from other documents then these import relationships are visualized as directed edges pointing from the document, which needs the import to the document, which provides concepts for the import. Furthermore, the system does not show the user the whole network at once. Instead it starts with the chosen OWL document and its direct neighbors (i.e., those documents, which are needed for imports). If the user wants to see more of the network, s/he has to expand those documents, in which s/he is interested in. This stepwise visualization document network has the following reason: It cannot be trusted that all the nodes in the network are available since the nodes (OWL documents) might be spread over different servers on the Web. Some servers might be not available. Therefore the system firstly only provides the start node and the direct neighbors of the start node (document). This is possible since the start node is available and the specification of the imports is given inside the start document. Figure 1 shows the situation at the beginning for the “Climatic Zone” ontology, which was taken from the ontology pattern web side (see: [32]).

B. Operations Provided in the Document Network

If the user selects one document s/he can open a popup menu with the right mouse button. Then s/he gets menu options for operations, which can be applied on the selected document. Namely, it is possible to

- Open a detailed graphical view containing the ontology (for more details see Section IV)
- Open a list view where all the ontology elements are listed (see Section V) and verbalized (Section VI).

- Open a summary view where a natural language summary is generated (see Section VII)
- Get all the import details for all the needed imports of the document. This operation is also available for the import edges between the OWL documents.
- Expand a node. This operation supports the navigation through the network (see Figure 2).

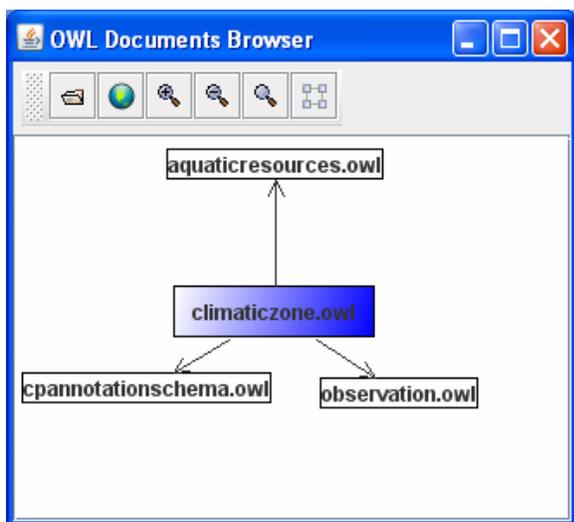


Figure 1. Relationships between OWL documents

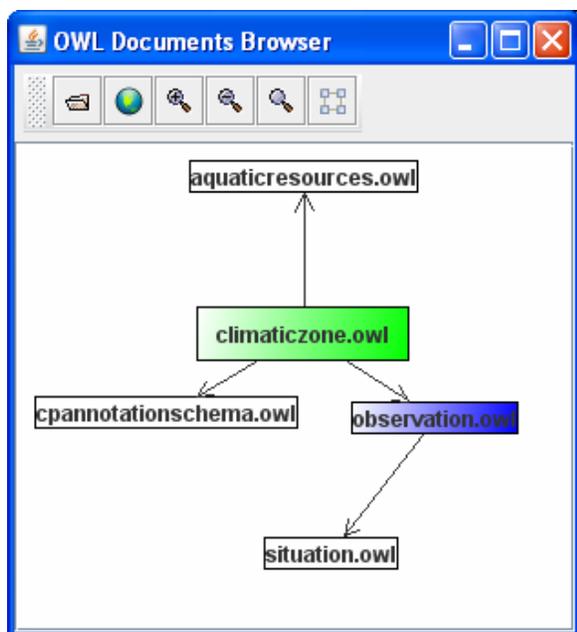


Figure 2. Expansion of observation.owl

C. Representation of Imports

The representation of imports has the same structure for both the OWL documents and the import edges. In both cases, the user gets a list of OWL elements and the resources (imported documents), from which they were imported. Only the resulting list is different. If the user chooses this list for an OWL document, then a list of elements from all the surrounding neighbors of that OWL document is provided (these neighbors are the documents, from which elements are imported). If s/he selects an edge then s/he only gets the list of elements from the document to which the edge points (the imported document). Figure 3 gives an example for the edge pointing from climaticzone.owl to observation.owl.

Imported Elements		Verbalization of Imports		
Type	Name	Relationship	Resource	Imported from
Class	AquaticResource	owl:disjointWith	Parameter	http://www.onto...
Class	AquaticResource	owl:disjointWith	Observation	http://www.onto...
ObjectProperty	hasClimaticZone	rdfs:subPropertyOf	hasParameter	http://www.onto...
ObjectProperty	isClimaticZoneOf	rdfs:subPropertyOf	isParameterOf	http://www.onto...
ObjectProperty	isResourceOf	rdfs:subPropertyOf	hasObservation	http://www.onto...
ObjectProperty	hasResource	rdfs:subPropertyOf	isObservationOf	http://www.onto...

Figure 3. Visualization of imports

The first column specifies the type of the OWL element (e.g., is it an OWL class or an object property), which needs a resource from another OWL document. The second column gives the name of the element. The third column names the kind of “relationship” to the imported element. The column resource gives the name of the imported element. The last column specifies the resource locator.

There was also an attempt here to verbalize this kind of information (second tab in the screenshot of Figure 3).

D. Visualization of the Navigation

Beside the visualization of the document network structure, also the behavior of the user is visualized. First of all, the document with which the user starts appears larger than the other documents (see also Figure 1).

For traditional web sites it is already an established strategy that links, which have not been followed yet appear differently to links, which were already opened by the user. Since this strategy has proven to be very successful, it was adopted for the navigation visualization. A coloring system was introduced to show

- Where the user currently is and executes an operation (e.g., the node s/he opens to browse through the OWL specification),
- Which nodes has been opened successfully in the past and
- Which nodes (documents) are not available (i.e., the server was down and hence it was not possible to open this specific ontology).

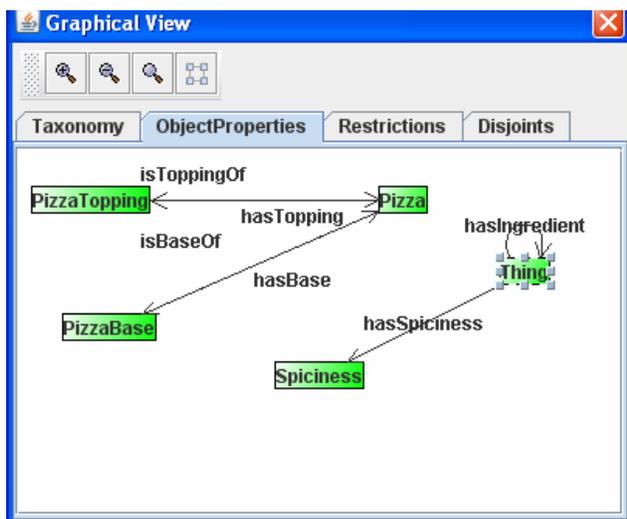


Figure 5. Object property view

C. Restrictions and Disjoints View

The last two tabs contain views on restrictions (namely: someValuesFrom, allValuesFrom, hasValue) and disjoints. The restrictions as well as the disjoints are presented as dashed lines between the concepts. Furthermore, restrictions can be distinguished by different colors and a different edge ending at the position of the class for which the restriction was defined. A solid circle represents an allValuesFrom restriction. A non solid circle represents a someValuesFrom restriction and no circle is used for a hasValue restriction. (see Figure 6 for graphical visualization of allValuesFrom and someValuesFrom restrictions).

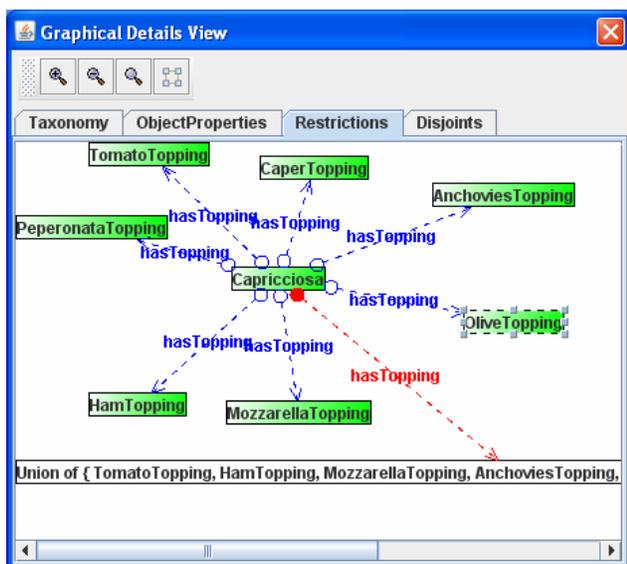


Figure 6. Restrictions view for OWL class Capricciosa

D. OWL Class Graphical Details View

The graphical details view for one OWL class has the same representation features as the graphical view for all OWL elements. The difference is, that the representation of the taxonomy, object property, restrictions and disjoints is shown for one selected OWL class. Figure 6 shows the restrictions for the selected OWL class “Capricciosa” in the pizza ontology.

V. RELEVANT CLASSES AND OTHER TABULAR REPRESENTATIONS

Beside the graphical views, the tool provides the human reader also with the following tabular and verbalized views: the OWL classes together with their relevance, a class taxonomy tree view, instances, restrictions and disjoints.

A. Relevant Classes

Several kinds of measures to sort OWL classes for their relevance are introduced in this paper.

- Weighted number of successors (wNS),
- Number of object properties (P),
- Weighted number of object properties (wP),
- Number of children (NC),
- Number of restrictions and disjoints (R/D),
- Instances of an OWL class (I),
- Total value (T).

The weighted number of successors (wNS) focuses on the subclass hierarchy. The successors (NS) of a class are counted and weighted. This avoids that OWL classes at the top of the hierarchy will always be the winners.

For the weighting factor, the distance of a certain class from the root *Thing* is taken (= dfr) and it is divided by the maximum distance to a leaf (= maxdtl). The distances are calculated by counting the edges from the root to the leafs respectively. The weighted number of successors for a class X hence is calculated as:

$$wNS(X) = NS * (dfr / maxdtl) \tag{1}$$

In the pizza ontology the distance from root to pizza is 2 (i.e., *Thing* → *DomainConcept* → *Pizza*). The maximum distance to a leaf is also 2. The weighting factor therefore is 2 / 2 = 1. Pizza itself has 34 successors. Hence, wNS(pizza) is 34.

The number of object properties for a domain class (P) considers the object property relationships from a domain class to its range class. It is assumed that the domain class is more likely a relevant class than the range class in the object property. Therefore each object property is counted only for domain classes. For example in the pizza ontology the class *Pizza* gets the value 2 for P since pizza is the domain class in two object properties (*pizza has pizza topping* and *pizza has pizza base*).

The weighted number of object properties (wP) is a refined version of the measure P. Instead of incrementing the counter by 1 for each class, which is involved as a

domain in an object property, the counter is incremented by the wNS value of the range. However, if the value is 0 then wP is incremented by 1 and it degrades to P. With this strategy the importance (weight) of the range is also forwarded to the domain.

Instances of an OWL class (I) are a third strategy to estimate the relevance of a class. In the pizza ontology the class *Country* has 5 instances (I). The number of children was taken from [2] and just counts the direct children of a class.

The number of restrictions and disjoints are calculated by incrementing a counter for an OWL class whenever a restriction (someValuesFrom, allValuesFrom, hasValue) or disjoint is specified for that class.

The default value of T is a combination of the wNS and wP measure. It is not a simple count but it divides the counted result into an ordinal value system ranging from 1 to 3. A more detailed description of the determination of these values will be given in Section VII. The value for T can also be set to be the sum of any selection of the measures wNS, NC, wP, P, R/D, I. Figure 7 shows how relevant classes are visualized in a listing. The elements in the listing can be sorted according to each of the measures.

B. Other Visualizations

Beside the visualization of relevant classes, the user can also choose among the following other tabular and J-Tree-based visualizations (object properties, class tree, class details, restrictions, disjoints and instances).

The tabular object property representation lists object properties. The table has three columns: object property column, domain and range column. The object properties together with their domain and range are verbalized. A J-Tree based representation is provided for the class taxonomy and the details for a class. In the J-Tree for the class taxonomy, each class and their subclasses are visualized as folders and subfolders. If a class has instances, also the instances are presented as leaves in the J-Tree. In the J-Tree for the details of all classes, the classes once again appear as folders. If such a class folder is opened, the sub folders contain the restrictions, instances, object properties sub classes etc., for the selected class.

The restriction are listed also in a tabular representation. This listing has three columns. One column is for the class on which the restriction is defined. The other column is for the restriction category and the third column for the specification of the restriction. The content of this last column is once again verbalized.

The instance list is another tabular representation, which focuses on the instances. It has two columns. The first column contains the instances. The second column contains the classes to which the instances belong.

In each of the tabular representations it is possible to sort by each column individually. For instance, object property specifications can be sorted either by their domain or range.

Class Name	wP	NC	P	I	R/D	T
Named Pizza	69	0	23	0	0	3
Pizza	34	29	11	2	0	4
Pizza Topping	25	34	10	1	0	3
Vegetable Topping	22	0	14	0	0	7
Cheese Topping	18	0	6	0	0	7
Domain Concept	18	0	5	0	0	1
Pepper Topping	10	0	4	0	0	12
Fish Topping	9	0	3	0	0	8
Tomato Topping	8	0	2	0	0	13
Meat Topping	7	0	4	0	0	7
Herb Spice Topping	6	0	2	0	0	7
Spiciness	6	0	3	0	0	0
Green Pepper Topping	5	0	1	0	0	3
Ham Topping	4	0	1	0	0	3
Onion Topping	4	0	1	0	0	13

Figure 7. Relevant classes list

C. OWL Class Details

For presenting details of an OWL class, most of the views in Section B can be reused. Only the relevant classes view and the taxonomy view make no sense in this context. The views on restrictions, disjoints, class details, instances and object properties can be applied for the selected class since only information is listed, which belongs to that class.

VI. VERBALIZATION OF AN ONTOLOGY

In the previous section, the listings presented also verbalized information. Therefore, this section explains the details.

A. Problems of Verbalization

In the verbalization step the following two problems are well known and discussed in literature [4],[5],[7],[9],[11]:

1. How can the formal knowledge representation be transformed into a natural language representation?
2. How can OWL class and object property labels be transformed into a natural language representation?

The first problem addresses the different OWL elements and constructs to define formal expressions (e.g., *Intersection*, *Union*, *someValuesFrom*, *allValuesFrom* etc.). Previous research work already provided solutions for that problem.

The second problem occurs since the human OWL designers cannot be influenced to use proper and standardized labels for classes and object properties. Many variations exist. In [4], the verbalization strategy recommends that classes and object properties are labeled exactly in a specific format. Namely, classes should be

nouns in singular. If there are compound nouns, then they must be separated using an upper character (e.g., *VegetarianPizza*) object properties must start with a verb. If a preposition is needed, then the preposition must be explicitly specified in the label. Once again single words must be separated by an upper character (*workWith*). The language used for all the labels must be English.

B. Verbalization of Classes and ObjectProperties,

This approach addresses the second problem and refines the work in [4] and [6]. In order to produce better results and give ontology designers more freedom of defining the labels, the following must be done:

1. Find the separation strategy and separate accordingly.
2. Remove redundant information in the object property labels.
3. Add additional information to the object property.
4. Add the article “a” or “an” respectively to the two involved classes of a relationship.

Most ontology engineers prefer to separate words by using upper case characters (e.g., *VegetarianPizza*, *worksWith* etc.). Some also use separators like “_” or “-” (e.g., *vegetarian_pizza*). If the separator strategy is detected, the words can be separated to their natural language form (e.g., “*vegetarian pizza*“, “*works with*” etc.).

If ontology engineers only would use verbs in object property labels the verbalization would be much easier. Unfortunately, they often add information about the domain and range to the object property itself. This is often necessary to avoid different object properties with the same labels (i.e., several object properties with the label “*has*”) but it produces artificial results in those verbalization approaches, which do not handle this. Particularly if the range (domain) or closing parts of the range (domain) are also mentioned in the object property label, then this must be removed (e.g., “*has*” instead of “*hasTopping*”). The verbalization result is much better if such changes on the object property labels are made (e.g., “*Pizza has Pizza Topping*” instead of “*Pizza has topping Pizza Topping*”).

Additional information is necessary if there is an object property, which contains the verb *has* together with a noun but the noun is not part of the range (e.g., *person has father man*). This is verbalized to *person has a father, which is a man*. However, this strategy fails, if the range class is not a noun. In some of the ontologies adjectives were specified as range concepts in object properties (e.g., *Wine has color Green*). In this case, only an additional lexicon with adjectives can help to avoid a wrong verbalization. For instance this can be solved as proposed by Sugumaran et al. [28]. There WordNet was taken as the lexicon. A word like “*mild*” can be searched in the lexicon. The lexicon returns all the meanings of “*mild*” together with the word category of this meaning and a measure how often the meaning is used. If the word category of the word with the most used

meaning is an adjective, then this can be considered for verbalization.

Finally, if the domain and the range of the object property are common nouns then, to each of the two classes involved, the article “a” or “an” respectively is added. If an adjective or a mass noun was used to label an OWL class, then no article must be added.

With these strategies of verbalizing labels, the ontology designer has much more flexibility to name the object properties and OWL classes. (see Figure 8)

Object Property	Domain	Range
A Pizza Base is base of a Pizza	Pizza Base	Pizza
Anything has a Spiciness	Anything	Spiciness
A Pizza has a Pizza Base	Pizza	Pizza Base
A Pizza Topping is topping of a Pi...	Pizza Topping	Pizza
Anything has an Ingredient which ...	Anything	Anything
Anything is ingredient of Anything	Anything	Anything
Anything has a Country Of Origin ...	Anything	Anything
A Pizza has a Pizza Topping	Pizza	Pizza Topping

Figure 8: object property verbalization

C. Verbalization of Restrictions and Disjoints

The verbalization of the restrictions and disjoints reuses the strategies for OWL class and Object Property verbalization. Currently it is based on patterns found in the OWL ontology. The pattern:

```
<owl:Class rdf:ID="Anjou">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        rdf:resource="#hasColor"/>
      <owl:hasValue rdf:resource="#Rose"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

is verbalized to “*Anjou has a Color, which is (has value) Rose*”. Particularly, if within the class description of Anjou a HasValue-Restriction of on the Object Property is found then it is transcribed as shown above.

If a AllValue-Restriction together with a Union operator is found within an OWL class specification as shown below,

```
<owl:Class rdf:about="#AmericanHot">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty>
          rdf:about="#hasTopping"/>
        </owl:onProperty>
```

```

<owl:allValuesFrom>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class
        rdf:ID="HotGreenPepperTopping"/>
      <owl:Class
        rdf:about="#JalapenoPepperTopping"/>
      <owl:Class
        rdf:about="#MozzarellaTopping"/>
      <owl:Class
        rdf:about="#PeperoniSausageTopping"/>
      <owl:Class
        rdf:about="#TomatoTopping"/>
    </owl:unionOf>
  </owl:Class>
</owl:allValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

then it is verbalized to

"An American Hot has a Topping, which is (all values from) the Union of { Hot Green Pepper Topping, Jalapeno Pepper Topping, Mozzarella Topping, Peperoni Sausage Topping, Tomato Topping }"

Disjoint specifications like

```

<owl:Class rdf:about="#AmericanHot">
  <owl:disjointWith>
    <owl:Class rdf:about="#Napoletana"/>
  </owl:disjointWith>
</owl:Class>

```

are verbalized as *"An American Hot is disjoint with a Napoletana"*.

D. Verbalization of Imports

In Section III it was mentioned that the list of imports is also verbalized. This is done by using sentence templates in which the specific import information (e.g., imported element, element that refers to an imported element etc.) is inserted. The template looks as follows: *The (Class / ObjectProperty) <name> (is disjoint with / is a Subproperty of) is a sub class of / ... the Resource <name>, which is imported from <import address>.*

VII. TEXTUAL ONTOLOGY SUMMARY

As pointed out in [1] a textual summary of an ontology is useful if a human reader just wants to get a first impression of the specification. Particularly, it can be more compared with a situation you will find if people read news paper articles. Many people look at the title, headlines or the short abstract before they start reading the article itself. Hence, what is needed is a technique that automatically summarizes the relevant content for the human reader. In other words, such a tool must be able to generate and verbalize a subset of the ontology structure that can be seen as an abstract, summary

or headline of the ontology. Only if the ontology is verbalized and the information about the relevant OWL classes is summarized, then the human reader can derive his first impression quickly and efficiently. A summary or abstract of an OWL ontology can be generated by combining the determination of relevant (key) classes with verbalization strategies. Particularly, the summary generation consists of the following steps:

1. Calculation of class relevance based on measures,
2. Categorization if classes are relevant or not,
3. Verbalization of classes and their object properties.

The Step 1 was already described in Section V. Step 3 was described in more detail in Section VI. Therefore this section concentrates on Step 2. In [1], the combination of the weighted number of successors and the weighted number of object properties were seen as a good basis for textual summary generation.

Therefore, once each class got a measured value for wNS and wP (step 1) they are divided into the three categories.

- Relevant class (must be in the summary)
- Considerable class (can be in the summary)
- Not relevant class (not included in the summary)

This is done for wNS and wP separately. In order to categorize it is necessary to know the maximum measured value for wNS (wP respectively) for a certain class. This maximum is taken as the basis. It represents 100 % of the reachable measured values for wNS (wP). The wNS (wP) values for all other classes are now compared with the maximum (= 100 %). A percentage for these values is calculated. If a class reaches a percentage higher than 65% then it is treated as a relevant class for a summary. If the calculated percentage is between 33 % and 65 % then it is a considerable class. Otherwise it is not relevant.

The class with the maximum wNS in the pizza example is *NamedPizza* (wNS = 69). It is followed by *Pizza* (wNS = 34), *PizzaTopping* (wNS = 25). With these values for wNS, *NamedPizza* is a relevant class. *Pizza* and *PizzaTopping* can be seen as considerable classes since they do not have a wNS value higher than 45 (65 %). The other classes are not relevant at all according to their wNS values.

Applying the wP measure, the classes with the maximum wP value 34 are *PizzaTopping* and *PizzaBase*. Both classes have this value since they are involved in an object property as domains where the range is *Pizza*. *Pizza* follows with a wP value of 29. Here all three classes are relevant classes.

With the two measures (wNS, wP) and the distinction between relevant and considerable classes, it can be parameterized how restrictive the generated summary is. The summary can be generated on the basis of one of the measures only (i.e., either wNS or wP) or based on the combination of the two measures together. The latter can be understood as a union of wNS and wP. Furthermore, it can be decided if only relevant classes will appear in the

summary or relevant and considerable classes will be listed. The next table outlines the several possible alternatives.

TABLE I: ALTERNATIVES FOR SUMMARY REPORT

	relevant only	relevant + considerable
wNS	(1)	(2)
wP	(3)	(4)
wNS + wP	(5)	(6)

In the following the combination of wNS + wP as a basis for listening relevant and considerable classes (6) is described in detail. It contains the other alternative reports (1) – (5) and thus is the most general report.

Hence, if the alternative (6) is taken, then the summary report would not only return verbalizations of relevant classes but also verbalization of considerable classes. That means:

- A class is a relevant class if it can be categorized as a relevant class on the basis of at least one of the measures (wNS, wP).
- A class is a considerable class if it is not a relevant class and it is categorized as a considerable class by at least one of the measures (wNS, wP).

If now the verbalization strategies (Section VI) are reused, then a summary can be generated by an introduction template. “The most relevant classes are <list of relevant classes> followed by <list of considerable classes>”. If only the relevant classes are needed (alternatives (1), (3), (5) in Table VII) then no considerable classes are mentioned in the generated output.

Afterwards each of the listed classes is verbalized. Particularly, its relationship to its more general classes is named (e.g., a Pizza is a Domain Concept). Furthermore,, verbalization of those object properties is given where the relevant (considerable) class is involved explicitly as a domain (e.g., a Pizza has a Pizza Topping). Figure 9 shows a textual summary report for the Pizza ontology.

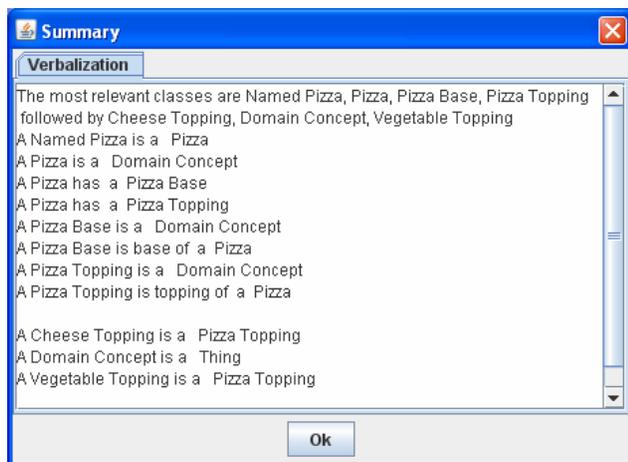


Figure 9. Textual summary

VIII. DISCUSSION

The ontology documents browser was tested with ontologies on the website [32] and with the ontologies mentioned in [1] and found in [15]. Especially in the case of the ontology design pattern web site, it turned out that it is very helpful if somebody wants to get an overview of the surrounding of a specific ontology (e.g., “climaticzone.owl”). The tool has the advantage to derive this information directly from the content of the chosen ontology. Then it visualizes the information graphically with the chosen ontology in the center and the ontologies needed for import as satellites of this ontology. Furthermore,, the tool provides him with navigation visualization. The starting point (starting document) is always presented larger than other documents. Successfully visited nodes (documents) are colored differently to nodes, which were not visited or nodes, which could not be visited (i.e., web server is down). Thus, the user gets a spatial overview of his navigation. S/he can even browse through the documents without looking inside one document. The users always have the spatial overview of the documents network.

In [1] the calculation of relevant classes was tested. It could be shown, that the calculation of relevant classes reflects the content structure of the ontology. It was therefore also “natural” to use these classes for the textual summary of the ontology content.

Regarding the verbalization, this paper focused on verbalization of labels. For certain patterns found in the OWL specification, verbalization patterns can be presented to the user. The verbalization process was also more sensible to word categories (e.g., nouns, verbs, adjectives). If an adjective is detected within an object property or within a restriction then the verbalization procedure considers this fact. This technique produces a better verbalization output.

Graphical visualization was included in this paper since it was the intention to show that a representation of an ontology must be a mix of different visualization techniques. It must be left to the user to choose the right representation for a certain situation.

IX. CONCLUSION AND FUTURE WORK

In this paper, a visualization approach and its implementation was described. In order to browse a network of ontologies it was proposed that different visualization strategies must be combined together. It is also necessary to provide the user with a view of interrelated documents and to help him during his navigation through the ontology document network. The prototype was implemented in Java. The Jena API was used for OWL ontology parsing. For the graphical visualization JGraph (www.jgraph.com) was used.

The intention also leaves many future research and implementation issues. First of all, it could be interesting to integrate also three-dimensional visualization views. Since there are also a lot of documents in other formats (e.g.,

RDF) it would be good to extend the functionality in such a way that also RDF documents are visualized. Finally an integration of this approach to Protégé is worth to look at. For users, which are more familiar with mathematical expressions and logic, a detailed view of the logical statements expressed in OWL could be interesting

REFERENCES

- [1] Ch. Kop, "How to Summarize an OWL Domain Ontology," ICDS 2010, Proc. of the International Conf. of the Digital Society, IEEE Computer Society Press, 2010, pp. 106 – 111.
- [2] D. Bezerra, A. Costa, and K. Okada, "SwTOI (Software Test Ontology Integrated) and its applicaton in Linux Test," Proc. of the 3rd Int. Workshop on Ontology, Epistemology and Conceptualization for Information Systems, Software Engineering and Service Science, CEUR-WS, Vol 460, 2009, <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [3] K. Bontcheva, "Generating Textual Summaries from Ontologies," Proc. of the Second European Semantic Web Conference, ESWC 2005, Lecture Notes in Computer Science (LNCS), Vol. 3532, Springer Verlag, Berlin, Heidelberg, 2005, pp. 531 – 545.
- [4] G. Fliedl, C. Kop, and J. Voehringer "From OWL class and property labels to human understandable natural language," Proc. of the 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, Lecture Notes in Computer Science (LNCS), Vol. 4592, Springer Verlag, 2007, pp. 156 – 167.
- [5] N. E. Fuchs, S. Höfler, K. Kaljurand, F. Rinaldi, and G. Schneider, „Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines," Reasoning Web, First International Summer School, LNCS 3564, Springer, 2005, pp. 213-250.
- [6] D. Hewlett, A. Kalyanpur, V. Kolovski, and C. Halaschek-Wiener, „Effective Natural Language Paraphrasing of Ontologies on the Semantic Web," End User Semantic Web Interaction Workshop, CEUR-WS Proceedings, Vol. 172, 2005, <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [7] N. Huang and Sh. Diao, "Structure-Based Ontology Evaluation," IEEE International Conference on e-Business Engineering (ICEBE06), 2006, pp. 1- 6.
- [8] K. Kaljurand and N.E. Fuchs, "Verbalizing OWL in Attempto controlled English," Proc. of the 3rd Int. Workshop on OWL Experiences and Directions (OWLED2007), CEUR-WS Proceedings, Vol. 258, 2007, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [9] Ch. Kop, "What are main concepts in an OWL domain ontology," 1st Int. Conference on Knowledge Engineering and Ontology Development, INSTICC Proceedings, Funchal, 2009, pp. 404 – 407.
- [10] D.L. McGuinness and F. van Harmelen., "OWL Web Ontology Language Overview," <http://www.w3.org/TR/owl-features/>, 2004, 19.01.2011.
- [11] C. Mellish and X. Sun, "The Semantic Web as a Linguistic Resource: Opportunities for Natural Language Generation," Knowledge Based Systems Vol. 19, 2006, pp. 298-303.
- [12] D.L. Moody, "Entity Connectivity vs. Hierarchical Levelling as a Basis for Data Model Clustering: An Experimental Analysis," DEXA 2003 Proceedings, Lecture Notes in Computer Science (LNCS), Vol. 2736, Springer Verlag, Berlin, Heidelberg, 2003, pp. 77-87.
- [13] D.L. Moody and A. Flitman, "A Methodology for Clustering Entity Relationship Models – A Human Information Processing Approach," Proc. of Conceptual Modeling (ER 1999), Lecture Notes in Computer Science (LNCS), Vol. 1728, Springer Verlag, Berlin, Heidelberg, 1999, pp. 114-130.
- [14] N. Shadbolt, W. Hall, and T. Berners-Lee, "The Semantic Web Revisited," IEEE Intelligent Systems Journal, May/June 2006, pp. 96-101.
- [15] <http://krono.act.uji.es/Links/ontologies/>, 19.01.2011
- [16] M. Lanzemberger, J. Sampson, and M. Rester, "Visualization in Ontology Tools," International Conference on Complex, Intelligent and Software Intensive Systems. IEEE Computer Society Press, 2009, pp. 705 – 711.
- [17] OntoViz: Visualizing protégé ontologies, <http://protegewiki.stanford.edu/index.php/OntoViz>, 15.08.2011
- [18] M. Horridge, OWLViz <http://www.co-ode.org/downloads/owlviz/OWLvizGuide.pdf>, 19.01.2011
- [19] P. Eklund, N. Roberts, and S.Green "Ontorama: Browsing RDF Ontologies using a hyperbolic-style browser," Proc. of 1st Intl. Symposium on CyberWorlds, 2002, pp. 405 – 411.
- [20] A. Bosca and D. Bonino, "Ontosphere: more than a 3d ontology visualization tool," SWAP 2005 the 2nd Italian Semantic Web Workshop, 2005, <http://ceur-ws.org/Vol-166/70.pdf>, 19.01.2011
- [21] S. Decker, M. Erdmann, D. Fensel, and R. Studer "Ontobroker: Ontology based access to distributed and semi structured information," Database Semantics: Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, 1999, pp. 351 – 369.
- [22] M. Lanzemberger, S. Miksch, and M. Pohl, "The stardines – visualizing highly structured data," Proc. of the International Conf. on Information Visualization, IEEE Computer Society Press, 2003, pp. 47 – 52.
- [23] M. Lanzemberger and J. Sampson, "AIViz – A Tool for Visual Ontology Alignment," Proceedings of the Information Visualization (IV'06), IEEE Computer Society Press, 2006, pp. 430 – 440.
- [24] S.M. Falconer, R. I. Bull, L. Grammel, and M-A. Storey, "Creating visualizations through ontology mapping," International Conference on Complex, Intelligent and Software Intensive Systems, IEEE, 2009, pp.688-693
- [25] O. Gilson, N. Silva, P.W. Grant, and M. Chen, "From Web Data to Visualization via Ontology Mapping," Eurographics /IEEE-VGTC Symposium on Visualization , Vol. 27, No. 3, 2008. pp. 959-966.
- [26] J. Garcia, F. Garcia, R. Theron, "Visualising semantic coupling among entities in an OWL ontology," appears in Proc of the 4th International Workshop on Ontology, Empistemology and Conceptualization for Information Systems, Software Engineering and Service Science, Springer Verlag, Vol. 62.
- [27] C. Bezerra, F. Freitas, J. Euzenat, A. Zimmermann, "ModOnto: A tool for modularizing ontologies," CEUR-WS, Vol. 427, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-427/paper3.pdf>, 19.01.2011
- [28] V. Sugumaran, S. Puroa, V.C. Storey, and J. Conesa, "On-Demand Extraction of Domain Concepts and Relationships from Social Tagging Websites," Proc. of the 15th Int. Conference on Natural Language Processing and Information Systems, Lecture Notes in Computer Science (LNCS), Vol. 6177, Springer Verlag, 2010 pp. 224 – 232.
- [29] S. Peroni, E. Motta, and M. d'Aquin, "Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures," Proc. of the 3rd Asian Semantic Web Conference on the Semantic Web (ASWC'08), Lecture Notes in Computer Science (LNCS), Vol. 5367, Springer Verlag, 2008, pp. 227 – 241.
- [30] J. Vrana and M. Mach, "Ontology concepts interpretation," 8th International Symposium on Applied Machine Intelligence and Informatics, IEEE Press, 2010, pp. 215 – 219.
- [31] X. Zhang, G. Cheng, and Y. Qu, "Ontology Summarization Based on RDF Sentence Graph," Proceedings of the 16th International Conference on World Wide Web (WWW 2007), ACM, NY, , 2007, pp. 707 – 715, doi 10.1145/1242572.1242668.
- [32] www.ontologydesignpatterns.org, 19.01.2011

The Future of the Internet: Scenarios and Challenges in the Evolution Path as Seen by EIFFEL Think-Tank

Broka Jerman-Blažič

Jožef Stefan Institute

University of Ljubljana, Faculty of Economics,

Ljubljana, Slovenia

University of Stockholm, Computer and System Sciences Department,

Stockholm, Sweden

e.mail: borka@e5.ijs.s

Abstract-The paper deals with the current findings of the Think Tank group of experts from all over the world working within EU FP7 project “Evolving Future Internet for European Leadership – EIFFEL”. The Think-Tank is working on meetings where the discussion about major challenges and scenarios of the current Internet is contributing to the debate about the future of the Internet trying to provide some answers and identify evolutionary mechanisms for its development. In the ongoing work it becomes evident that different views exist regarding the missing parts or concepts of the current Internet network architecture. Some scenarios of Internet evolvments based on the stakeholder’s incentives show divers directions of development. This paper introduces the major findings regarding the identified challenges and the evolutionary mechanism suggested by the EIFFEL Think Tank.

Keywords - Future Internet evolving mechanisms; Scenarios and visions; EIFFEL project think-tank.

I. INTRODUCTION

The current Internet is considered as remarkable success of the technology advancement. The Internet platform enabled innovation that far exceeds the original vision of the system as a research instrument. The Internet and associated services today have transformed the lives of billions of people in areas as diverse as democracy, education, healthcare, entertainment, commerce, finance and civil infrastructure. It can be easily claimed that the Internet is the 21st century's fundamental societal infrastructure, comparable to the railways of the 1800s and roadways of the 1900s. The Internet and the associated services have contributed to the transformation of the world economy and society. They catalyse new forms of communication, collaboration, creativity and innovation. They deeply affect the human communication, interactions and transactions, and the way humans deal with information and knowledge.

All the data and statistics related to the Internet are *still* growing at exponential rates. According to the last report of the Task Force of the European Commission DG INFSO the Internet connectivity is expanding rapidly in both terms: geographical distribution and size [1]. Currently there are

about 1.6 billion Internet users worldwide (from 360 million in 2000) and 4 billion mobile users (from 2.7 billion in 2006); 570 million Internet-enabled handheld devices are in use. The number of people who use the mobile phones for web surfing has doubled since 2006 [2]. It is expected in 2012 the number of mobile and wireless users to outnumber the wired ones. In parallel with user growth the stored information is growing as well or even faster. In 1998, Google indexed 26 million web-pages; in 2009 it indexes 1 trillion. There are 400 million web pages and 55 trillion links between these web pages. The Web is processing 100 billion clicks per day, 2 million e-mails and 1 million instant messages per second. Video traffic over the Internet is growing by 60% every year and will be multiplied by 1000 over the next 5 to 8 years. Web 2.0 and social networks with popular social sites are attracting more than 125 million regular users within just 5 years of existence [2]. Internet is today indispensable part of the businesses as most of the businesses processes have been significantly automated by the underlying Internet technologies in business systems, production, development and communication. The current Internet is the most important infrastructure of the digital society that is adapting itself by use of ad-hoc technical solutions that help to meet the demands of the users and devices, applications and services enabling human activities that were not foreseen in its original design.

The networking community being aware of the rising number of seemingly ad-hoc solutions to the technical problems has come to agreement that these problems are of architectural nature and for that reason a general re-design may be needed. It is common understanding of the community that the design of the Future Internet should enable smooth evolvement of the current IP network and should not lay on the current practice of patches being developed and implemented to overcome the existing tussles. It is also a common understanding that the structural and architectural problems of the current Internet cannot be solved without understanding of how a system with the size of the Internet interacts with the world either being human or just some mechanical part of it.

This paper presents some of the identified immediate problems and challenges that require fundamental rethink of the set of mechanisms in use in the today Internet and its

architectural origin. It is based on the work within the EU FP7 project Evolving Future Internet for European Leadership –EIFFEL [3]. The project is organizing semi-annually think-tank meetings where technical and other experts from all parts of the world are contributing to the debate about the future of the Internet trying to provide answers to the major challenges and tussles. Most of the identified agreements and disagreements regarding the major problems of the current Internet are provided at the FIPEDIA site [4] maintained by the EIFFEL core team. This paper introduces the major findings that are presented in detail in the EIFFEL White Papers on the Future of Internet [3].

The paper is organised in five parts, the first part introduce shortly the on-going activities about the Future of the Internet around the world, then focuses on the EIFFEL evolutionary mechanism approach in developing the Future Internet and finally provides an insight to the socio-economic challenges that need to be approached. The paper ends with conclusions and ideas for continuation of the research agenda debate.

II. THE NETWORK COMMUNITY DEBATE

The networking community is aware that the Internet network in use is still based on the best-effort, point-to-point service model, well suited to applications between two endpoints that can tolerate occasional performance degradation. Considering the current level of service where performance degradation is not acceptable but in the same time many of the used applications involve multiple endpoints and their identification in the Internet network the design of the new model becomes even more difficult. Deep consideration of the alternative service and network architecture to solve the tussles is becoming even more necessary. However, the views and the approaches within the research initiatives and efforts towards Internet evolution differ.

In U.S the NSF NetS research program FIND [6] is the major long-term initiative in the area of the Future Internet program where "clean slate process" research proposals in the broad area of network architecture, principles, and design, are trying to answer to many questions within the area of Future Internet. The philosophy of the programme is to help conceive the Future of Internet by enabling a network design that is free from the current collective mindset about the constraints of the network. The NSF is recently considering the NetSE (Network Science and Engineering Committee) report program published in mid-2009 [7], in which further of R&D activities based on theoretical approaches that help to overcome the barriers in future network design are recommended. GENI [8] is also another U.S based program focusing on a flexible and reconfigurable network "test-bed" experimental facilities and related experimental projects.

The EU through the FP7 program is engaged in funding a very wide range of research activities that relate to the

future Internet. Given the scale of this activity, and the rate, at which it is generating results, a complete, up-to-date, snapshot of all related European R&D activities in the area is difficult to be provided. Some form of cross-project, cross-domain body that promote information sharing and helps to set a balance between coherence in order to exploit knowledge generated by number of participants, and the existing diversity is happening within the Future Internet Assembly that was established in March 2008 in Bled, Slovenia. FIA with semi-annual meetings is ensuring appropriate coverage of this very large and challenging research domain that includes innovative research in the area of networking, experimental facilities and testing within the FIRE [9] program. Recently the initiative related to the Future Internet enterprise system – the project cluster FinES [10] was added to the FIA program. Recently in July 2009 the final report of the EU DG INFSO [2] Task Group on Interdisciplinary Research Activities for the Future Internet was published where the design, implementation, testing and validation platforms are identified as major research challenges for the EU in the incoming years. Cross-disciplinary research activities are the essential part of these platforms. Japan, Korea (KOREN) and India have set up similar initiatives and Asia with China has as well its own research initiative on the Future of the Internet – AsiaFI. Cooperation and exchange of information between this initiative and the EU FP7 projects have been recently set up.

The Internet has influenced many changes in the world in the society, culture, commerce and technology. Activities about the Future of the Internet that includes discussions about the Internet governance and business models are on-going in other communities e.g., in the international governmental and non-governmental organizations such as OECD [11], ITU [12], UN-IGF [13]. Internet Society as well together with ICANN are developing position papers and projects on issues such as the Internet Economy, Internet Governance, Network neutrality are being presented and discussed on ITU and IGF forums. The recently expired contract between ICANN and the U.S government (NTUA) is one step forward toward building up of real internationally governed corporation and inclusion of the civil society as its constitutional part. Important observation in exploring these initiatives is that the balance between the new network design for the expected new Internet in the attempt to bias the Internet towards one particular model of governance and business model is difficult to be achieved. In other words, the architecture to be designed must attempt not to prescribe the outcome of particular tussles in the (future) market place beforehand rather than allow for tussles to commence inside the architecture at runtime. Articulating the grand challenges and working towards solutions needs a wider debate as well as concrete work among a growing community of (interdisciplinary) researchers and major stakeholders. The need is clearly understood by the members of the think-tank of the EIFFEL project [3]. Different views exist in respect of what may be

missing from the current architecture or why such concepts are missing. Some of the agreements achieved during the think tank meetings are presented in the section 7 of the paper. Full report is available in the EIFFEL White paper [5] and on the recently set up FIPEDIA portal [4].

III. EVOLUTIONARY MECHANISMS

EIFFEL think-tank has come to agreement that a consideration of a large-scale system such as the Internet need to be carefully observed before starting the new design. The evolution of the Internet is becoming compromised [14] when the architecture does not allow legitimate concerns to be expressed after its original design. As a result, users, providers and business customers solve their problems in *ad hoc* ways, adding carbuncles in violation of the original architecture. Then subsequent requirements are even more difficult to satisfy, because of all the feature interactions with the number of exceptions to the original architecture.

The root of this problem lies deep in the processes used to design architectures and solutions. Currently, much emphasis is placed on the design phase of the architecture, with requirements phases and use case definitions, accompanied by processes of standardization. This *inevitably leads to an emphasis of the concerns that are important to the players who are deeply involved in this phase while often neglecting the concerns of the actors entering the scene after the solution has been fixed.* This Newtonian-Descartian concept of system design, relying on such requirements and use case definition phases, assumes the ability to capture all *relevant* concerns and therefore resolve the most probable run-time tussles at design time. The widening scope of the Internet beyond mere technology and the observed increase of ad-hoc solutions after the design of the original architecture bring this design process into question. Some authors propose [15] a shift from these reductionist Newtonian-Descartian towards Darwinian approaches [16], where the *evolutionary kernel* is a component that has been proven to be successful for multiple uses, so it may act as a platform for evolution around it, (see [15]) and becomes the design process itself, i.e., a process, in which concerns of actors are incorporated into the system at runtime, recognizing the inability to cater to all possible requirements during design time. However, this requires an understanding of what had been good and should be preserved or used in the new design.

These consideration and agreements achieved during the discussions of the Internet evolution can be summarized as follows:

- There is a need for evolution as a gradually developing process, like for any large-scale system. This evolution of the system is particularly important considering the evolution of society due to the impact of the system itself. In order to understand the suitability of the system to evolve, we need to understand the dynamics forcing the changes and devise an architecture that is suited for

these dynamics to commence in runtime. These dynamics will need to define the required steps and their size in evolution that is being necessary and therefore the changes in the underlying architecture that are being required.

- The scope of the dynamics affecting change of the Internet is widening. The Internet has become more than a technical artefact – it has transformed from a network for geeks to a crucial infrastructure used in society and business. Its impact on these areas is obvious, from e-commerce to e-government, the change in the perception of privacy to many other societal changes since its introduction. The virtual and the real world abide to similar rules regarding human rights and respect for personal space as guiding principles. Hence, the question of evolving the Internet is not a mere technical one anymore.
- Evolution speed is increasing with the advances of technology. For instance, memory is becoming so cheap, in particular compared to the formative years of the Internet, that solutions for caching vast amounts of content locally is likely to transform the way users and customers deal with content.
- In that context another problem needs immediate attention: consumption of energy related to increase of used memory and processing power. Internet is become another area for energy saving and low energy consumption devices on infrastructure and on application level.

Coping with the changes and the research agenda preparation is the issue discussed and worked within the think-tank meetings. It was obvious that the old models of the development mostly based on engineering approach are no more sufficient. The complexity of the system and the interrelation with the society needs scientific methods based on facts and measurement to understand and react to the global picture and the expected evolution in search of solutions.

The root of the problem lies deep in the **processes** that are used to design architectures and solutions. Much emphasis is placed on the design phase of the architecture, with requirements phases and use case definitions, accompanied by processes of standardization. This *inevitably leads to an emphasis of the concerns that are important to the players who are deeply involved in this phase while often neglecting the concerns of the actors entering the scene after the solution has been fixed.* This Newtonian-Descartian concept of system design, relying on such requirements and use case definition phases, assumes the ability to capture all *relevant* concerns and therefore resolve the most probable run-time tussles at design time. The widening scope of the Internet beyond mere technology and the observed increase of ad-hoc solutions to concerns of actors after the design of the original architecture bring this design process and the applied research methods into question.

IV. POSSIBLE RESEARCH METHODS

The basis of scientific research lies in the ability to formulate and test falsifiable hypotheses. The role of engineering is to create, evolve and maintain operational systems according to a particular design brief. The Internet provides an environment that is rich in possibilities for research that is experimental and analytical, which at the same time, must be set in the context of engineering; likewise, Internet engineering must respect the need to use the engineered system as an experimental platform and as a platform for innovation, both of which might cause the underlying design brief to change.

The Future Internet is, consequently, more about process than product. Although it is likely that a Future Internet will result from agreement by committees representing industry players and governments, it is crucial for individuals (including researchers) to understand how to influence the key decision makers to eventually adopt the 'right' solutions. The present Internet design is taking part within organizations that have historically focussed more on engineering than innovative vision and social interactions. Some believe the future Internet will come about through the same institutions that fostered the current Internet; the networking research community and the Internet Engineering Task Force (IETF), but this is unlikely to be realistic, simply because the importance of the Internet has changed with time and the list of stakeholders with an interest in design outcomes has grown. As things stand, it is undoubtedly the case that many proposals will be standardised in a variety of committees belonging to several associations. Those proposals that are most worthy and manage to attract support of the key stakeholders will be deployed, and those that survive the rigours of the marketplace will become the Future Internet.

In the same time it is obvious that even very successful network architectures should change over time and this fact should be present in the overall considerations of the future of the Internet. All new systems start small. Once successful, they grow larger. The growth brings the system to a new environment that the original designers may not have envisioned, together with new requirements that must be met. For example the security threats facing the Internet in recent years should not be blamed upon the inadequate design of the original architecture. Rather, it is due to poor understanding of its limitations and the missing adoption from the users. Continued success requires continued scientific research on networking practice, to identify new problems and evolve the architecture to meet the new demands.

Another aspect that needs to be addressed is full understanding of the driving forces behind the Internet's success. The Internet would not have succeeded so greatly without Moore's Law. Computing technologies are moving forward with accelerated speed. The Internet architecture facilitated the technology advances. The rapidly advancing

technologies in turn drive new application developments and user population growth on the Internet.

Technology advances and Internet growth have created the new demands on the architecture. The need for security, manageability, and scalability showed up over time. Today they are more pressing than ever, as they were not promptly identified and fixed ahead of the crisis. This requires continuous identification and address of the new demands. One unfortunate fact to be claimed is that there has been a big gap between reality and how the research community understands it. Since the Internet commercialization in mid 90's, the networking research community gradually lost touch with the frontier of the Internet, lost the opportunity to observe real problems. The community by and large retreated back to work on isolated or point problems, and used simulations or small, isolated test beds for design evaluations. The research community's lack of attention does not mean real problems do not occur, but only that the problems are solved by others frequently on ad-hoc basis.

Designing a technical system creates an economic one, while the latter is enabled by a variety of technical systems. In reality however, the process of (technical) system design is mostly disjoint from the process of designing business models and strategies for sustaining them over a period of time. Combining these two processes is difficult, largely because of the communities that are required to interact. The challenges are in the sustainability of the systems, which cannot be assured without a joint design process. A solution to this problem will not only have an impact on the design of systems but also, for instance, on the way the educational system shape the talents in their understanding of these fields, as has been recognized during the debate of the EIFFEL think-tank [3]. For this to happen, however the differences of *research styles* that exist between research fields, like economics, engineering and social science should be accommodated and make usable in the engineering design.

V. INTERNET SOCIETY -ISOC SCENARIOS

The Internet Society (ISOC) [17] contributed to the EIFFEL think-tank discussions with an illustration of the possible evolution based on the »Internet Futures Scenarios" exercise done in 2009. This exercise produced four visions of possible future in cases when stakeholders' interests could achieve dominance in the practices development. The scenarios are presented on Fig.1. The scenarios illustrate possible futures designed around two axes that point to different outcomes: whether the future Internet will remain true to the old open Internet model (generative, rather than reductive) or whether it will become distributed and decentralized. Other alternative is the Internet to become a subject to command and control of regimes. These axes represent two key areas of external world tussles (social, economic) between Internet stakeholders, impacting the deployed Internet reality.

Together, they form four quadrants, each of which can be described as an illustrative scenario.

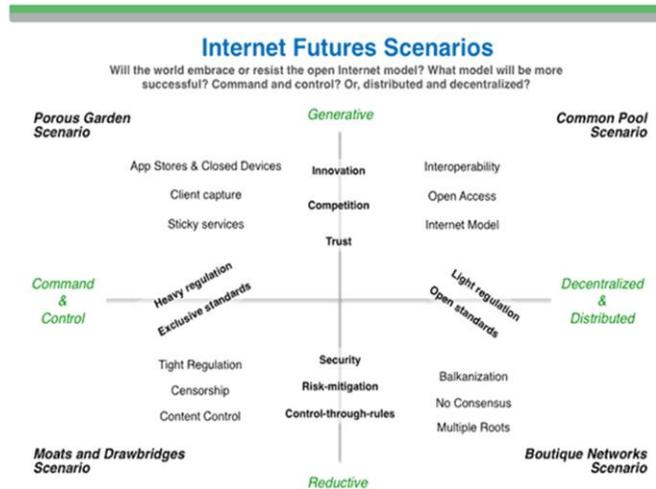


FIGURE 1: ISOC FUTURE INTERNET SCENARIOS

The quadrants between the two axes reflect the effects of misalignments in the incentives of the Future Internet between stakeholders. The main incentives and what drives the stakeholder’s actions are presented on Table 1. The columns provide categorization of the major stakeholders in regards to their fears of and what they are greedy about.

TABLE 1. INTERNET STAKEHOLDERS INCENTIVES

	Fear	Greed
End Users	Privacy, Overcharge, Pricing Unfairness	Cheap/free services, Cheap/free content, Cheap/free network access
Service/Content Providers	Losing their market share because of competition, innovation and regulation	Market dominance, Monitor users’ behaviours, Tiered Services (no network neutrality)
Network Providers	Losing their market share because of competition, innovation and regulation	Market dominance, Monitor users’ behaviours, Tiered Services (no network neutrality)
Governments	Security, Politics	Control of Content, Control of Access to Content, Monitoring (Spying) Users

The scenarios consideration has shown that among the four quadrants the Common Pool quadrant is most positive regarding the "generative" and "distributed and decentralized" properties of the Future Internet. All resources that are part of the future Internet are made available in the "Common Pool" scenarios to the overall community. This scenario can be considered as the ideal, for which Internet development and deployment has always striven, though never achieved in perfection. This scenario tends to provide maximum flexibility and deployment, innovation and opportunities to all stakeholders. Technologies are planned to be built out "horizontally", rather than in full service verticals. This quadrant is named as a Common Pool in order to reflect the notion that it

represents for the future where (information service and application) gardens will not be completely walled, but still somewhat restricted to particular channels.

The Porous Garden scenario is designed around the stakeholder’s incentive for increased control over business and revenue. In this quadrant the application and service provider stakeholders are leading the evolvement with architectures that feature increased command and control in the "vertical" services. In this vision of the Future Internet, the networks remain global but access to content and services are tied to the use of specific networks and associated information appliances. Financial incentives for content producers and software developers would result in continued innovation within the appliance-based model, but network operators will be constrained to evolve their services to support appliances and not the general Internet services. Consumers would have to purchase multiple appliances and associated subscriptions to avail themselves of the full range of innovation on the network. This scenario reflects the general mis-alignment between the incentives of the content producers and those of end users, as well as (ultimately) the network operators.

The Boutique Networks quadrant present a scenario where applications are not expected to be the dominant driver of the future but contrary to that allows the networks to be. This quadrant reflects the possibility of network specialization to become dominant. In that case Internet is not expected to be a single, general network, but rather individually constrained and composed from purposed networks that provide "boutique" services. It envisions a future in which political, regional, and large enterprise interests fail to optimize on the social and economic potential of a shared, global set of richly connected networks (the today Internet), but instead reflects the outcome of parties intended to optimize control in small sectors (political and otherwise). While these balkanized networks continue to leverage the benefits of existing Internet standards, they do not collectively provide the basis for generalized application and service development and deployment. In that regard, this quadrant represents the converse of the Porous Garden quadrant, in that it is network development interests that are expected to dominated.

The Moats and Drawbridge quadrant reflects a future where stakeholders are seeking tighter command and control and more reductive, constrained network environments are expected to prevail. The increased (perceived) need to provide security and consistent environments through "command and control" operation and closed development practices, this scenario is drawing an Internet that is heavily centralized, dominated by a small number of big players who create their own rules in a few "big-boys" clubs. In this scenario it can be expected strong regulation as governments will seek to impose some public interest obligations on the industry, as the user interests and incentives will not be natively supported. Control could extend to limiting equipment that could connect to the network, content could

be proprietary and protected by strong intellectual property rights. This quadrant shows the highest barrier for entry of new applications, networks, services and end users.

The description of the above scenarios allows a selection of the best vision based on the consideration discussed at EIFFEL think-tank meetings. From an economic point of view (the most democratic at least), the perfect scenario would be the one that exists in a perfect market; the one for which the following statements are true [18]: Perfect market information, No participant with market power to set prices, No barriers to entry or exit and Equal access to production technology.

An immediate result of the perfect market information should also be the existence of perfect pricing mechanisms. Doubtlessly, the Common Pool scenario is the one that mostly resembles to the perfect one and is close to the current "ideal". The significant question here is if it is possible to influence the design of the Future Internet so that it can naturally stabilize itself to this quadrant (or towards to a perfect market)? The answer is probably "yes" and the way to achieve it is via a design for change. In that context a special attention has to be paid to the information part, which in a network based system translates to network measurements and monitoring

VI. BUSINESS AND SOCIAL DEMANDS

The particular (technical) approach to the Internet has created business structures that evolved around it, such as expressed in transit and peering relationships of autonomous systems. Any evolution of technologies but in particular any fundamental approach to change the current Internet will undoubtedly have an impact on these existing business structures. Too radical a change will cause problems in adopting the change – and the lack of understanding the proper impact can delay the advancement. Hence, technical and economic migration strategies from *here* to *there* are crucial when targeting a wide adoption of proposed changes. For this reason it is a must the grand challenges in economics to be addressed as well. This needs to start with gathering the right audience for this work and it needs to be driven by a clear emphasis on the concrete problems and the quest for some answers to these.

Regarding social interaction it is common fact that Internet in its early days was a vehicle for both enabling email based communication (reasonably immediate, but not requiring real-time end-to-end connectivity) and for simply improving the information flow between parties, which would have otherwise exchanged the information, but more slowly or in lesser quantities. In addition, along with this beneficial relationship with social structures, it brought as well antisocial opportunities and mis-use. The demands of improving social communication and reflecting social structure are growing but in the same time increasingly issues of privacy and safety in a completely connected world are being addressed. This may follow many possible alternative paths of development. Birth/death records,

medical records, banking records and so forth were kept long before there was an Internet, but the Internet not only made them aggregately, but also made it simpler for malefactors to get at them, even attacking the infrastructure itself. It was only as the infrastructure became increasingly integrated into, and critical to, our society transformed now in digital society that attacking it also became increasingly worthwhile.

In that context it is important to be understood that Internet is a reflection of society, but this reflection is always a partial. As such, it will evolve to provide increasing aspects of social infrastructure requirements, but it is unlikely the prediction where the next step will be to be accurate. In fact, some of the innovation comes from other quarters. Who would have thought that carrying around small wireless cell phones with tiny keyboards would turn into instant messaging and from there make the leap to the Internet and soon into all the different modes of social networking with the user designed Web 2.0 tools? Hence, innovation will always have an element of surprise for some stakeholders in society.

One of the interesting social challenges the user community is facing to is the information overload. There is too much information. There are too many services that want to claim the user trust. There are too many options and too many individuals who want an attention. A challenge will be to evolve approaches that reflect the human and social approaches to dealing with overload. This is already happening in what are probably simple ways in social networking contexts. Users group the friends; create channels for topics, create wikis, follow the friends via twitters and so forth. The world is being actually clustered, but this can be understood as the early stage of the social change induced by the Internet. Newspapers were a mechanism for filtering, organizing and limiting information that otherwise would overwhelm the reading audience. With the demise of newspapers, what elements of the almost infinite flow of bits will bring order that is reflective of the human mind and human social structure? In the longer run, will that also allow each of the humans to retain a somewhat personal view in large social structures? How will the individuality and privacy be retained?

Next question to be answered is the impact of the governance on the Internet or vice versa: what is the impact of Internet on the governance. It is clear that the low-cost and pervasive availability of a uniform communications substrate has had an immeasurable impact on the global society that is becoming digital society. Historically explorers circled the world and laid claim on behalf of their home countries to other lands, thus beginning the political and economic connectedness around the globe. The presence of the Internet has qualitatively changed the nature and degree of that connectedness. In the current economic and political situation, no country can make decisions that will have only a local effect. There is no more isolation. Given that, the relationship between the *Internet* and

governance is becoming even more important. Perhaps even more importantly is the possibility Internet to change forever governance of, by or for a people. Blogging and cell phone cameras that can transmit photos are having profound effects on the capability of individuals to constrain their governments at times when the governments may not want that. This is likely to have an impact on, e.g., regulation when considering a growing role of end users in the participation of the Internet, i.e., end users potentially grow into an essential part of the Future Internet, moving away from their current pre-dominant role of a mere consumer. How this will affect ways to regulate certain parts of the Internet will be important to understand.

VII. NEXT STEPS FORWARD

The most important observation of the EIFFEL Think Tank is that the future architecture to should not be a balance at design-time towards the wanted world, instead minimum substrate should be designed that allows the Internet flexibility to behave in different ways at different times and in different places depending on the outcome of market selection and social regulation mechanisms [19], [20] whilst retaining levels of performance that render it fit for purpose. Hence the research should move from a largely design time to a largely runtime model for resolving potential tussles. Some of them as identified by the EIFFEL Think Tank [3] can be:

- **Resilience, failure tracking & management:** The Internet's distributed design is popularly renowned for its robustness to failure. Indeed failures often do heal automatically, but not quickly. The result is an increasingly unreliable service. Also many failures are not amenable to automatic solution, being due to human errors in configuration and so forth. It is generally believed that the Internet as of today does not have effective solutions these problems.
- **Availability & robustness to attack:** The Internet is continually being used as the means for malware to attack both services and the Internet infrastructure itself. Solutions to these problems often block innovative legitimate uses of the Internet as well as illegitimate ones, effectively slowing down the Internet's evolvability. Proper architectural support to address the root means of these attacks is needed, but there is no consensus between the contending partial solutions.
- **Information security scalability:** The state of the art in information security techniques is sufficiently robust to assure any form of security, except that the techniques do not scale to global proportions in non-hierarchical groups. Another aspect of information security is that of information accountability. While the Internet can cause information to be shared or not, once it has been shared at all, any control is essentially lost of any further sharing and exposure and are dependent on

some vague sense of trust in those with whom we have shared.

- **Resource accountability:** The Internet architecture allows everyone to use any resource anywhere on the Internet to the extent that they want. However, at present, network operators are deploying boxes to limit or block communication with certain users or by certain applications. Even if the Internet networks were trying to share the capacity without making judgements about content, the architecture does not reveal the information they need to make other networks and their users accountable when they are over-using stressed resources. The consequent inability to properly limit free riding (or to deliberately allow it) leads to uncertainty over whether capacity investments can be recouped, which in turn negatively affects the whole value chain of the Internet.
- **Network-application coordination:** Over the years, the application programming interfaces (APIs) at the top of the TCP/IP protocol suite have become ossified and stale, but more importantly they have become almost impenetrable. In the downward direction, middle boxes (e.g., firewalls and network address translators) only recognise those protocols that existed when they were deployed. So they block out all attempts by applications to use new APIs to new lower layer protocols and services. In the upward direction, applications cannot find out about the network or their paths through the network in order to create richer services themselves—services that could exploit knowledge of network topology, network failures or traffic characteristics.
- **Scaling for more extreme dynamics:** The dynamic range of the Internet architecture is hitting its limits. For instance, increasingly the inter-domain routing system cannot converge quickly enough following a change, leaving longer periods of disconnection. More sites are connecting to the Internet through multiple links to improve resilience, but the inter-domain routing system is designed so it then has to treat these sites as distinct networks rather than as stubs off a single-provider network. This makes the routing system appear much larger without the Internet growing at all. Also the Internet's congestion control mechanisms have hit the end of their dynamic range since higher bit-rates require higher accelerations to reach them.

By trying to see into the future through debates such those taking place on the Fipedia site a value judgement with respect to the current identified and potential (possibly unidentified) tussles could be the best approach for choosing a particular evolution path for perhaps technical, moral, ethical, legal, or business reasons. The nature and impact of

this choice, however, need to be made explicit as well as understood. Since such choices are inherently constraining, the establishment of an orthodoxy that results from making a constricting decision must be balanced by inviting challenge and weighing evidence. For this, it is most important to pay attention in addition to the identified tussles to the evolutionary mechanisms of the Internet—the aspects that determine how evolution progresses and if it progresses at all. Decisions made at this point must remain relevant and fresh for at least as long as the current Internet has proved valuable, in a world, in which Moore's law continues to apply. Investment of time and effort in widespread changes to the whole system will not occur unless such changes both deliver in the timescales needed for cost recovery and continue to give returns over many decades in a constantly evolving technological, economic and societal environment. Along this line, the EIFFEL think-tank has still intention to stimulate, even provoke discussion on the major points of why and how the world will be going about the Future Internet.

ACKNOWLEDGMENT

This paper is a result of the work of the think-tank of EU funded FP7 project EIFFEL. Contributions of all members and caretakers of the project are appreciated.

REFERENCES

- [1]. B. Jerman-Blažič. The future of the Internet: tussles and challenges in the evolution path as identified by EIFFEL think tank V: BERNTZEN, Lasse (ur.). The Fourth International Conference on Digital Society 2010, 10-16 February 2010 St. Maarten, Netherlands Antilles. ICDS 2010. [S. l. IEEE Computer Society: = Institute of Electrical and Electronics Engineers, 2010, pp. 25-30.
- [2]. Draft Report of the Task Force Interdisciplinary Research Activities applicable to the Future Internet, version 4.1. <http://forum.future.internet.eu>, 13.7.2009
- [3]. EU FP7 Project EIFFEL, www.fp7-eiffel.eu, 6.1.2011.
- [4]. www.fipedia.org 12.12.2009.
- [5]. D. Trossen (ed), "Starting the Discussion", whitepaper of the EIFFEL think tank, July 2009, at <http://www.eiffel-thinktank.eu>, 12.12.2009
- [6]. FIND (Future Internet Design). www.nets-find.net, 8.9.2009.
- [7]. Network Science and Engineering Council, "Network Science and Engineering (NetSE) Research Agenda", at <http://www.cra.org/ccc/docs/NetSE-Research-Agenda.pdf>, 2009, 8.9.2009.
- [8]. GENI, www.geni.net, 10.12.2009.
- [9]. FIRE Future Internet, www.future-internet.eu, 10.1.2.2009.
- [10]. Future Internet Enterprise System (FIInES) Cluster, Position Paper Version1, 15.5.2009, www.cordis.eu.
- [11]. [The Future of the Internet Economy OECD Ministerial Meeting](http://www.oecd.org/futureinternet) Report, www.oecd.org/futureinternet.
- [12]. ITU Technology Watch Report 10. April 2009 www.itu.int/dms_pub/itu/oth/.../T230100000A0001PDFE.pdf.
- [13]. Forum (IGF) - The First Two Years' Report, www.intgovforum.org/ 5.12.2009
- [14]. D. Trossen (ed), "The Core-Edge Story", whitepaper of the Core-Edge Dynamics working group at the Communications Futures Program, MIT, 2005
- [15]. C. Dovrolis, "What would Darwin think about clean-slate architectures?" In: ACM SIGCOMM Computer Communication Review 38 (1) pp. 29--34 (2008)
- [16]. R. Hollingsworth and K. Müller, "Transforming socio-economics with a new epistemology," *Socio-Economic Review*, vol. 6, pp. 395–426, 2008
- [17]. Internet Society, www.isoc.org, 6.1.2011
- [18]. Wikipedia, URL: www.wikipedia.org/wiki/Internet_marketing, 6.1.2011
- [19]. D. Clark, K. Sollins, J. Wroclawski and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," In: *IEEE/ACM Transactions on Networking* 13 (3) pp. 462--475 (June, 2005).
- [20]. D. Clark and M. Blumenthal, "The End-to-End Argument and Application Design: The Role Trust." Conference on Communication, Information, and Internet Policy (TPRC), 2007.

The Impacts of the Digital Divide on Citizens' Intentions to Use Internet Voting

France Bélanger

Accounting and Information Systems
Virginia Tech
3007 Pamplin Hall
Blacksburg, USA
belanger@vt.edu

Lemuria Carter

School of Business and Economics
North Carolina A & T State University
1601 East Market Street
Greensboro, North Carolina 27411, USA
ldcarte2@ncat.edu

Abstract – Internet voting is increasingly used by governments and corporations as a means for individuals to cast their votes. However, not everyone has access to and is comfortable with the use of technology. This digital divide is composed of the access divide and the skills divide. This study explores the impact of the digital divide on Internet voting (I-voting). We propose a model of the effects of the digital divide on I-voting, which suggests that age, income, education and frequency of Internet use have an impact on I-voting utilization. Online and paper-based surveys were administered to a large sample of citizens of varied backgrounds to test the model. The results of multiple linear regressions indicate that age, income, and Internet use (representing the access and skills divide) have a significant impact on Internet voting. Education was not found to be significant. These findings indicate that, like other e-government services, I-voting is subject to the barriers associated with the digital divide, and this digital divide introduces several challenges to government agencies.

Keywords: *Internet voting, digital divide, technology adoption, e-services, access divide, skills divide*

I. INTRODUCTION

This paper explores the impact of the digital divide on Internet voting [1]. Voting is an important democratic right, and voter turnout is vital to the health of all democracies. A key element of a democracy is the continuing responsiveness of the government to the preferences of its citizens. Turnout rates in U.S. presidential elections (which are the most popular in that country) vary between 50 and 60 percent, with winners never receiving more than 60 percent of the turnout. Hence, presidents are selected by the votes of 25 to 30 percent of the electorate [2]. In fact, the United States ranks at the bottom, or just above last place, in voter involvement when compared to other democratic nations [3]. Research suggests Internet voting could increase voter participation [4]. Internet voting, or I-voting, is defined as “an election system that uses encryption to allow a voter to transmit his or her secure and secret ballot over the Internet [5, p. 2].” Researchers suggest that I-voting has the potential to increase “turnout” among individuals between

the ages of 18-25 since they have experience in surfing the Internet and like the idea of using the latest technology [6]. Morris [7] agrees that the Internet has the potential to mobilize the otherwise disenfranchised voters under the age of thirty-five.

I-voting would be an ideal option for many citizens. Done [4] argues that one of the most important social impacts of Internet voting is the effect it could have on voter participation. A survey conducted at the University of Arizona suggests that 62 percent of the unregistered voting age population would register to vote on the Internet. The survey results also suggest that Internet voting would increase voter participation by about 42 percent while conserving costly resources. These increases would be realized across all sex, age, ethnicity, and education groups [4].

Many countries have conducted research on or experimented with Internet voting [8]. In the Netherlands, 62% of the people with access to the Internet would prefer to vote online [9]. In New Zealand, a taskforce concluded that Internet technology might boost the number of voters, speed the count, and reduce costs. In Japan, the Center for Political Public Relations experimented with poll site Internet voting in the 2001 gubernatorial election in Hiroshima. In 2005, Estonia was the first country to offer Internet-voting as an option nationwide for mayors and city councilors [10].

In the United States, the 2000 Arizona Democratic primary offered the first binding Internet election for public office [4, 11, 12]. In 2008, Okaloosa County in Florida allowed hundreds of military personnel in Germany, Japan and the United Kingdom to cast their votes in the presidential election [13]. Despite the gradual implementation of I-voting and its potential to increase participation, some citizens may not benefit from this innovation due to the digital divide.

The paper explores the relationship between the digital divide and I-voting. Whereas one can argue that I-voting offers simply another electronic service, voting is a fundamental right in democratic societies, available to all

citizens, irrespective of their income, education or social status (within legal limits).

The paper is organized as follows. First, we discuss the issues of the digital divide to provide background for the development of the research model and the hypotheses. The methodology section describes the research conducted. The next section presents the results and their implications for research and practice.

II. THE ISSUE OF THE DIGITAL DIVIDE

As governments worldwide begin to implement more technology-based voting systems, in particular Internet voting, concerns about the potential impacts of the digital divide continue to grow. The digital divide refers to the distinction between the information haves and have-nots; the gap between the computer literate and the computer illiterate. Researchers have been interested in the digital divide from a variety of perspectives, including a demographic view [14-16], a global view divide [17], an urban view [18, 19], and a psychosocial view of the digital divide [20].

The digital divide is composed of two major barriers: access to technology and comfort with technology [21]. Both of these barriers may play a role in limiting the use and convenience of Internet voting. Not surprisingly, researchers have found that demographically, citizens who use the Internet for political purposes differ from the rest of the population, particularly in terms of income and education [22]. It could be because education and income increase the likelihood of openness toward Internet voting [4]; it could also be due to the digital divide barriers of access and skills. We discuss each of these digital divide barriers before presenting the research model in the remainder of this section.

A. The Access Divide

The access divide refers to factors that may limit an individual's access to technology that can be used, in this case for Internet voting. Prior research has identified ethnicity, income, age and education as significant predictors of access to technology [23, 24]. A more recent study finds that income, education and age significantly impact who is willing to use e-government services such as electronic tax filing or license renewals [21]. This is not surprising since other researchers have found that approximately 78 percent of households with income between \$50,000 and \$75,000 had Internet access compared to only 40 percent of those with household incomes between \$20,000 and \$25,000. Others find that young citizens (18-24) and their parents (45-54) report the highest levels of home Internet access, reaching better than 61 percent [25]. Research also shows that more younger Americans have an Internet connection than older Americans [6]. Thomas and Streib [24] suggest that among Internet users, ethnicity and education are important predictors of government Web sites utilization, with white

and better educated users more likely to be uses such sites [24]. Interestingly, gender differences in access and use of computers has narrowed over the years, with recent research suggesting that it does not impact use of e-government services [21]. This is consistent with findings from the Pew Internet Project report, which suggests that although men and women have different attitudes toward technology, the surge in the number of women online has eliminated some of the disparity in access between genders [26].

B. The Skills Divide

In addition to Internet access, comfort with Internet technology is also a major element of the digital divide. The skills divide refers to a disparity in skills necessary to effectively interact with online systems [23]. Other researchers call this the second order digital divide [27]. Mossenburg, Tolbert, and Stansbury [23] identify two components of this skill divide: technical competence and information literacy [23]. Technical competencies are "the skills needed to operate hardware and software, such as typing, using a mouse, and giving instructions to the computer to sort records a certain way". Information literacy is "the ability to recognize when information can solve a problem or fill a need and to effectively employ information resources." Researchers have found that the old, less-educated, poor and minority individuals (African Americans and Latinos) were more likely to need computer assistance (such as help using the mouse and keyboard, using e-mail, or using word processing and spreadsheet programs), although recent studies show some of the differences disappearing after a year or two of use [28]. It is also possible that as new user interfaces such as multi-touch screens and touch screens become more popular, skills require to use the computers will become less of an issue. Nevertheless, comprehension of the navigation, applications, and resulting information will still be required for completing digital tasks.

In this study, we use frequency of Internet use in general as a proxy measure of technical competence and information literacy. The use of this proxy is consistent with Belanger and Carter [29]. Citizens who use the Internet frequently should possess a level of technical and information literacy.

C. The Research Model

In summary, differentials in age, income, education, and Internet usage, seem to create a digital divide that should affect which individuals will choose to use Internet voting as a means of performing their constitutional right. Figure 1 summarizes the access and skills divide factors that are expected to affect one's intention to use I-voting.

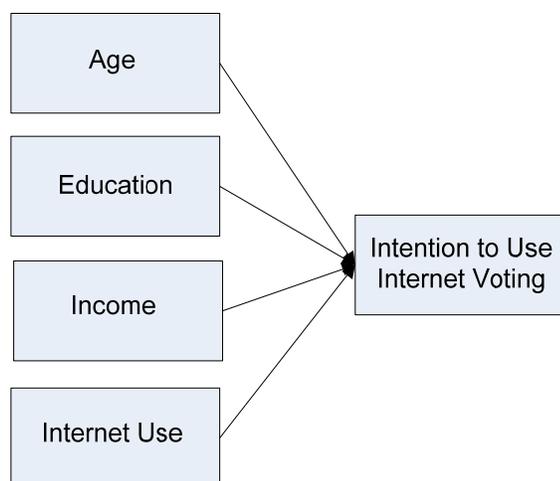


Figure 1. I-Voting Digital Divide Factors

As depicted in the model, there are four hypothesized relationships:

H1: Age will have a negative impact on intention to use Internet Voting.

As discussed before, a larger number of younger adults tend to have Internet access than older citizens. Since Internet access is a condition for I-voting, it can therefore be expected that there is negative relationship between age and I-voting intentions. Furthermore, since younger adults also tend to have more computer skills than older adults do, it further reinforces the negative relationship between age and I-voting intentions.

H2: Education will have a positive impact on intention to use Internet Voting.

Education is closely linked in two ways to the relationship between the digital divide and I-voting intentions. First, individuals with higher education tend to have more computer and Internet skills, as previously discussed. As a result, it is expected that education has a positive relationship with I-voting intentions. Of note, education can also impact I-voting intentions through the higher income that highly educated individuals tend to have. For the sake of parsimony and simplicity, our model does not test the potential mediating effect of income between education and I-voting intentions. Nevertheless, this potential relationship would also involve a positive link between education and I-voting intentions.

H3: Income will have a positive impact on intention to use Internet Voting.

As prior research suggests, individuals with higher income tend to have more access to the Internet (and other

technologies). They also tend to have more education, and potentially computer skills. As a result, it is expected that income is positively related to I-voting intentions.

H4: Experience using the Internet will have a positive impact on intention to use Internet Voting.

Because individuals who have used the Internet for a longer period of time are expected to be familiar with the terminology, tools, and features of the Internet, it is expected that their skills will easily translate to the use of the Internet for voting. As a result, we expect that Internet experience is positively related to I-voting intentions. We discuss in the next section the methodology used to test the research model and hypotheses.

III. METHODOLOGY

A. Overview

To identify the salient I-voting divide factors, we surveyed a diverse pool of citizens. Both online and paper-based versions of the resulting instrument were administered to participants. There were various sources of data collection for each version. The paper version of the survey was administered to members of a church choir, students in a religious seminary class, attendees of a symphony concert, and employees in a county agency. The online version was posted on a local website, disseminated through a graduate student listserv at a university, and sent to the listserv of a community fitness group. 372 surveys were used for data analysis: 133 paper responses and 239 online responses.

An independent samples t-test was used to identify any differences between online and paper responses. Since the two groups did not exhibit differences for the dependent variable - intention to use an I-voting system - a combined sample was used in the data analyses.

B. Instrument Items

Each I-voting divide factor was measured using categorical data on the survey instrument, except for age, which was measured by respondent writing their actual age. Five age categories were then used to classify the data: 18-24 years, 25-29 years, 30-44 years, 45-54 years, and 55 years and older. Education was measured using four categories (Grade school/some high school, High-school Diploma (or equivalence), Some college: no degree, and College degree/post graduate). Income was measured using seven categories (Less than US\$20,000, US\$20,000 - US\$34,999, US\$35,000 - US\$49,999, US\$50,000 - US\$74,999, US\$75,000 - US\$99,999, US\$100,000 - US\$149,999, and US\$150,000 and above). Internet usage was measured using four categories representing the number of years a citizen has been using the Internet (0-3 years; 3-6 years; 6-9 years; 10 years or more). Finally intentions to use I-voting (USE) was measured using four items adapted from a study of e-government [29], which

used a seven-point Likert-type scale (from strongly disagree to strongly agree).

C. Sample Demographics

Regarding sample demographics, the age range of participants is 18 to 75 years with an average of 33 years (see Table 1). Most participants (78%) have a college degree, and the reported income range is well distributed. Forty-four percent of the sample makes US\$50,000 or more a year.

TABLE 1. AGE DISTRIBUTION

Age Category	Frequency	Percent	Cumulative %
18-24 years	92	24.7	24.7
25-29 years	71	19.1	43.8
30-44 years	104	28.0	71.8
45-54 years	57	15.3	87.1
55 years and older	48	12.9	100

In addition to the demographics mentioned above, general information about the participants was collected. The sample was 63% female. A majority of the subjects were Caucasian (64%). African-Americans accounted for 26% of the sample and Hispanic, Asian and Native Americans accounted for seven percent of the sample. The remaining three percent of the subjects did not report ethnicity. In terms of access to and experience with the Internet, most participants reported high levels, with the exception of having used e-government services, where only 70% of respondents indicated having done so, as can be seen in Table 2.

TABLE 2. INTERNET AND WEB EXPERIENCE. PERCENTAGE OF RESPONDENTS WHO...

...have access to the Web at home	91%
...used the Web to make a purchase	90%
...had used the Web to complete a government transaction.	70%
...voted in the 2004 presidential election	82%

IV. RESULTS

Multiple regression analysis was used for hypothesis testing. Prior to testing the hypotheses, assumptions of multivariate normal distribution, independence of errors, and equality of variance were tested. The USE variable was slightly skewed with a mean of 4.78. Pearson correlation coefficients revealed low correlations among variables, except for age and income with a correlation of 0.48. Variance inflation factors (VIF) confirmed that multicollinearity was not a concern with this data set (VIF range from 1.11 to 1.29). Outlier influential observations were identified with leverage and studentized residuals. This analysis indicated that thirteen data points were considered outliers. They were removed for data analysis. There were no violations of the other assumptions.

A. Model Testing

The regression analysis results in a model with an F-value of 9.344, resulting in a p-value of $p < 0.0001$, which indicates that at least one of the coefficients corresponding to an independent variable is not equal to zero. The r-square value was 9.5 %, indicating that digital divide factors identified in this research account for nine and a half percent of the variance in intentions to use I-voting. This is important because this is the variance explained on top of what typical adoption factors from theories such as the Technology Acceptance Model (TAM) or the Unified Theory of Acceptance and Use of Technology (UTAUT) should account for.

B. Hypothesis Testing

Since the model is significant, the individual beta coefficient t-tests can be used to identify which digital divide factors are significant. Table 3 shows the results of the hypothesis testing analyses.

TABLE 3. HYPOTHESIS TESTING RESULTS

Hypothesis	Beta	P-value	Support?
H1: Age → I-voting Intentions	-0.271	<0.0001	Yes
H2: Education → I-voting Intentions	0.056	0.301	No
H3: Income → I-voting Intentions	0.211	<0.001	Yes
H4: Internet Usage → I-voting Intentions	0.142	0.008	Yes

Results from Table 3 indicate that age, income, and Internet usage are significant predictors of I-voting intentions. Figure 2 shows the significant results, which are further describes and discussed in the next section.

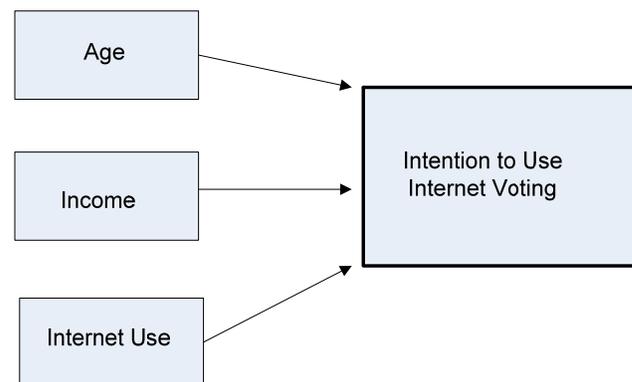


Figure 2. I-Voting Significant Factors

V. DISCUSSION

This study explored how the digital divide impacts citizens' intentions to use Internet voting. The overall result of the study is that there is indeed an impact of the digital divide. More specifically, the levels of income, the age of citizens, and their level of Internet use impact their intentions to use I-voting.

Before we discuss the results and their implications in this section, we need to acknowledge some limitations to the study. One of the primary limitations was the sample's lack of variance in education. The participants in this study are highly educated. Seventy-eight percent have a college degree. Future studies should test a sample with more variance in education. Lyons and Alexander [30] found that education beyond high-school increases the likelihood of voting by almost 15%. Alvarez and Hall [6] found that individuals who have attended college are approximately two times more likely to vote than individuals without a high school education. These findings illustrate the importance of obtaining responses from people with diverse educational backgrounds. The demographic characteristics of this sample may account for education not having a significant impact in this study. Future studies should seek to collect data from individuals with diverse educational backgrounds; an ideal sample could include those who have a high school diploma and those who do not.

An additional limitation of the sample is the lack of variance in ethnicity with only a few Asian, native American, or Hispanic respondents. As such, the sample is not truly representative of the American population in general. However, to avoid any potential lack of representativeness issues, we did not include or test ethnicity as a digital divide factor.

A. Significant Results: Age, Income, and Internet Use

The results of the regression analysis indicate that age, income, and Internet usage are significant elements of the I-voting divide. Younger citizens, individuals with higher income levels, and individuals with more experience using the Internet are more likely to use Internet voting. These findings indicate that, like other e-government services, I-voting is also subject to the barriers associated with the digital divide. In other words, both the access divide (impacted by income and age) and the skills divide (impacted by age and Internet usage) affect I-voting intentions.

As hypothesized, age has a negative relationship with I-voting intentions. In general, this means that younger individuals are more likely to vote using the Internet. There are several potential reasons for this. First, they may have more readily access to the Internet via school, their parents' computers and networks, or even friends' networks. Older citizens are not only less likely to have access to the Internet, but also less likely to possess the computer skills necessary to take advantage of Internet voting. In a post hoc analysis, we show the means of I-voting intentions across age categories in Table 4. To eliminate potential bias, we only used observations from registered voters for this analysis. As can be seen from the table, there is a steady decline in I-voting intentions means as age increases.

TABLE 4. I-VOTING INTENTIONS BY AGE FOR REGISTERED VOTERS

Age Category	n	I-voting means	I-voting St. Dev.
18-24 years	80	5.22	1.65
25-29 years	55	5.16	1.78
30-44 years	98	5.11	1.68
45-54 years	57	4.38	1.96
55 years and older	47	4.29	2.20
	325	4.90	1.85

Since one of the arguments often heard is that I-voting could increase voter participation among younger adults, we identified registered voters who voted on the 2004 United States Presidential election by age category in Table 5.

TABLE 5. LAST PRESIDENTIAL ELECTION VOTING BEHAVIOR BY AGE FOR REGISTERED VOTERS

Age Category	n	Voted	Did not vote	% Did not vote
18-24 years	80	65	15	18.8 %
25-29 years	52	48	4	7.7 %
30-44 years	91	88	3	3.3%
45-54 years	56	55	3	5.4%
55 years and older	46	45	1	0.1%
	325	299	26	8.0%

As can be seen, the results are extremely revealing in that the younger adults are by far less likely to have voted in the last election. Even taking out a portion of the respondents who were not old enough to vote in the last election, there remains a large portion of these younger adults who did not vote. Yet, these same younger adults state they would use the Internet to vote. It is possible, therefore, that I-voting would indeed increase voter participation among younger adults.

Income is positively related to intentions to use I-voting. Citizens with higher income levels are more likely to have access to the technology necessary to take advantage of Internet voting. While the regression analysis identifies a general relationship between these constructs, a closer look at the distribution of I-voting use intentions per income category, presented in Table 6, reveals a more complex situation.

TABLE 6. I-VOTING INTENTIONS BY INCOME LEVELS

Income Category	n	I-voting means	I-voting St. Dev.
Less than US\$20,000	89	4.91	1.79
US\$20,000 - US\$34,999	63	4.55	1.97
US\$35,000 - US\$49,999	53	4.51	1.88
US\$50,000 - US\$74,999	62	5.25	1.86
US\$75,000 - US\$99,999	51	5.35	1.62
US\$100,000 - US\$149,999	31	5.05	1.89
US\$150,000 and above	10	4.95	1.26
	359	4.92	1.84

As can be seen from the data in the table, it is true that lower income individuals show fewer intentions to vote using the Internet than higher income levels, except that the relationship does not seem to be linearly constant across categories of income. There are several possible explanations. First, our income categories are probably too granular, with the effects of the digital divide finding its way into lower than US\$ 50,000 versus higher than US\$50,000. There also appears to be a lower intention to vote using the Internet when income levels are in the very high (greater than US\$150,000) category. It is possible that there is a bell shape (curvilinear) relationship between income and I-voting intentions. It is also possible that these results are simply due to the unequal distribution of responses in our sample. Future research should further explore these possibilities.

Finally, experience using the Internet has a positive impact on intentions to use I-voting. Regular use of the Internet translates into an affinity towards Internet voting. These findings support prior suggestions that I-voting will be more appealing to citizens who use the Internet regularly [31]. In the literature review, we discussed how experience using the Internet is one factor that may reduce the skills divide. However, since voting is such an important civil act, we believe that prior e-government usage might also be an important predictor of intentions to use I-voting. This would be consistent with prior findings in e-commerce where Schaup and Carter [32] found that prior use of an e-commerce or e-government service is positively related to intention to use an I-voting system. To verify this possibility, we ran a post hoc analysis on the effects of e-government usage, of which two measures were available in our dataset, on intentions to use I-voting. One of these variables is whether individuals have used a government website to collect information (EgovInfo), and the other is whether an individual has used a government website to conduct a transaction (EgovTrans). While the EgovInfo variable proved to be non significant, the prior use of a government website to conduct a transaction (EgovTrans) was highly significant with a p value of 0.003. In this study, 70 percent of the sample has completed a government transaction online and 90% has purchased a product or service online. Participants in this study have adopted e-service initiatives in both the public and private sector. As suggested by the literature, citizens who have adopted other e-services are more likely to adopt I-voting.

Even with these results, it is possible that online voting will introduce unique concerns, even among frequent Internet users. Future studies should explore the impact of concepts such as Internet trust and Internet self-efficacy on I-voting acceptance. Future studies should also explore the impact of technology adoption variables on intention to use Internet voting. Perhaps, constructs such as compatibility and social influence would have a significant impact on I-voting intentions.

B. Factors Not Affecting I-voting Use Intentions

Interestingly, education did not have a significant impact on one's intention to use an Internet voting system. This finding could be a result of our sample, which did not have a large variance in education. Seventy eight percent of our survey respondents have a college degree. This percentage is far greater than the population at large. Future studies should continue to explore the effects of education on the digital divide.

C. Implications for I-voting Diffusion

As municipalities begin to make I-voting a viable option for civic participation it is imperative that whole sectors of the population are not "left behind." This digital divide introduces several challenges to government agencies: 1) the sectors in danger of exclusion are already disenfranchised and 2) as long as there is a divide, the government will need to maintain traditional voting methods in addition to Internet options. Older, lower income citizens will need an advocate to ensure that they are not disregarded as I-voting initiatives become more commonplace. The existence of this divide means that I-voting should be used as an accompaniment to, not a replacement of, existing voting procedures.

Government agencies need to discover ways to make online services more appealing to older citizens. The results of this study indicate that younger voters are more inclined to use Internet voting than older citizens. Perhaps government agencies could work with community and/or non-profit organizations designed to help senior citizens, such as the American Association for Retired Persons (AARP) (www.aarp.org), to increase adoption among older users. As senior citizens often become increasingly less mobile, having an easy way to cast their vote could improve the level of participation of this group of citizens in the democratic process.

As I-voting becomes more popular, municipalities also need to make I-voting options available to low-income citizens that may not have Internet access at home. For instance, the government may be able to make voting kiosks available in public places such as libraries, supermarkets and post offices to increase citizens' access to this innovation.

Can I-voting lead to more individuals actually voting? It is unclear that I-voting alone can achieve this, but a post hoc analysis of our data shows at least a potential for this to happen. We compared the individuals who voted in the 2004 presidential elections with those who did not on their intentions to use I-voting if this technology was available to them. Surprisingly, non-voters (59 individuals) exhibited a higher mean for I-voting intentions (5.31) than voters (300 individuals; 4.85). An independent samples t-test reveals that this difference is significant only at the 0.10 level ($p = 0.06$). While this not a highly significant test, it does suggest that future research should explore more in-depth the perspectives of non-voters. A potential avenue to

do this would be through interviews of non-voters on the topic. Alternatively, researchers could conduct experimental studies where non-voters would be presented with an Internet voting option.

D. Social Impact of Increased Voter Participation

The impact of I-voting on political participation cannot be fully ascertained until Internet voting actually becomes a common option for voting in major elections. Recent studies suggest that its diffusion is steadily approaching. Researchers at the Georgia Tech Research Institute (GTRI) predict that kiosk I-voting will be available at post offices, malls, and automated teller machines. By 2012, they predict that some states, especially Oregon, which only uses mail-in ballots, will be the first to adopt Internet voting [33]. In light of the potential for Internet voting to increase voter participation, it is important to consider the potential impact of increased voter turnout on the nation's political system.

Given the current disparities in Internet access and literacy, agencies should be aware of the potential emergence of a democratic digital divide [23]. A democratic digital divide occurs when advancements in technology increase political inequality. This inequality results from the unequal distribution of political power among population groups. Future studies should explore the existence and implications of a democratic digital divide. Will certain groups reap the benefits of Internet voting at the expense of others? As technology transforms the voting process, will socio-economic status persist as a discriminating factor, or will other factors such as political motivation become more salient?

E. Personal Impact on the Act of Voting

In this study, participants were receptive to Internet voting; the mean of intention to use was 4.79 on a seven point scale (for all 359 valid responses), where seven represents the highest level of acceptance for I-voting. In light of this notable adoption potential, it is important to consider the impact of Internet voting on the voting experience.

Some opponents are critical of Internet voting because it deviates from traditional voting methods. Critics of I-voting argue that it will contaminate and eventually replace the most fundamental form of citizen participation in the democratic process. It may result in the loss of an important civic ritual: citizens going to the polls. Coleman [34] writes "reducing a vote to a mere key stroke of a personal computer may diminish, not heighten, the significance of the act. At a minimum, voters who bother to actually go to the polls tend to be people who are motivated enough to learn about issues. The solution to a lack of commitment of voters is not to reduce the necessary commitment needed to vote (p. 2)."

Some critics even argue that I-voting would make elections less of a community event, which might create a

gap between citizens and government, thereby decreasing participation. In light of the diverse predictions regarding the impact of Internet voting on the democratic process it will be interesting to explore its actual implications as this innovation is diffused throughout society.

F. Personal Impact on the Act of Voting

In addition to societal and personal implications, there are also technical implications affiliated with the use of Internet technology to cast a vote. In addition to increasing voter participation, I-voting can also potentially increase the accuracy with which votes are cast. I-voting may increase both the number of ballots that are submitted, and it may also increase the accuracy of the ballots submitted. Tomz and Van Houwelling [35] conclude that the use of appropriate voting technologies can greatly decrease the number of invalid ballots. Internet voting could be one such technology.

G. Additional Research

One important digital divide factor mentioned in prior literature that may impact intentions to use I-voting is ethnicity. As explained before, we did not include ethnicity in our model because we could not obtain sufficient variance in ethnicity levels to conduct proper analyses. However, we provide in Tables 7 and 8 descriptive data on I-voting intentions and voting behaviors per ethnic category for registered voters.

TABLE 7. I-VOTING INTENTIONS BY ETHNICITY FOR REGISTERED VOTERS

Ethnicity Category	n	I-voting means	I-voting St. Dev.
Caucasian	218	4.92	1.85
African-Americans	98	4.72	1.88
Hispanic	6	5.29	1.56
Asian	4	6.63	0.75
Native Americans	3	6.08	1.59
Other/ Not reported	8	5.00	1.62
	325	4.90	1.85

TABLE 8. LAST PRESIDENTIAL ELECTION VOTING BEHAVIOR BY ETHNICITY FOR REGISTERED VOTERS

Ethnicity Category	n	Voted	Did not vote
Caucasian	210	192	18
African-Americans	96	93	3
Hispanic	6	5	1
Asian	4	1	3
Native Americans	3	2	1
Not reported/Other	6	6	0
	325	299	26

As can be seen from the tables, there might be some important impact of ethnicity on I-voting intentions. As such, we believe that future research should seek samples with a wider variety of ethnicities to conduct statistical analyses on the impact of ethnicity on I-voting intentions.

Future studies should also explore the impact of concepts such as Internet trust and Internet self-efficacy on I-voting acceptance. Conversely, future studies of adoption of technologies should include relevant digital divide variables that may have an effect in technology acceptance.

An additional avenue for future research is to expand the digital divide model by exploring additional factors that can impact the intentions to use I-voting. While we included the digital divide factors that are most often found to impact use in electronic services contexts, it is possible that additional factors could be of importance.

Finally, as previously stated, future research would benefit from finding a sample of respondents that is more representative of the current population of the United States of America in order to have conclusions that are more representative [36].

VI. CONCLUSION

The combination of I-voting and the digital divide provides a solid foundation for research on the new areas of e-democracy, civic mindedness and civil society. This paper enhances the Internet adoption literature and emphasizes its relevance to the developing research on e-participation. This study identifies prominent demographic predictors of I-voting intention. Using only digital divide factors (demographics), the proposed model explains 9.5% of the variance in intention. Considering their significance, these variables should be used to enhance the explanatory power of future e-services models that explore technology adoption using established theories such as diffusion of innovation [37] and technology acceptance [38]. The factors identified here can serve as a foundation for future studies of the digital divide and I-voting adoption

As local, state, and national governments begin to experiment with Internet voting, now is the time to identify the characteristics that distinguish potential I-voters from non-I-voters. This study identifies digital divide factors that affect one's intention to use an I-voting system. Governments should find ways to reduce the digital divide issues related to income (by providing more inexpensive access) and Internet experience (by providing community training and access to technology), and find ways to minimize the effects of the age-based digital divide.

ACKNOWLEDGMENT

A prior and shorter version of this work was presented at the 2010 Cyberlaw Conference in St-Marteens, Netherlands, February 11-14, 2010.

REFERENCES

- [1] F. Bélanger and L. Carter, "Digital Divide and Internet Voting Acceptance," in *The Fourth International Conference on Digital Society (ICDS 2010)*, St. Maarten, Netherlands Antilles, 2010, pp. 307-310.
- [2] J. Petrocik and D. Shaw, *Nonvoting in America: Attitudes in Context*, New York: Greenwood Press, 1991.
- [3] R. E. Wolfinger and S. J. Rosenston, *Who Votes?*, New Haven: Yale University Press, 1980.
- [4] R. S. Done, *Internet Voting: Bringing Elections to the Desktop*, The PricewaterhouseCoopers Endowment for the Business of Government Report, 2002.
- [5] A.-M. Oostveen and P. V. D. Besselaar, "Internet Voting Technologies and Civic Participation: The Users' Perspective," *The Public*, vol. 11, 2004, pp. 1-18.
- [6] M. R. Alvarez and T. E. Hall, *Point, Click, and Vote: The future of Internet Voting*, Washington, D.C.: Brookings Institution Press, 2004.
- [7] D. Morris, *Vote.com*, Los Angeles: Renaissance Books, 1999.
- [8] C. Eliasson and A. Zuquete, "An electronic voting system supporting vote weights," *Internet Research*, vol. 16, no. 5, 2006, pp. 507-518.
- [9] W. Pieters, M. J. Becker, P. Brey *et al.*, "Ethics of e-voting: An essay on requirements and values in Internet elections," *Proceedings of the sixth International Conference of Computer Ethics*, Enschede, The Netherlands, 2005, pp. 301-318.
- [10] Anonymous, *Report of the National Workshop on Internet Voting*, Internet Policy Institute, 2001.
- [11] M. R. Alvarez and J. Nagler, "Internet Voting and Political Representation," *Loyola of Los Angeles Law Review* vol. 34, 2001, pp. 1115-1152.
- [12] F. I. Soloop, "Digital Democracy Comes of Age: Internet Voting and the 2000 Arizona Primary Election," *Political Science and Politics*, vol. 34, no. 2, 2001, pp. 289-293.
- [13] E. Sofge, "Internet Voting in Florida Raises Security Concerns: Geek the Vote," *Popular Mechanics Online*, October 1, 2009, <http://www.popularmechanics.com/technology/industry/4288327.html>, Accessed January 15, 2011.
- [14] E. P. Bucy, "Social access to the Internet," *Harvard International Journal of Press/Politics*, vol. 5, no. 1, 2000, pp. 50-61.
- [15] T. Teo, "Demographic and motivation variables associated with Internet usage activities," *Internet Research: Electronic Networking Applications and Policy*, vol. 11, no. 2, 2001, pp. 125-137.
- [16] W. E. Loges and J.-Y. Jung, "Exploring the digital divide: Internet connectedness and age," *Communication Research*, vol. 28, no. 4, 2001, pp. 536-562.
- [17] W. Chen and B. Wellman, "The global digital divide-Within and between countries," *Information & Society*, vol. 1, no. 7, 2004, pp. 39-45.
- [18] D. B. Hindman, "The rural-urban digital divide," *Journalism and Mass Communication Quarterly*, vol. 77, no. 3, 2000, pp. 549-560.
- [19] B. Mills and B. Whitacre, "Understanding the non-metropolitan-metropolitan digital divide," *Growth and Change*, vol. 34, no. 2, 2003, pp. 219-243.
- [20] L. Stanley, "Beyond access: Psychosocial barriers to computer literacy," *Information & Society*, vol. 19, no. 5, 2003, pp. 407-416.
- [21] F. Belanger and L. Carter, "The Impact of the Digital Divide on E-government Use," *Communications of the ACM*, vol. 52, no. 4, 2009, pp. 132-135.
- [22] B. Bimber, "Information and Political Engagement in America: The Search for the Effects of Information Technology at the Individual Level," *Political Research Quarterly*, vol. 54, no. 1, 2001, pp. 53-67.
- [23] K. Mossenburg, C. Tolbert, and M. Stansbury, *Virtual Inequality: Beyond the Digital Divide*, George Washington University Press, Washington, D.C., 2003.
- [24] J. C. Thomas and G. Streib, "The new face of government: Citizen-initiated contacts in the era of E-government," *Journal of Public Administration Research and Theory*, vol. 13, no. 1, January, 2003, pp. 83-102.

- [25] P. G. Harwood and W. V. McIntosh, *Virtual Distance and America's Changing Sense of Community*, New York, NY: Routledge, 2004.
- [26] Pew Internet Project, "Tracking Online Life: How Women Use the Internet to Cultivate Family and Friends," Pew Internet Project Report, Accessed June 1, 2005; www.pewinternet.org/reports.
- [27] G. Peng, "Critical Mass, Diffusion Channels, And Digital Divide," *The Journal of Computer Information Systems*, vol. 50, no. 3, 2010, pp. 63-71.
- [28] L. A. Jackson, A. v. Eye, G. Barbatsis *et al.*, "The impact of Internet use on the other side of the digital divide " *Communication of the ACM* vol. 47 no. 7, 2004 pp. 43-47
- [29] F. Bélanger and L. Carter, "Trust and Risk in eGovernment Adoption," *Journal of Strategic Information Systems*, vol. 17, no. 2, 2008, pp. 165-176.
- [30] W. Lyons and R. Alexander, "A Tale of Two Electorates: Generational Replacement and the Decline of Voting in Presidential Elections," *Journal of Politics*, vol. 62, no. 4, 2000, pp. 1014-1034.
- [31] S. Henry, "Can remote Internet voting increase turnout?," *Aslib Proceedings*, vol. 55, no. 4, 2003, pp. 193-202.
- [32] C. L. Schaupp and L. Carter, "E-voting: From Apathy to Adoption " *Journal of Enterprise Information Management*, vol. 18, no. 5, 2005, pp. 586-601.
- [33] J. Sanders, "The Future of Voting: Researchers Explore the Social and Technical Issues of Voting Via the Internet," Georgia Institute of Technology Report, www.gtresearchnews.gatech.edu/newsrelease/VOTING.html, Accessed March 12, 2006.
- [34] K. Coleman, *Internet Voting*, Order Code RS20639, CRS Report for Congress, 2003.
- [35] M. Tomz and R. P. Van Houwelling, "How Does Voting Equipment Affect the Racial Gap in Voided Ballots?," *American Journal of Political Science*, vol. 47, no. 1, 2003, pp. 46-60.
- [36] B. Al-Rababah and E. Abu-Shanab, "E-Government And Gender Digital Divide: The Case Of Jordan," *International Journal of Electronic Business Management*, vol. 8, no. 1, 2010, pp. 1-8.
- [37] E. Rogers, *Diffusion of Innovation*, New York: The Free Press, 2003.
- [38] V. Venkatesch, M. Morris, G. Davis *et al.*, "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly*, vol. 27, no. 3, 2003, pp. 425-478.

Free-Libre Open Source Software as a public policy choice

Mark Perry*

Faculty of Law - Faculty of Science
University of Western Ontario
London, Ontario
mperry@uwo.ca

Thomas Margoni*

Faculty of Law - Faculty of Science
University of Western Ontario
London, Ontario
tmargoni@uwo.ca

Abstract—Free Libre Open Source Software (FLOSS) is characterised by a specific programming and development paradigm. The availability and freedom of use of source code are at the core of this paradigm, and are the prerequisites for FLOSS features. Unfortunately, the fundamental role of code is often ignored among those who decide the software purchases for Canadian public agencies. Source code availability and the connected freedoms are often seen as unrelated and accidental aspects, and the only real advantage acknowledged, which is the absence of royalty fees, becomes paramount. In this paper we discuss some relevant legal issues and explain why public administrations should choose FLOSS for their technological infrastructure. We also present the results of a survey regarding the penetration and awareness of FLOSS usage into the Government of Canada. The data demonstrates that the Government of Canada shows no enforced policy regarding the implementation of a specific technological framework (which has legal, economic, business, and ethical repercussions) in their departments and agencies.

Keywords-Public policy, data analysis, derivative works, licences, public sector, policy recommendations.

I. INTRODUCTION

Free-Libre and Open Source Software (FLOSS) is about freedom, access, transparency, and accountability [1]. However, at many levels FLOSS is considered for use by provisioning departments only on the basis of the asserted cost reductions that it may bring. This is reflected in the fact that the acronym is sometimes replaced by other non-standard terminology. For example, the Canadian Government recently made a 'No Charge Licensed Software' Request for Information, thereby avoiding the FLOSS terminology [2]. Although terminology may seem like a minor issue, many advantages of FLOSS are found in software that fits strictly within its definition, and not within acronyms and labels that could look similar to a non-discerning reader. Examples of this include 'free-ware', 'share-ware' and 'no charge software'.

In this paper, we look at the attributes of FLOSS that are linked with the specific legal requirements under which it is distributed, as well as the specific framework under which the software is developed. We focus on the public sector, i.e., government, public administrations, federal or provincial agencies. FLOSS is also successfully employed in the private sector, however, we have not analysed the

idiosyncrasies of the corporate and business initiatives [3]. The following sections are included: basic terminology; licence compatibility; benefits for governments and public agencies to adopt FLOSS, and whether such beneficial aspects are caught by the Government of Canada (GoC). Finally, we present the data of a recent survey that suggest that the GoC is not taking advantage of the benefits this innovative phenomenon would include.

II. TERMINOLOGY

When dealing with FLOSS, two main concepts of the taxonomy should be separated: "Free-Libre" and "Open". There have been disputes between these two approaches, which lie behind similar code development methods: "[T]he obvious meaning for the expression 'open source software' is 'You can look at the source code'", and most people seem to think that's what it means. That is a much weaker criterion than free software, and much weaker than the official definition of open source. It includes many programs that are neither free nor open source. Since the obvious meaning for 'open source' is not the meaning that its advocates intend, the result is that most people misunderstand the term" [4].

Opposing the Free Software Foundation (FSF) position is the Open Source Initiative (OSI) position: "it was decided at a conference that it was time to drop the moralising and confrontational attitude that had been associated with free software in the past, and sell the idea strictly on the same pragmatic, business-case grounds that had motivated Netscape. They brainstormed about tactics and a new label Open source, contributed by Chris Peterson was the best they came up with" [5].

Despite advocates of different points of view towards these positions, and the different weight that each position gives to ethical and business concerns, the difference lies more in a philosophical level of abstraction rather than at a substantial level. Further, if it is true that Open Source is appealing for the private sector, then in the public sector (whose primary goal is not to make profit) a definition that includes 'Freedom' seems to fit perfectly.

However, more important than definitions, when a provisioning department is deciding the type of software that

should be used it must look at the legal requirements that establish the terms of use of the software, i.e., the licences.

A. Licences

The first and most famous of these agreements is the GNU General Public License, (GNU GPL or simply GPL). The first version of the license was established in February 1989 and was quickly followed by a second version in June 1991. The wording "version 1 (or 2, or 3) or any later version" is often used. The current version 3 dates back to June 2007. Between 50% and 70% of all FLOSS projects are released compliant with one of the versions of the GPL [6]. Although the GPL covers more than a half of all major FLOSS projects, there is a plethora of other licences that are commonly used. The Open Source Initiative (OSI) reports 55 OSI approved licences. The Free Software Foundation (FSF) reports 43 Free Software approved licences compatible with the GPL and 39 licences that are deemed Free Software compliant but not GPL compatible.

In many cases, to be GPL compatible is a matter of version and different versions of the same licence can be either GPL compatible or not. It is interesting to note how, technically speaking, version 2 (v2) and version 3 (v3) of the GPL are not compatible with each other. The issue of compatibility is of paramount importance when a public agency decides to develop its own software tools, either because it must be aware of the licensing already set up for existing code, or if they want to create new code, or open-sourcing an existing one, they need to be aware of the differences, opportunities and limits connected with different licences.

B. Updating the GPL

While GPLv3 aims to maintain and further the principles of GPLv2, technological pace and new threats to FLOSS obliged GPLv3 drafters to insert specific provisions to address these new issues. The drafting of v3 was a socially distributed effort and many criticisms that emerged on early drafts regarding new requirements have found their place in the current version. In particular, besides a general improvement in terminology, especially for internationalisation [8], and a more detailed section on definitions, v3 contains the following new sections: Digital rights (or restrictions) management/TiVoization (sec. 3); Licensing patents (sec. 11); Short term compliance (30 or 60 days) before automatic termination of the licence (sec. 8); Clarification regarding peer-to-peer distribution of object and source code (sec. 9); And a less monolithic general framework (sec. 7).

Unfortunately, since both v2 and v3 are strong copyleft licences (see *infra*), this excludes their mutual compatibility. If a work is released under GPLv2 and somebody wants to distribute a modified version of this work, this modified version shall be under GPLv2. The same happens with GPLv3. The problem arises when a modified version is

based on more than one program, where at least one is under v2 and the other under v3; in a case like this, there is no legal way to make them compliant.

However, it must be kept in mind that the copyleft requirement applies only to modified versions. That is to say, it is possible to distribute a software package (e.g., an operating system) containing programs under different licences, even those not compatible with each other. This is what happens with the most common GNU-Linux distributions, where the Linux kernel distributed under GPLv2 happily coexists with other tools or applications released under GPLv3 (or many other non-compatible licences). Cases like proprietary Loadable Kernel Modules (LKMs) or binary blobs, i.e., those object files loaded into the kernel without a publicly available source code, are considered borderline cases, meaning that in some limited circumstances their use is accepted because it is recognised that they do not form a derivative work of the kernel.

GPLv3 may be combined with important licences that are not compatible with the former version, namely the XFree86 (v. 1.1), the Apache (v. 2.0) and the GNU Affero GPL (v.3) licences. In particular, the latter is a GPLv3 licence specially aimed for network-interactive software, thus allowing users of web-applications to be able to receive the source code (technically speaking, to run a server is not an act of distribution).

C. BSD: licence and versions

A criticism of FLOSS licence regimes is as to the naming system. Law requires certainty in many aspects, including terminology. If versioning in regards to the GPL licence sounds confusing, then the Berkeley Software Distribution (BSD) licence offers a much more challenging example.

BSD is a FLOSS licence (FSF recognises it as Free Software) but it is a permissive licence, meaning neither strong nor weak copyleft (see *infra*). The BSD licence should, more correctly, be referred to as an entire family of licences, rather than only one. The main reason for this classification is the multiple modifications that the original licence has suffered, thus when software is distributed with a BSD licence, it is of pivotal importance to know the exact version.

The *new*, or *revised*, or again *3-clause* BSD licence is clearly Free Software and GPL-compatible. However, this compatibility does not exist when referring to the *original*, or *old*, or *4-clause* BSD licence. In the latter, an extra clause imposes a requirement that makes it incompatible with GPL. This clause, also called the 'advertising clause', requires authors of derivative works to include an acknowledgement of the original source, which, could lead, and sometimes has, to many pages of acknowledgements. Each of these sets is basically composed of the same licence with slight variations in the wording.

In addition to these two main categories, the BSD family has grown. Among the more widespread, there is the NetBSD, the 2-clause BSD (similar to the MIT), the FreeBSD, and the Clear BSD. All of these variations of BSD are usually GPL compatible, though this does not mean that their actual wording should be ignored. On the contrary, it is important to know, for instance, if the licensor reserves the right to sue you for patent infringement or not (see the Clear BSD).

III. COPYLEFT'S REACH

In certain production circumstances the use of some types of FLOSS licences are perceived to be problematic. Some creators and distributors of 'packaged' software have detracted from GPL due to its so-called viral nature. The word 'viral' is unfortunate, as it projects a negative connotation upon a clause in a legal document. To see such a characteristic with favour or not is a matter of personal choice, but as a matter of legal definitions, it should be referred to with a more neutral epithet: here we will refer to this characteristic as 'persistence'.

A. Strong copyleft

One of the characteristics of the GPL is its strong copyleft status. Strong copyleft licences are those licences that require any subsequent distribution of the work, or a modified version of the work, to be under the same licence. A new program based on a GPL licensed code must be distributed under the GPL. This persistence has represented a major issue in the field of FLOSS. Some supporters of FLOSS models that are based on non-persistent, but permissive licences (i.e. similar to the BSD), have accused the GPL of cannibalising BSD software: while the permissiveness of BSD-like licences permits protected code to fall under the GPL, however, the converse is not possible due to the copyleft requirement of the GPL. GPL supporters argue that it is not the GPL cannibalising code, but rather the BSD that permits every type of licence and even proprietisation of BSD software. Copyleft, proponents say, is necessary to protect and foster the development of a "contributory commons".

B. Weak copyleft

There are some FLOSS licences that are copyleft but their requirements are not as strong as the GPL. Consequently, they are labelled 'weak copyleft' licences. Examples of this category are the LGPL (where 'L' stands for 'Lesser'), the Common Public Licence, and the Mozilla Public Licence. These licences allow combining the software with other types of licensed software without the necessity of distribution under the same licence, but this does not mean that they don't need to be compatible: the CPL and the MPL, unlike the LGPL, are not GPL-compatible [17][18]!

The difference between weak copyleft and permissive regimes is the possibility to combine, for example, LGPL

and closed-source software without turning the output into LGPL. However, such a feature applies only to linking activities. If a piece of software released under the LGPL is going to be modified in order to produce a new version or a fork (or every other activity but linking) the new software will have to be released under the same licence (or eventually the standard GPL), thereby fulfilling the copyleft part of the label. Since persistence only works for some types of activities and not others (linking in LGPL case), such a copyleft regime is not strong, but weak.

C. Derivative works

This brief overview of the compatibility issues regarding weak copyleft licences necessarily brings us to the concept of the derivative work. The aim of this paper is not to provide an exhaustive analysis of what this concept could legally mean, as, due to the ubiquitous nature of the Internet, such a survey would have to be completed for all jurisdictions. What we analyse is the meaning of derivative works in the case of a program being linked by another one, usually a library, and observing the unique consequences derived from the wording of the GPL in cases of dynamic linking and static linking, and ultimately whether this distinction does, or should, matter.

A program statically linked with a library, creates a new, modified work. If either piece of software is released under the GPL, the derived work (the program statically linked with the library) shall be under the GPL. Since part (a 'substantial part') of the library is copied into the executable of the program at compile time, the output is the program plus the library (substantial part thereof), and thus a new work based on the two precedents. If one of the two works is released under the GPL, the new derivative work will have to be under the same licence as per the GPL requirement.

A more complicated case is that of a program that is dynamically linked with a library. In such a case, no substantial part of the library is present into the executable, so besides being connected, the latter is not a derivative work. However, while the FSF and GNU agree with this general framework, they further affirm that when a dynamically linked library and program share a more 'intimate' existence, they should be considered once again a derivative work. More precisely "[i]f the program dynamically links plug-ins, and they make function calls to each other and share data structures, we believe they form a single program, which must be treated as an extension of both the main program and the plug-ins, while if the program uses fork and exec to invoke plug-ins, then the plug-ins are separate programs, so the license for the main program makes no requirements for them" [12].

A complex, borderline case, where in presence of a dynamic linking structure the FSF and GNU support the thesis of a derivative work (extension of the two codes), due to the relation between the two pieces of software, which is so strict (reciprocal function calls, sharing of data

structures) that, even in the absence of a substantial portion of the source code of one of the programs into the other, the functional result is not far from it. Nonetheless, FSF and GNU recognise the presence of undefined areas: "If the program dynamically links plug-ins, but the communication between them is limited to invoking the main function of the plug-in with some options and waiting for it to return, that is a borderline case" [12].

The Canadian Copyright Act, for example, gives little guidance for such situations (as is common in almost all jurisdictions), only generally reserving the right to "produce or reproduce the work or any substantial part thereof in any material form whatever... and to authorise any such acts" to the rights holder, and adds specific cases of adaptation, that are inapplicable to software (sec. 3, especially d. and e.). The Act is in much company with the United States and many European countries, as the legislation does not deliver a granularity that is fine enough to deal with a library dynamically linked to a program with which it shares system calls and data structures. This is probably the better situation as the legislation is meant to provide general and abstract rules, leaving it for the interpreter to adapt them to a specific case. Here, it might be relevant to recall that in Canada, while rewriting a computer program from one language into another could be interpreted as a translation under certain circumstances [13], compiling the source code into object code is an act of reproduction [14]. The main consequence of this distinction is that to compile a program (being either an application or library) requires the right to reproduce [15].

Determining exactly what a derivative work is within linked computer programs is a contentious issue. It obviously depends on the legal system where one claims protection. However, there are claims that the issue of static and dynamic linking is a red herring and what really matters is not the name of a specific program or call (mkisofs, ld, exec, or the like), which undoubtedly has functional consequences, but the specific grade of dependence or independence between the two programs. This relationship establishes whether the output is a derivative work or a mere aggregation [16]. The latter approach introduces some uncertainty because it suggests a case-by-case analysis, rather than a "static = derivative" equation. The door is still open to deeper analysis on this issue, as is evidenced by the comments of one of the fathers of the Linux kernel, Linus Torvalds, when he said that 'there was not much need for the LGPL' [16].

IV. NOT JUST MONEY

Usually, obtaining FLOSS requires nothing more than an internet connection. Inherent in both the FSF and OSI models is the ability for anyone to access the code. There are no royalties to be paid, no required tie-in to service contracts, and no up-front acquisition costs. In addition to the economic aspects, there are many advantages to adopting

FLOSS: although price is not the primary advantage, it is often viewed as such, which results in FLOSS being incorrectly assimilated with other non-immediate-fee software. Such naivety should be avoided, especially when the interested entity is a public body whose main objective is to offer public services and not to make a profit. An important aspect of FLOSS is the availability of the source code. This means that the ability to modify and redistribute improvements is a contractual obligation. This specific feature is common to all licences fitting in the category and therefore entails legal, economic, technical and social consequences. We will explore nine examples of these consequences, which are particularly pertinent to governments' use of code.

A. Accountability and transparency

Source code availability permits users to know what the program does at a depth that would otherwise be impossible. Without the source code one can only deduce what the program does through expensive and time consuming reverse-engineering without ever having the opportunity to know all of the original code. Source code availability is critically important for software applications in the core areas of government (such as national defence and homeland security, financial and economic administration, health databases, and wherever privacy and reliability are deemed substantial), as well as the fundamental infrastructure of public administration [7]. The possibility for the general public to understand and to rely on the activities of public bodies is directly connected to the use of a software model that is transparent and accountable (e-Democracy). This is a cornerstone in providing citizens with the guarantees of a fair, efficient and impartial administration of the public good. A good example of this can be seen in electronic voting systems [9].

B. Interoperability

The availability of the source code allows for better interoperability with other applications. If an application is not perfectly compatible, the availability of the source code, combined with the contractual permission to use and adapt it, permits modifying the code with interoperability as the likely result. If there is FLOSS and closed source software (proprietary), greater compatibility is possible in contrast to the case of two closed source software provided by different suppliers [11]. This is of particular interest for public bodies since it grants the possibility to share resources among many different departments and agencies. Despite being autonomously organised, public bodies do not suffer from the strict competition that affects corporate entities. This is what allows for strong scale economies with significant savings for the whole public administration and, consequently, for taxpayers. In some jurisdictions (e.g., Italy) this is prescribed by law [10].

C. Avoid lock-in

Vendor lock-in is the phenomenon that causes customer dependency on a given vendor with regard to a specific good or service. Switching vendors has high transaction costs connected with technological and organisational changes and, in some cases, penalty clauses due to early cancellation of a supply contract. These 'switching' costs are pernicious to the market and can represent strong barriers to entry. With closed source software the customer is generally bound to a specific supplier, both contractually and technologically. As an example, in the case of freeware, a typical business model that is sought is lock-in. In this case, once the lock-in has occurred the software distribution model can switch to a traditionally priced one since the transaction costs connected with the migration to another type of software are prohibitive [35]. In the case of FLOSS, both the licence and the technology allow for a supplier-independent business model [19]. For public administrations, it is mandatory to choose suppliers that are able to grant reliable services at good prices and provide for long-term maintainability (public administrations usually last longer than private companies). However, it is also critically important that if a better offer or player enters the market the public body should not be impeded from transitioning to the more efficient solution. This will immediately reflect in the cost and quality of service enjoyed by citizens.

D. Long-term maintainability and technological ecumenism

A public administration cannot discriminate the public based on the type of software used. A private company has the option to use closed source software compatible with 85% of the software used by citizens and incompatible with the remaining 15%: the market will decide if this decision pays. However, a public administration cannot exclude 15% because they chose a different operating system. FLOSS is the solution that grants the highest compatibility, thus minimising the phenomenon of technological exclusion by both FLOSS users and closed source software users. FLOSS also means Open Formats, which are those formats that are publicly documented so as to permit anyone to implement programs (both FLOSS and closed source software) that can optimally use, store, and retrieve such data. This is another manifestation of the absence of lock-in problems [20].

Many times the reason for staying with an old supplier (which usually means also old technology) is that they are the only ones owning the (closed) format technology enabling data retrieval.

E. Security and error correction

Security is not a static concept that can be reached once for all, nor easily maintained. FLOSS is known not only for the transparency and accountability of its code, but also for its stability and intrinsically greater security. It is a common principle in computer science that the security of a system

depends on the quality of its structure, not on its obscurity (a variation of Kerckhoffs' principle in cryptography). Only if the source code is available is there the possibility for quick bug-correction and exploit-detection. In the case of FLOSS, the pace at which the stability level of the code grows is much faster than in other types of software, where it is necessary to wait for the supplier security updates [21]. A sound and accountable technological infrastructure is a key point for all e-Government and Government-to-citizens (G2C) initiatives, where the reliance of citizens is fundamental for the success of the electronic offered service.

F. Democracy and pluralism

FLOSS in the public sector is more generally a matter of democracy [22]. In case studies such as those involving electronic voting machines, or "technology enhanced trials", the people need to rely and trust not only in their representatives and the courts, but also the process of electing the candidates or of condemning the guilty [22]. FLOSS seems to epitomise those basic principles commonly found in many constitutional and fundamental charters, of fair administration of the justice, of pluralism, of freedom of expression, and of access to information and knowledge. A long list of public administrations around the world has already started, or is seriously considering migrating from proprietary to open code. Among the most successful initiatives is the German city of Munich with the LiMux project. They report to already have 1,200 workstations migrated to Gnu-Linux, 12,000 using Open Office, and 100% of the city administrations using Firefox and Thunderbird [33]. Another interesting case study can be seen in Spain, where the different comunidades autonomas (Spanish provinces) have different levels of FLOSS implementation, all coordinated by a specific constituted public agency: Centro Nacional de Referencia de Aplicacion de las TIC basadas en fuentes abiertas (www.cenatic.es). The "dollar price" connected with the absence of royalties is only one potential saving: "Contrary to what is often assumed, cutting costs was not the main reason for the migration. The motivation is independence [...] now we're able to decide on our own how we want to spend our IT budget in the long run [...]" [23]. This approach is also consistent with the Open-Government instances which hold that the business of government and state administration should be opened at all levels to effective public scrutiny and oversight. To translate the Open Government principles in programming terms, involves the use of FLOSS.

G. Portability to other languages

The possibility, both technical and legal/contractual, to translate software into any language is of paramount interest if due importance is given to linguistic and cultural pluralism. Although this sounds more like a European, Asian or African based argument, also in America (both North

and South) language plays a key role in the protection of indigenous and traditional knowledge and in effectively reducing the phenomenon of 'digital divide'. This feature of FLOSS may be easily confirmed by checking the language packs or language ports of some of the most widespread projects and comparing them with similar non-FLOSS products. For example, Firefox v3.1 has 62 language ports and 78 different language packs, among which many minority languages are present. On the other hand, Internet Explorer 8 has 3 language selection possibilities. Opera has 41 supported languages, Chrome 44, and Safari 18. The difference is even greater with the office suite: OpenOffice.org has 123 supported languages, while Microsoft Office has 35. Finally Outlook 2007 is available in 14 languages, while Thunderbird 2, in 39.

The reason for this difference in language policy clearly resides not only in the sensibility of the project managers but rather in the declared legal, contractual and technological features of FLOSS.

H. Fostering competition

Another major advantage of FLOSS is that it creates and favours a more competitive ICT environment usually populated by many local Small and Medium Enterprises. Licence fees, from a microeconomic point of view, represent huge barriers to entry the markets, thus favouring monopolistic and oligopolistic situations. As it has been reported [23][24], a public administration investing in FLOSS solutions is usually interested in hiring or contracting with local ICT companies for services like updating, maintenance, training, and customisation. In this way the immediate benefit for local economies is apparent.

I. Total Cost of Ownership

A major saving in using FLOSS is royalties. Quite simply, there are none. During the 2005-2006 fiscal year the Canadian government spent 425,602,327CAD on software licence fees [25]. Clearly, this represents a huge amount of money. Unfortunately, using FLOSS does not mean that there are no costs whatsoever. For example, due to the so-called *alumni effect*, many people have learned how to use computers through non-FLOSS applications. This means that even though FLOSS solutions nowadays are user-friendly enough, there are still some costs connected with migration, such that in the short term, it is not always true that there are significant monetary savings. Nonetheless, there are savings that become substantial in the medium/long term and that will endure and increase with time. Some of these savings have already been identified (no lock-in, enhanced security, etc), while others are more concealed (such as cross-platform availability, maintenance, updating and long-term upgrading, compatibility with 'older' hardware, etc) [26].

Taking into account all of these variables provides a better portrait of the actual benefit in terms of economic

and financial costs. As demonstrated in many studies, the huge Total Cost of Ownership savings resulting from the use of FLOSS is undisputed. The public sector reports from Sweden show yearly savings of billions of dollars [27]. Another benefit is that the agency taking the FLOSS route will need to spend money on the development of internal staff skills, which means that the skill base for the organisation will be improved, giving better overall support for the department and creating a greater pool of skilled persons in Canada.

V. THE SURVEY

During this research, we conducted a survey on the use of FLOSS by the different Canadian ministries and other Canadian public departments and agencies. We contacted a total of 53 Information Technology (IT) departments. We decided to only target the category of IT departments in the agencies, since this allowed us to access the real technological situation of the department. In this survey we are not interested in what people do, whether given Canadian civil servants use or not FLOSS. Our survey was focused, and our data demonstrate, the use of FLOSS in Canadian governmental and other public agencies departments. Of the 53 IT departments contacted, 20 agreed to participate in our survey, either by live interview, phone interview, or through email. We preferred the live interview because it allowed keeping track of more variables than what appeared on the answer sheet. Interviews also allowed the operator to record the immediate reactions to the questions, which was not possible when using an emailed questionnaire. The participation rate was 38%, which although not very high, places itself at the top average of similar studies. For example, samples of reference participation data are 23.8% in the UK and 18% in Germany, without subdividing by the sector. A slightly higher participation rate is observed if considering only the Public Sector (37% and 29%, respectively). It must be noted that a third surveyed country, Sweden, has much higher participation ranges in every sector at approximately 60% [28]. Another seminal study in this field, Flosspols, reports an average participation of 22.8%, even though the variations from country to country are very high [29].

We had hoped that the participation rate to be higher than what was achieved because our respondents were public administrations, public departments, and ministries of the Canadian government and, as such, we stressed our identity (a renown Canadian University) and the fact that the study was funded and promoted by an important Canadian public agency (Social Sciences and Humanities Research Council, SSHRC). Unfortunately, this proved to be an incorrect assumption, as our viewpoint was not widely shared. The questionnaire was formed by 11 multiple-choice questions. The 12th question was left open so that the respondents

could add whatever they deemed important that was not covered in our interview.

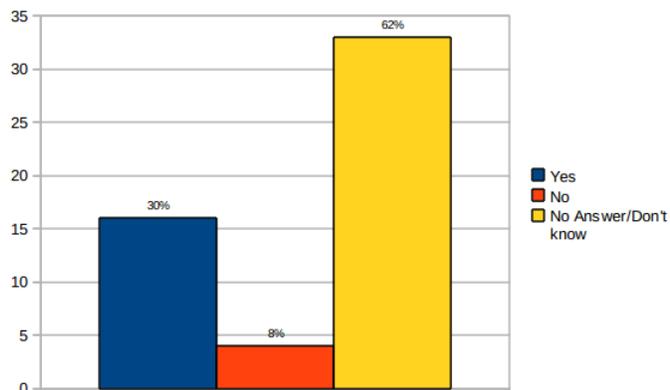


Figure 1: Do you use any Free-Libre Open Source Software in your department?

The results show that FLOSS in the technological infrastructure of the Canadian government is used only partially. In fact, its use is limited and not exclusive to what is referred to as 'Desktop purpose machines', but rather is utilised only in a very limited amount of cases on servers. Regarding Desktops, the fact that it is used only partially may be easily explained by a simple consideration. While it is uncommon to have non-FLOSS applications running on FLOSS Operating Systems (OS), the contrary is quite common. Such an inference is confirmed by the results of our next question, regarding the type of FLOSS (identified by name) used by the respondents.

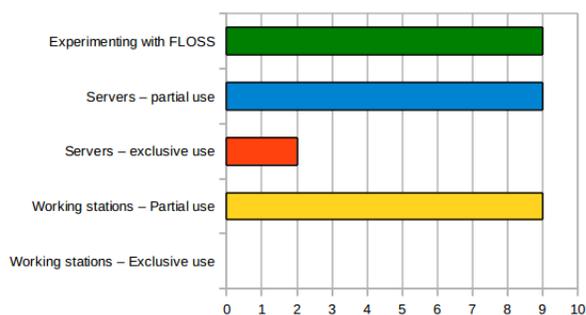


Figure 2: What type of FLOSS is it in use in your department?

Among the programs under consideration, Mozilla appeared most frequently (our questionnaire includes all the different varieties of projects that such a brand encompasses: Firefox, Thunderbird, SeaMonkey, Camino, Fennec, etc.) with 17% of the respondents reporting its use. Others widely used FLOSS programs included administration and database management or programming tools such as PHP (16%), Pearl (13%) and MySQL (14%) and interoperability tools such as Samba (8%). OpenOffice.org was used by only 5% of respondents, which might be explained by compat-

ibility issues and *alumni effect* (see above). Among the less used programs were graphical desktop environments such as Gnome (6%) and KDE (5%). Graphical desktop environments are those programs used to provide users with a Graphical User Interface (GUI) and are much more platform dependent than other reported applications. In fact, while it is possible to run either Gnome or KDE on some other Unix-like distributions, it is not possible to run them on other platforms such as Microsoft Windows (which has its own GUI). This portrait is consistent with the data gathered, which suggests a strong usage of Microsoft Windows as the main Desktop operating system (see Fig. 4), on top of which, with varying degrees, FLOSS tools are installed. The appearance of Mozilla as the most used software is shown in the figure below.

The reason why there is no score amid the exclusive use of FLOSS on desktops, while there was a total of 5% and 6% of respondents declaring that they have FLOSS GUI distributions on their machines (as mentioned, are usually run on FLOSS OS, though they might be run also on some other Unix-like non-FLOSS OS) might be explained by the so called Dual Boot configuration. In development environments and amongst experts, quite often a single desktop machine is configured in a way that, when the power button is pressed, a program called Boot Loader opens and asks what operating system (and/or kernel) should be loaded. In this case, many operating systems can reside simultaneously on the same machine without the possibility of running contemporaneously (virtualisation is another issue).

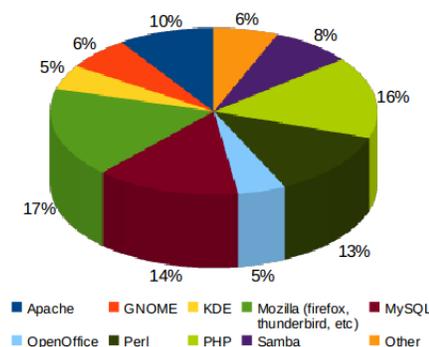


Figure 3: What kind of programs or applications are in use in your department?

That being said, such a configuration is relegated by a large extent to experimental uses (see Fig. 2), demonstrating that one of the two OS is not meant for productivity. In the present case, this most likely means that FLOSS might be installed but not used.

It is more difficult to explain the reason why only a very limited number of respondents (10%) declared to use Apache on their servers. Apache is a web server and is very widely distributed (official statistics of June 2010 report a total

usage of more than 54% of the World Wide Web [32]). The reason for this might be found on a taxonomic level. In fact, Apache is not usually associated with FLOSS, especially with Gnu-Linux (it is released under the Apache licence, a FLOSS licence, but not GPL), and therefore a perception might exist where this type of tool does not pertain to the FLOSS family. Our questionnaire was purposely vague in asking what kind of FLOSS tools are in use, and may have resulted in the respondents discarded Apache if they do not believe it to be FLOSS. If this is the explanation of why our data do not mirror the general market situation, it is noteworthy that specifically trained IT departments are not aware of this misconception which is taxonomic in its origins, but very pragmatic in its consequences. Of course, it may simply be that the representative of the department did not know.

Our data also suggests that the use of OS in our analysis reflects the situation in the general market, where the dominance of Microsoft Windows (client side) is clear (world market data report 85% to 90% of MS Windows usage on clients [30][31]). In our enquiry, 48% of respondents declared that they use Microsoft products in one of its variants (98, XP, NT, Vista, 7, etc.). Gnu-Linux (in any of its variants: Debian, Red Hat/Fedora, Ubuntu, Slackware, Mandrivia, SuSe, etc) followed at 21%, then MacOS/X at 12%. In the case of Macintosh, the data closely mirrors the general data reported by the referenced statistics, however, the numbers regarding Windows and Linux do not accurately reflect the same data. In our data Windows achieves a 48% (contrasting with 85% to 90%) and Linux, a 21% (contrasting with 3%). Our data might suggest that there is wider use of the Gnu-Linux OS in the Canadian public sector; however, we must temper such an inference as our survey asked what types of OS are run on the (theoretically thousands of) clients managed by the respondents.

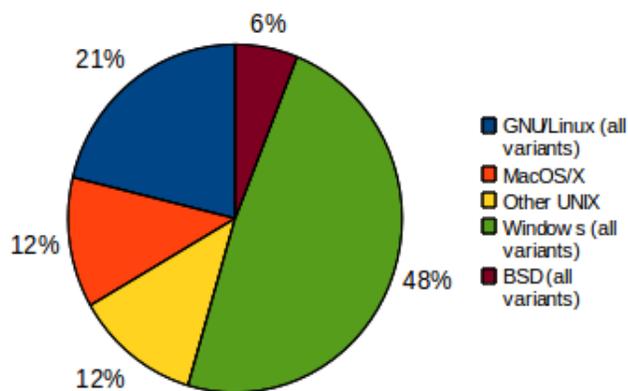


Figure 4: What kind of operating systems are installed in the department?

It is well known that FLOSS solutions, are not commonly used, as main operating systems on desktops. Nevertheless, an overwhelming majority of those interviewed agreed that a higher deployment of FLOSS would be beneficial to their department (65%). Less than one fifth of the respondents did not agree with this sentiment, it must be noted that a higher deployment does not equate to integral substitution. Out of the 65% of respondents in favour of a wider usage, only 11% would welcome a total substitution of their current software with FLOSS. Conversely, 78% would prefer a coexistence of proprietary and FLOSS.

It is interesting to note that that access to the source code is not the most important parameter to users that answered that they would welcome the use of FLOSS in their desktop systems (only 27% believe this). A far more important consideration was the price: 75% of respondents agreed that pure access to the source code (which includes the possibility to modify and redistribute it), not combined with the elimination of costs associated with licensing would render FLOSS unattractive to their departments.

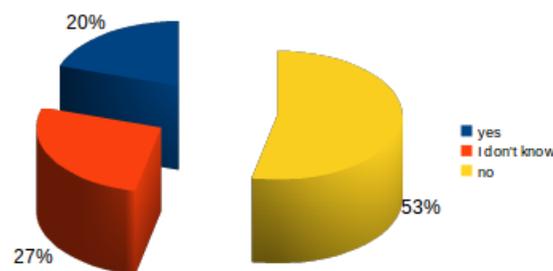


Figure 5: Is access to the source code an important factor for the department?

Regarding the more technical aspects, it was observed that a vast majority (85%) of interviewees acknowledged that FLOSS has higher customisation capabilities. Again, customisation is a characteristic that streams directly from the source code availability; however, the respondents (which were carefully identified in the IT departments) did not see such a connection. On the contrary, a majority of respondents believed that the main advantage of FLOSS is the fact that there are no licence fees. Another counter-intuitive result is connected with software reliability: 64% of the respondents believed that FLOSS is less reliable than non-FLOSS software. As discussed above, reliability may be considered an open issue, with strong advocates on both sides, supported by studies and data. Interestingly, among the surveyed category, there is a significant concentration of supporters of one specific view of the matter. More in

keeping with the general perception were the respondents' views in terms of the ease for normal (i.e., non-technical) users in using FLOSS: 74% agreed that FLOSS is more difficult or complicated at first use. However, 80% did not see migration and training as an impediment towards a wider usage of FLOSS. As such if an adequate migration and training process is scheduled, the initial unease connected with FLOSS should be overcome.

The data reported so far give us a contradictory portrait of the perception of FLOSS in the Canadian public sector. We have seen how some of the positions held by the majority of the respondents are significantly diverging from general market trends. For example, in terms of reliability, FLOSS is believed to be much more reliable and accountable by large numbers. The same divergence is observable in more technical aspects, such as the availability of the source code. In the technical arena, this aspect (not only in FLOSS cases – think, for example, of beta-tester, premium user/developer of specific applications, specific important institutional customers such as homeland security departments, etc.) is considered essential for a great majority of the benefits we have identified above in this study. In our survey, however, respondents did not put much importance on the availability of the source code. Conversely, respondents believed the monetary aspect to be much more important. This is notable, considering that the respondents were IT departments of public administrations whose main objective is not to make profit but to offer a public service.

This does not mean that a public administration should not conform its activity simply to principles such as economisation, efficiency and rational usage of resources. On the contrary, it is exactly for these reasons that they should implement solutions that grant longer-term savings both monetary and in the possibility to re-utilise and scale-economise the (software) resources they use/produce. A fair and balanced administration of the public good is a science based on principles such as rationality, efficiency, accountability, and transparency. The tools analysed here, for the reasons explained in the relevant sections, are the most suited to meet both economic and social requirements of the management of public bodies.

A possible explanation for the contradictory feedbacks in our survey – impressions supported by the oral comments and further notes expressed during the interviews – is that even in the IT manager area the situation is strongly polarised or even ideological. On one side there are the majority who are supporters of one model, i.e., closed source software, who are prejudicially against any alternative model seen as a threat to "their model and to their jobs." Affirmations such as "we do not use any Open Source Software, we pay our licences!" or "we have internal guidelines not to use any Open Source software, so I had to remove also some amusement machines I had in my office" help to clarify why we have used such evocative wording.

In the middle there is a small category (approximately as large as the category supporting FLOSS) of IT departments who are undecided in which model is better; they perceive the pros and cons of both models, and who – most of the time – simply do their jobs "with the tools at their disposal". One 'Chief Technology Officer' said that the GoC is interested in FLOSS and encourages its use even though there is some uncertainty regarding the Intellectual Property issues and connected responsibilities.

There are also overt supporters of the FLOSS model; a minority who are strongly motivated and use FLOSS in their departments. These people are either working on the technical side (server, database management, programming, etc.) or in productivity workstations. These subjects are more sensitive to issues such as the availability of the source code, though do not delve deeply into the reasons why a public administration, more than a private corporation, should implement FLOSS solutions.

However, IT departments should be concerned mainly with technical decisions, while the more substantive ones should come from representatives of the decision-making bodies whose subjects are specifically appointed and trained to evaluate a great many differences in variables in choosing a fundamental instrument like the technological infrastructure of a public body. Such a simplified tri-partition of the respondents is particularly important because they (the respondent IT departments) have identified themselves as the decision-making subjects when purchasing new software in a good deal of cases (43%), while the financial department decides in only 17% of the cases.

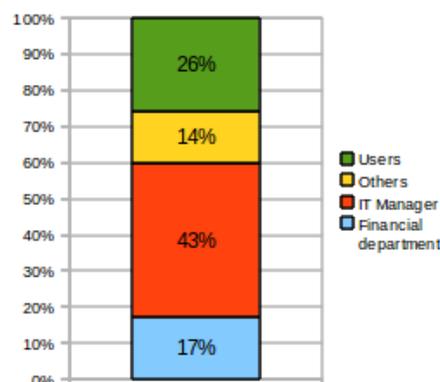


Figure 6: Whose opinion is decisive at the moment of purchasing new software?

VI. RECOMMENDATIONS AND CONCLUSIONS

The GoC is not taking advantage of the many different features that an innovative model of software production and distribution, such as FLOSS, offers.

The landscape that emerges from the data reported here is not encouraging. FLOSS has proven to possess a long list of advantages in comparison to other software development and

distribution models, especially in the public sector. The TCO is not the major concern, but together with the other FLOSS advantages exposed in this study, is an important one. Also in the TCO, FLOSS proved to be a strongly competitive and innovative model. However, easily discernible from our data is that the Canadian public sector completely lacks any coordination or guidelines in deciding which type of software tools to adopt in their departments. Even if guidelines existed, they are largely unattended, which paradoxically means that the Canadian governmental bodies distend their own rules.

Information Technology and, less frequently, financial departments decide which model to adopt, however, neither department are equipped or trained in making these decisions and therefore cannot take on the responsibility alone. As mentioned above, those departments are mainly concerned by financial or technical (customisation, stability, interoperability) issues – and it could not be otherwise. The problem is not so much what the Information Technology and financial departments believe but that in the majority of cases, they have the last word in deciding what to buy. Information Technology departments carry on a fundamentally important task and they have great experience. However, Information Technology departments cannot be left alone when making decisions regarding software use. Software is not a mere product but is a choice involving specific policy and political decisions that represent a specific set of values, public morality and ethics. Such political decisions need to be made by those whom have been elected who are ultimately responsible for the financial repercussions of software use in the public sector.

In short, there are many advantages that have strong economic value, in both the short and long term, that can only be eventualised by adopting FLOSS, with the technical and legal availability of the source code, and the possibility of its modification and redistribution. In addition to the monetary savings connected with the absence of licence fees, there are huge advantages that relate to the independence from software providers, the creation of a competitive market usually on a provincial, or regional level, transparency and accountability, the ease of customisation, digital inclusion and pluralism, and the further savings connected with scale economies amongst different public administrations [34]. The GoC should take full advantage of FLOSS in its technological infrastructure, because, as demonstrated, in many situations and at many levels it would be beneficial to Canada. Currently it is not making full consideration of FLOSS.

ACKNOWLEDGMENT

The authors would like to thank SSHRC, and the Law Foundation of Ontario for their support; also Constance Keunhee Yoo and David William George Morrison for their research assistance.

REFERENCES

- [*] The current article represents an extended version of the following paper: M. Perry and T. Margoni, 'FLOSS for the Canadian Public Sector: Open Democracy', in Digital Society, 2010; ICDS '10. Fourth International Conference on, pp.294-300, 10-16 Feb. 2010
- [1] R. Stallman, 'The Free Software Definition', available at <http://www.gnu.org/philosophy/free-sw.html>; All the websites cited in this work have been last visited during January 2011.
- [2] See 'Canadian Public Sector Contracts, Bids, and Tenders' web-site at <http://www.merx.com>.
- [3] See for example R. Goldman and R. Gabriel, 'Innovation Happens Elsewhere - Open Source as Business Strategy', Morgan Kaufman - Elsevier, San Francisco, 2005.
- [4] R. Stallman, 'Why "Open Source" misses the point of Free Software', available at <http://www.gnu.org/philosophy/open-source-misses-the-point.html>.
- [5] M. Tiemann, 'History of the OSI', available at <http://www.opensource.org/history>.
- [6] See 'Freshmeat GPL tagged projects' available at <http://freshmeat.net/tags/gnu-general-public-license-gpl>.
- [7] B. Schneier, *Open Source and Security*, in Crypto-Gram Newsletter, September 15, 1999.
- [8] For a deep analysis of legal issues surrounding FLOSS see L. Guibault and O. van Daalen, 'Unravelling the Myth around Open Source Licences', ITeR, The Hague, 2006.
- [9] H. Kaminski; L. Kari; and M. Perry, 'Who counts your votes? [VEV electronic voting system]', e-Technology, e-Commerce and e-Service, 2005. IEEE '05, Proceedings. The 2005 IEEE International Conference pp. 598-603, 29 March-1 April 2005.
- [10] See arts. 68 and 69 *Codice Amministrazione Digitale*, Legislative Decree 7 March 2006, n. 82, (as amended).
- [11] See for example the 'Microsoft Open Source Interoperability Initiative' at <http://www.microsoft.com/interop/>.
- [12] See the GPL/Plug-ins FAQ at <http://www.gnu.org/licenses/gpl-faq.html> GPLAndPlugins.
- [13] See Prism Hospital Software Inc. v. Hospital Medical Records Institute [1994], 97 B.C.L.R. (2d) 201, [1994] 10 W.W.R. 305, 57 C.P.R. (3d) 129, 18 B.L.R. (2d) 1.
- [14] See Apple Computer Inc. v. Mackintosh Computers Ltd., [1990] 2 S.C.R. 209, 110 N.R. 66, 30 C.P.R. (3d) 257, 71 D.L.R. (4th) 95, 36 F.T.R. 159.
- [15] D. Vaver, 'Translation and Copyright: a Canadian focus'; in E.I.P.R., 1994, 16(4), 159-166, at 160.
- [16] See the 'GPL only modules' thread at lkml.org, available at <http://lkml.org/lkml/2006/12/17/79>.

- [17] See the Mozilla Public Licence FAQ at <http://www.mozilla.org/MPL/mpl-faq.html>.
- [18] See the Common (now Eclipse) Public Licence available at <http://www.eclipse.org/legal/cpl-v10.html>.
- [19] B. Scott, 'Lock in Software', Open Source Law Publications, 2003.
- [20] B. Perens, 'Open Standard, Principles and Practice', available at <http://perens.com/OpenStandards/Definition.html>.
- [21] D.A. Wheeler, 'Secure programming for Linux and Unix HowTo', 2003, available under the GFDL license at <http://www.dwheeler.com/secure-programs>.
- [22] M. Perry and B. Fitzgerald, 'FLOSS as Democratic Principle', in International Journal of Technology, Knowledge, and Society, vol. II, 3, 2006, pp. 156 – 164.
- [23] K. Gerloff, 'Declaration of Independence: the LiMux project in Munich', Open Source Observatory and Repository (OSOR), European Commission's IDABC project.
- [24] Department of Finance and Administration 'A guide to Open Source Software for Australian government agencies', Australian Government Information Management Office, 2005.
- [25] M. Perry and T. Margoni, 'Floss for the Canadian public sector: Open Democracy', in Digital Society 2010, 2010, pp. 294
- [26] K. Wong, 'Free/Open Source Software – Government Policy', UNDP – Asia-Pacific Development Information Program, Elsevier, New Dehli, 2004.
- [27] The Swedish Agency for Public Management 'Free and Open Source Software – a feasibility study', Stockholm, 2003.
- [28] Source: *Free/Libre and Open Source Software: Survey and Study*, International Institute of Infonomics, University of Maastricht, The Netherlands, June 2002, available at <http://www.flossproject.org>.
- [29] See 'FlossPols Government Survey Report', Deliverable D3, Maastricht, August 25, 2005 - MERIT, University of Maastricht, available at <http://flosspols.org>.
- [30] See 'Operating System Market Share' at <http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8>.
- [31] See 'Global Web Stats' at <http://www.w3counter.com/globalstats.php>.
- [32] See 'January 2011 Web Server Survey' available at <http://news.netcraft.com/archives/category/web-server-survey>.
- [33] See LiMux web-page project at <http://www.muenchen.de/Rathaus/dir/limux/english/147197/index.html>.
- [34] Cenatic, 'Software de fuentes abiertas para el desarrollo de la administracion Publica Espanola - Una vision global', Observatorio Nacional de Software de fuentes abiertas, Badajoz, 2008.
- [35] B. Boyle, 'Open Source Software', Minister of State Service of New Zealand, New Zealand, 2003.
- [36] State Service Commission, 'Guide to legal issues in using Open Source Software v2', New Zealand Government, 2006.
- [37] Y. Benkler, 'The wealth of networks – how social productions transforms markets and freedom', Yale University Press, 2006.
- [38] J. Dickson, 'Use of open source: licenses and issues', in e-Commerce Law and Policy, March 2009, pp. 8.
- [39] B. Fitzgerald and N. Suzor, 'Legal issues for the use of free and open source software in government', in Melbourne University Law Review, 29, 2005, pp. 412.

Lessons Learned on Enhancing Performance of Networking Applications by IP Tunneling through Active Networks

Tomas Koutny and Jakub Sykora

Faculty of Applied Sciences

University of West Bohemia

Plzen, Czech Republic

txkoutny@kiv.zcu.cz, jsykora@students.zcu.cz

Abstract— In 1995, DARPA initiated a work on a programmable concept of computer networking that would overcome shortcomings of the Internet Protocol. In this concept, each packet is associated with a program code that defines packet's behavior. The code defines available network services and protocols. The concept has been called Active Networks. The research of Active Networks nearly stopped as DARPA ceased funding of research projects. Because we are interested in research of possible successors to the Internet Protocol, we continued the research. In this paper, we present an active network node called Smart Active Node. Particularly, this paper focuses on its ability to translate data flow transparently between IP network and active network to further improve performance of IP applications. We describe the translation mechanism, its possible use and discuss particular implementation aspects.

Keywords- Active Networks, Smart Active Node, IP tunneling, routing

I. INTRODUCTION

This paper extends the original paper [1] (Sections 1 – 6, sub-sections A and B of Section 10 and a portion of Section 11), as it captures recent advances on the project since Section 5, sub-section D.

Today, IP networks suffer from low scalability and deployment of new networking services is a subject to a long standardization process. A particular problem that lies within the scope of this paper is content delivery over IP, with respect to time-sensitive traffic – e.g video. Simply said, an effective solution is possible with a programmable network and for that task we need Active Networks [2, 3].

For example, a number of multi-cast schemes and protocols were developed. They try to do their best in optimizing a multi-cast tree to satisfy and guarantee a proper quality of service. These protocols cover multi-cast tree creation, optimization and client group membership management. This requires special hardware and software support from both network and clients. In fact, there is a complex overlay network built on a top of the IP network. While it addresses needs of today, there is still a room for an improvement [4]. We desire to be ready even for needs of tomorrow.

We do not aim at solving a particular problem. We try to build a general solution, which could be used to solve a variety of tasks and issues in a simple manner. To solve this

general problem, we did not decide to use a traditional network. Instead, we decided to use the concept that is known as Active Networks.

Active Networks is such concept, where every network node is active, when compared to passive elements used today. The activity is meant as the ability of a network node to process data in a context of application that created them. To make this possible, a packet has been superseded with a capsule. Along data, each capsule is associated with a reference to a program code. The code is downloaded through the network as needed and executed, as a capsule is run at a node. As the code executes, the node is able to handle the capsule's data in an application specific context. Thus, it is possible to teach the network new things on the fly. Note that capsule can route itself.

Active application is such networking application that injects capsules, which replace packets, into the network. In turn, a capsule may inject another capsule or an active application into the network. Both, application and capsule have an access to a server-offered API to use its functionality. Any custom code runs in a sand-box that is called Execution Environment.

As it is not realistic to assume that Active Networks would suddenly replace IP networks, these two networks would have to co-exist for a certain period. Thus, instead of awaiting a revolution in networking, we focus on adding more functionality to existing IP solutions via tunneling them into the world of Active Networks.

A preceding work is presented in Section 2. Section 3 explains our motivation. Fourth section describes proposed solution, while the next section is focused on implementation. Sections 6 and 7 focus on policy-based routing and worth-path routing. We discuss results in Section 8. The following section gives additional details on the most needed improvement – code execution. Related work is given in Section 10. Section 11 finishes with conclusion.

II. PRECEDING WORK

The PANDA [5] project was the proof of the concept of tunneling the IP protocol over an existing active network. The PANDA software ran on a top of ANTS [6] active-network server. It was a demonstration of active network's capability to transfer UDP datagrams transparently and to possibly recode contained video stream in order to satisfy bandwidth limits. The project, namely its PIC component, was implemented as a kernel module that communicated

through a BSD socket with the active network node. The node performed recoding and distribution of the stream. The demonstration showed that there is no need to modify neither source stream server nor the client software. By using active network as the underlying network, there was a significant spare of bandwidth and better QoS, was presented; QoS stands for Quality of Service.

III. MOTIVATION

In our research, we focused on problems, which showed up in the preceding work. They include IP tunneling, security, resource allocation and performance. We develop our own, general-purpose active network server. Its design addresses many shortcomings of the previous active network implementations. Preceding projects were generally aimed at particular problems, but without researching consequences among those. Capsule and application programming interfaces, performance and security have to be addressed altogether, not as standalone issues.

An important issue of active networking is performance. This is given by a number of possibly flowing capsules, and the need to execute their code in a sand-boxed environment to guarantee a required security. This is very challenging goal and no satisfying solution was present. Perhaps, this was the main reason, why DARPA ceased research funding on Active Networks.

However, thanks to our research ideas and comparison with other projects, we consider this issue as solvable. Therefore, we did not decide to favor performance over security and server's design.

Thus, not taking the performance as a limiting factor, we have a general-purpose active server that anyone can deploy, write an application and investigate its behavior without studying server's source code.

The research project is called Smart Active Node, SAN in short [7].

IV. PROPOSED SOLUTION

Our efforts on building an active network started with an idea of a general-purpose server and IP-tunneling.

A. Generality, Usability and Security

The server does not make any assumption about applications, which will run in the network. However, programmer of an active application should aim for low resources consumption. Otherwise, security monitor may consider increased demands for resources as a possible attempt of a denial of service attack, or a malfunctioning application. The resource is anything that can be allocated to the application, or a capsule – i.e. memory, processor time, network bandwidth, etc.

Developing an active application should be as comfortable as developing a traditional application. Usage of a common IDE to develop active application is desired.

Any active code runs in a sand-boxed environment to meet security measures. No instance of any program code can affect another instance by mistake. For an inter-process communication, it is necessary to use server-offered API.

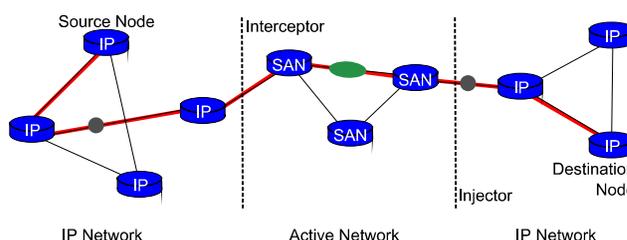


Figure 1. IP Tunneling Network Scenario.

In the present implementation, programmer supplies Java byte-code that is executed in an execution environment, the sand box, and controlled by the security monitor.

B. IP Tunneling

The goal is to let the Smart Active Node to provide a seamless IP tunneling through the active network. Fig. 1 depicts an illustrative network scenario. Consider two IP networks interconnected with an active network, where a source node sends IP packets to a destination node. The active nodes, which are connected to the IP networks, act as hybrid devices with both, IP stack and active networking functionality. As the IP packet gets to the hybrid node, it is intercepted at the third ISO/OSI layer. A component named Interceptor is responsible for this.

Then, the packet is encapsulated into a capsule and routed through the active network to the hybrid border node that is connected to the destination IP network. It is the capsule's program code, what makes the difference in performance. Note that as SAN runs on a standard operating system, both networks can overlay each other as well.

The destination-border hybrid node unpacks capsule's payload and injects the extracted packet into the IP network. The responsible component is called Injector.

Finally, IP network routes the packet to its IP destination.

The principle is the same for both directions so that Interceptor-Injector pair is present on each border node to satisfy two-way communication.

Fig. 2 depicts a view on assignment of responsibilities. Active network server and IP stack of underlying operating system cooperate. Oriented lines show the data flow. Starting with data coming through the IP stack, the interceptor component, called saninterceptor, receives the data as a

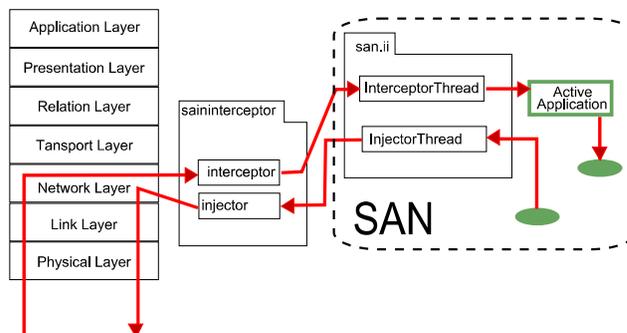


Figure 2. Linux IP Tunneling Components.

packet. Subsequently, it is passed to the ii component; ii stands for interceptor-injector. This component is responsible for encapsulating it into a capsule.

When receiving data as a capsule from an active network, the ii component delivers the data to the injector component. Then, the data are injected at the ISO/OSI network layer into the IP stack.

V. IMPLEMENTATION

Having the design, we continued with implementation. Initially, we assumed that Java and JVM will be fast enough, to get acceptable values of throughput and latency. Therefore, only OS-dependent parts of the tunneling were written in C++.

A. Generality, Usability and Security

As our solution is a general-purpose server, there is no special software needed to create custom active applications. Recently, some basic applications already work and development of others is in progress. The working ones include ping, trace route and IP tunneling. The work in progress comprises of dynamic routing, telnet, SSH and possibly a port of AVNMP [8] – a tool to predict network’s load.

SAN active application is written in the standard Java language, compiled and packaged into a Java archive with a manifest file. The applications can be developed in IDE such as Eclipse or Netbeans with no expenses.

The application’s, or capsule’s, code is interpreted as a Java byte-code in the present state. We developed our own byte-code interpreter. As it has been written from scratch to allow strict control over the execution process, it interprets everything, down to Java native methods. As the result, nearly every valid Java construct can be used to create an active application. Moreover, we have a full control over the code. Thus, passing a special file system identifier to obtain undesired access on particular operating system can be forbidden, as well as a simple constructs like calling System.exit(0) to shut down the server maliciously. Preceding works, such as ANTS, used directly the Java machine they run within, thus virtually providing no security.

B. Optimization

SAN started as a Java project for various reasons. As already mentioned, we need to address the performance. To improve it, a C++ clone of the server is being written. From this step, we expect a performance increase and the possibility to deploy the server on such nodes, where Java is not available, e.g. switches and routers.

In an active network server, the most likely bottleneck is byte-code interpreter and scheduler. To run the byte-code, it is necessary to prepare execution environment, i.e. the sand box, and to schedule it for execution. Preparing the execution environment is a time-significant part of total run-time, in a case of shortly running capsule codes such as ping. Thus, the overhead does not matter, if the application run-time is long and frequency of runs is low. However, it matters with applications such as the IP tunneling. The IP tunneling run-

time per capsule is very short and the frequency of runs can be very high. It depends on the data stream being transferred.

To speed up the code execution, we would like to have a mechanism that would optimize parts of code being executed frequently, and to cache them subsequently. The optimization would be a byte-code transformation into processor’s native instruction set.

Last optimization task is to examine the internal scheduler. It is currently implemented as a fair-share.

C. IP Tunneling on Linux

We have implemented the IP tunneling over active network on Linux first.

The idea behind the tunneling is following. If we want to pass IP packets transparently through the active network, we have to intercept IP packets either on physical layer, link layer or network layer to prevent the operating system from managing these packets. Otherwise, the operating system could possibly send ICMP error packets back, because it is not aware of being a part of active network.

We chose to use unmodified Linux kernel along with the Netfilter/Iptables [9] project to preserve simplicity, generality and ease of use. We used the Iptables’ NFQUEUE target along with the ipq library for queuing packets into user space. There is a benefit coming from the usage of this approach – we can easily decide, which packets from and to the IP networks are transferred through the active network.

After en-queuing a packet, entire datagram containing all headers is fetched into user space with libipq API calls. And, it is sent unmodified through a network socket to the SAN along with information about active code that handles its data.

Packet data and meta information exchange between saninterceptor and SAN ii component is accomplished through a standard socket, while using a special type of PDU to transfer the data. The PDU format and primitive data types are shown in Fig. 3; PDU stands for Protocol Data Unit. The first position of the PDU is the name of the active application being executed upon receiving the data. Then, an array of

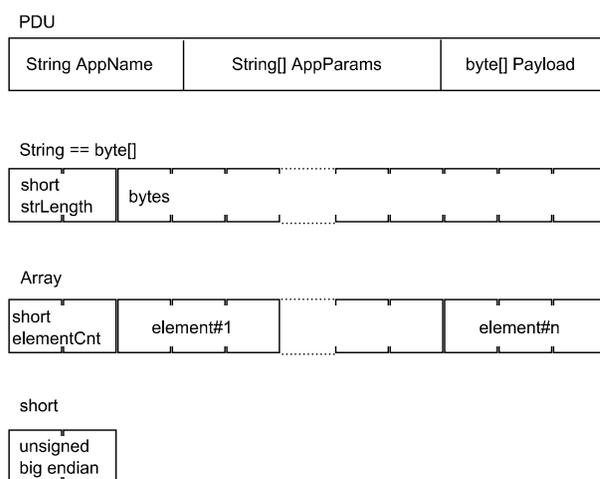


Figure 3. SAN Interceptor-Injector PDU.

application parameters apply. For example, they can represent routing, QoS or ToS information; ToS stands for Type of Service. Finally, entire datagram is attached. The PDU is flexible enough to handle a datagram up to 65kB. This is enough even for super jumbo frames.

Upon receiving PDU, the active node passes the datagram to the active application that is responsible for the IP tunneling. The application creates a capsule and injects it into the network. When the capsule arrives at the destination active node, datagram is unpacked and sent through the socket to the injecting application. Injector injects the datagram into the IP network. As the packet is not modified on its route, the process is fully transparent to IP applications.

We did tests with HTTP, SSH and FTP protocols. They worked flawlessly, like if no active network was presented.

D. IP Tunneling on Windows

We continued with implementation of the WDM driver model that applies to Windows 2003, Vista and 7. ReactOS uses the WDM model as well, but we made no tests on ReactOS yet. Fig. 4 depicts the implementation.

Legacy IP applications communicate via the TCP/IP NDIS protocol as usually. SAN filter intercepts the communication. Intercepted IP packet is accompanied with additional information and sent to SAN server via an inter-process communication. In SAN address space, an active application converts it into a capsule. Then, SAN server handles the capsule in a standard way.

When the capsule is to be converted back into the IP packet, NDIS driver does this as instructed by SAN. A legacy IP application gets the packet from the TCP/IP NDIS protocol as usually.

With the further development, we aim to support two kinds of applications – legacy IP applications and SAN-aware applications. SAN-aware applications would be free to use SAN capabilities directly. Thus, they would be able to exercise a finer control over the transmission.

VI. WORSE-PATH ROUTING

References [4, 5] give existing enhancements on multicast and tunneling of existing IP applications. We would like to go a step further by proposing such routing scheme that will rearrange network flows to benefit time-sensitive networking applications.

A. Policy-Based Routing

Let us classify network traffic into two categories. First one is time-sensitive traffic, for instance IPTV and VoIP. Second category is such traffic, where it is possible to tolerate some increase of delivery delay. For instance, SMTP and file-sharing services fall into this category.

We do not use terms real-time and non-real time traffic, because we discuss additional scenarios such as MPI in subsection D. While we assume a possible benefit for MPI, we do not assume a real-time application using MPI.

Multiple routes to target nodes may exist in a computer network, or an interconnection of computer networks – especially the Internet. Some routes are better in terms of

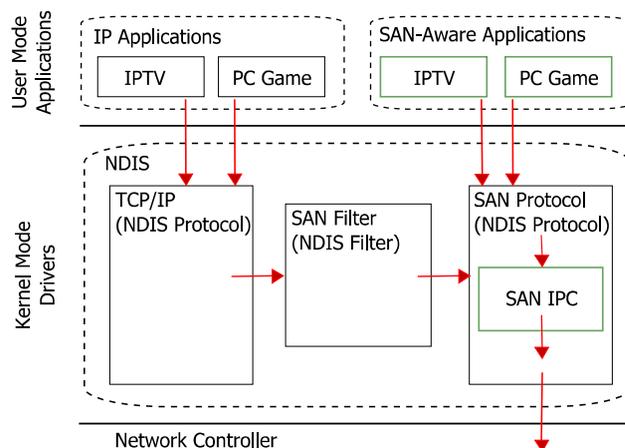


Figure 4. Windows IP Tunneling Components.

bandwidth, load, reliability, number of hops, etc. Routing algorithms try to find an optimum route. Regarding Internet Service Providers (ISPs), a price of link plays a role as well.

Let us consider an ISP with two different links to other ISP. One link is cheaper, but there is a lower bandwidth. To reduce costs, ISP would prefer such policy that would route most of the traffic through the cheaper link. Nevertheless, ISP should route the time-sensitive traffic through the faster link to maintain a quality of services to customers. In IP, this concept is implemented as policy-based routing.

However, the other ISP may not be interested in maintaining such quality of services to the customers of the traffic-originating ISP. By addressing this issue, the proposed concept differs from policy-based routing, as it is implemented in IP.

We give such IP tunneling scheme that routes the delay-tolerant traffic through slower links. As a result, it reduces the need to throttle the transmission speed of time-sensitive traffic on faster links. The proposed approach does not impose a need for agreement on common routing policies between two ISPs.

B. Principle

First task is to intercept such IP packets, which do not belong to the time-sensitive traffic. Then, we wrap these packets into capsules. Finally, associated program code routes the capsules through slower links – the worse-path.

Let us consider SMTP and IPTV for demonstration. Once SMTP server retrieves MX record for target domain, it opens a TCP connection to the destination server. Routers will direct the flow of connection's packets according to routing tables, as it would happen with the IPTV packets. SMTP and IPTV packets may share the same link. QoS can throttle transmission speed to favor time-sensitive traffic such as IPTV. However, QoS cannot route a particular TCP connection over a different link to gain yet more bandwidth for the IPTV. With IP and policy-based routing, we would need ISPs, which agreed on compatible routing policies. With a programmable network, we can apply the following concept.

First, we need an additional routing table at the router. To fill the table, it is necessary to modify routing metric so that it favors slower links. For example, OSPF uses inverse value of bandwidth. Then, we would take the bandwidth as the metric.

Second, we need a data unit that would be routed by the alternative routing table. In active networks, the router would execute the capsule's code. So, the capsule would look-up the alternative routing table and set its destination accordingly. If the router would not execute the code, e.g. for security reasons, the capsule would be forwarded according to the standard routing table. So, it would reach the destination as well, just sooner.

Capsules route themselves through slower links. Thus, they leave more bandwidth for the time-sensitive traffic on the faster links. Considering a possibility of different routing policies in transit networks, capsule's behavior increases the probability that time-sensitive traffic will use the faster links.

As capsule's program code does not change, the capsule acts the same way in all transit networks. Therefore, no two ISPs have to make a prior agreement on common routing policies.

C. IP-Programmable Hybrid Network

Let us consider a scenario, where a programmable node would aid a traditional IP network. As we do not tunnel the time-sensitive traffic, we can route it the standard way. On the other hand, the tunneled traffic is wrapped into capsules. We can distinguish such traffic easily, e.g. by port number, or a header bit. Therefore, it is possible to establish an efficient routing policy. Such policy would route capsules to the programmable node, while leaving rest of the traffic untouched.

Fig. 5 depicts a case scenario. Various clients from the source network #1 want to connect to particular hosts in the destination network #4. There is a policy-based routing enabled at the router that acts as their default gateway. It identifies particular protocols by port numbers. Selected traffic goes to the SAN server. Otherwise, the router forwards rest of the traffic to the IP-based border router. SAN server intercepts incoming IP packets and transforms them into capsules. According to programmable rules, it forwards them to the next SAN server. Note that the associated code can do much more than just policy-based routing. SAN servers in the transient networks act the same way. In the destination network #4, SAN server transforms capsules back into IP packets and forwards them with the standard IP routing mechanism.

The IP-programmable hybrid is not a fully programmable network. However, Fig. 5 depicts such scenario, where it is possible to route a defined amount of traffic to the programmable servers. As a result, we can test responsiveness and stability of the programmable servers to given load, while having backup routes.

Note that it is not necessary to deploy the hybrid network at the Internet scale. It can serve as well for networks of a single organization, or its department.

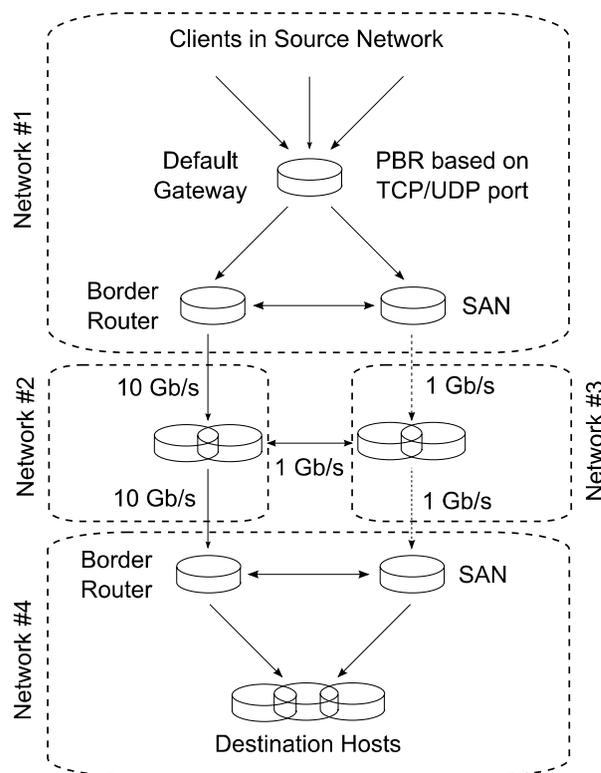


Figure 5. Hybrid Scenario of Worse-Path Routing.

D. Additional Case Scenarios

Let us consider a grid computing, for an illustrative example. A large grid may consist of several sub-grids, which are connected with slower links than the links inside the sub-grids. For distributed computing, there are tools such as GridMPI and PVM available. These libraries provide means for asynchronous and blocking communication.

For example, MPI_Send function is blocking. The caller does not continue its execution, until it receives a confirmation message. The communication overhead affects caller's performance, i.e. the completion time. For this reason, we can consider such communication as time-sensitive. Therefore, we should route it through faster links.

On the other hand, MPI_IbSend function is non-blocking. The caller continues its execution, while MPI delivers the message. For such programs, we could route such messages through slower links to reduce the waiting time of blocking operations such as MPI_Send.

Another possible scenario is secure, anonymous communication. The TOR project provides a network of nodes, which route communication in such manner that it is too hard to find its origin. TOR uses so-called onion routing and it supports applications, which use TCP. With tunneling, SAN could implement the same behavior for any packets, i.e. to build a secure network by default. With policy-based routing and client's IP, it would be possible to enable such service per individual user.

E. Comparison with Source Routing

IP offers loose and strict source routing. With such routing, packet's route is set in advance by the source node. While the strict routing sets entire path, routers may forward the packet through other routers as well with the loose routing. In both cases, sender must have such knowledge of topology so that the packet reaches the destination. There are two problems.

First, internal topology of other ISP is supposed to be opaque. At the best, there is no guarantee on knowledge of the topology, including bandwidths, utilization and other factors. In a worse case, ISP can choose to block the source routing.

Second, we do not discuss a use of source routing for an administrative task. We discuss use of the source routing for a regular traffic. In such case, the source node would have to maintain a complete routing table for entire Internet. This would impose overwhelming requirements on the node, thus rendering such solution as impossible.

In contrast to the source routing, SAN-based solution uses the well-established concept of routing tables along the route to the destination node.

VII. EXPERIMENTAL IMPLEMENTATION OF WORSE-PATH ROUTING

To implement the worse-path routing, we use the following technologies – AntNet for routing and Rendez-Vous to expose an additional programming interface to active code of capsules.

A. AntNet

First, we needed a routing algorithm. To benefit from the programmability, we implemented the AntNet algorithm [10]. This algorithm was originally designed for mobile agents. It is inspired by a behavior of ant colony. Using indirect information, a simulated pheromone trail, simulated ants find shortest path thanks to their cooperative behavior.

Making a reference comparison to OSPF, AntNet has better distribution of packet delays and negligible impact on the use of network bandwidth [10]. There are recent improvements to AntNet. They further improve throughput, stability and shorten time delay [11, 12]. Improved AntNet deals with topology changes better than OSPF does [11]. Reference [10] gives a comparison with other routing protocols as well.

For our experiments, we simulate Internet with a set of interconnected nodes. We simulate networks of individual ISPs with these nodes. In this fashion, we use AntNet as an exterior gateway routing protocol.

B. Rendez-Vous

Having the routing algorithm implemented, we needed to implement an alternative routing table in a general fashion. We wanted to avoid any ad-hoc solution. We implemented Ada rendez-vous synchronization mechanism into our Java byte-code interpreter.

In our implementation, an active application can register a rendez-vous server. Incoming capsule can lookup a particular server and call its entries. This way, we have

achieved a possibility of having so-called installable APIs. As a result, SAN exposes just the minimum set of functions through a pre-defined interface. Then, it is upon networking services and applications to expose desired application programming interfaces – APIs. It is possible to register and unregister particular API dynamically, without a need to restart SAN server.

The choice of rendez-vous has a security background. Synchronizing with a monitor, the calling thread executes monitor's code, thus using its internal data structures. A malicious code could possibly exploit such design. With rendez-vous, the calling thread is suspended, until the called thread, the server, finished the execution. Thus, the caller has no access to server's internal data structures by the very principle of rendez-vous. This is important to us as we consider a possibility of execution environment reuse to speed-up code execution.

There are stability benefits as well. Having a rendez-vous executing in a standalone thread, we avoid priority problems. As the rendez-vous thread keeps its priority, a critical section will not be blocked for too long by a thread with low priority. Moreover, the rendez-vous thread can exercise a better control over resources being allocated in a critical section, than calling threads could do. An unknown calling thread is more likely to be terminated by security monitor than a rendez-vous server thread is.

Java has no native program construct to implement rendez-vous as Ada has. There are two ways to implement the mechanism into present Java language standard.

One way is to develop a pre-processor that would allow Ada syntax in .java file. The preprocessor would generate a .java file that a Java compiler would accept. Therefore, any debug info would be valid for the generated .java file.

It is always necessary to develop a package, which would synchronize threads in the rendez-vous manner. Therefore, the second way is to use this package directly, without the pre-processor. This way, we have a .java file that programmer understands, compiler can process it and generates a debug info for it. We chose this way.

The following code shows a fragment of the rendez-vous server, which implements the worse-path routing table.

```
public void worsePathRoutingServer() {
    //1. Register Rendez-Vous server
    if (!applicationAPI.registerServer(this,
                                     WPRTTable_GUID)) {
        //Error registering the server, however
        //traffic is still routable. Capsules will
        //just use the default routing table.
        return;
    }

    //2. Register supported entry-calls
    applicationAPI.registerEntryCall(WPRTTable_GUID,
                                    GetGateway_NAME);

    //3. Handle the entry-calls
    try {
        applicationAPI.makeAccept(WPRTTable_GUID);
    } catch (InterruptedException ex) {
        handleException(ex);
    }
}
```

```
//4. Clean-up
applicationAPI.unregisterServer(WPRTTable_GUID);
} //end of worsePathRoutingServer()
```

The following code fragment shows as capsule routes itself, using the worse-path routing table:

```
public void routeCapsule() {

//1. Get capsule's destination; it is not stored
// in capsule's header as the header may
// change on the route. Therefore, we store
// the real destination in the payload.

NetIdentifier realDst = readCapsuleDestination(
    capsuleAPI.getPayload());

//2. Try to get worse-path routing record
// for the real destination.
try {
    RoutingRecord rr =
        capsuleAPI.callEntryCallByName(
            WPRTTable_GUID, GetGateway_NAME, realDst);

//3. If the previous call succeeded, extract
// the gateway and set it as capsule's
// destination.

if (rr != null)
    capsuleAPI.setDestination(rr.getGateway)

//4. In a case of failure, set the real
// destination. Then, the capsule will
// reach its destination through a normal
// route. However, this code will attempt
// to resume the worse-path routing
// at the very next node.

else capsuleAPI.setDestination(realDst);

} catch (InterruptedException ex) {
    capsuleAPI.setDestination(realDst);
}

} //end of routeCapsule()
```

While we have the key components done, the worse-path routing is not finished yet. As given in the following section, we need to switch to the SAN C++ port, first. Then, we can finish the implementation with such execution speed that is fast enough for a productive use. As SAN servers make a distributed environment, where each node is a parallel application, the execution speed is an important factor for debugging.

VIII. RESULTS

Initially, we expected better performance results than we achieved. Eventually, it turned out that speed and security with Java-in-Java in JVM will not run fast enough, despite source code optimization, runtime profiling and performance tuning done by JVM. For this reason, the paper ends with a focus on speed of code execution.

A. Background

The PANDA project [5] was built on a top of ANTS project [6]. ANTS project ran in Sun JVM. The JVM executed capsule code as well as ANTS' code. While this allowed a greater throughput than a Java-in-Java approach, the solution was not secure enough. Later on, secured solutions appeared. They included PLAN [13], RCANE [14], SANE [15] and SNAP [16]. They were able to verify integrity of server's code and configuration, and authenticity of capsule's code. Also, some of them limited programming constructs to avoid creating of a possibly dangerous program code.

We decided to strengthen the security measures by being able to monitor code execution on-the-fly. For this reason, we replaced the use of Sun JVM with our own Java-in-Java interpreter. This became the performance bottleneck of our server. Eventually, it became obvious that we need to abandon Sun JVM to run the server. We chose to port the server to C++, while leaving the Java development branch as a sand box. Presently, we use it to test new ideas and to develop active protocols in advance, prior to finishing the C++ port of the SAN server.

B. Linux IP Tunneling

We performed a couple of tests with the IP tunneling implementation. In all tests, there was a saturating traffic flow from one computer to another. We generated a continuous stream of IP packets to saturate link's bandwidth.

The first test was the performance test of the saninterceptor itself. It was aimed to prove correct memory management, effective CPU and bandwidth usage. Table 1 shows results for two directly connected PCs with 100 Mbps network cards and the same PCs connected through two instances of saninterceptor. Looking at the Table 1, we can say that use of saninterceptor nearly does not affect data transmission, even if it is implemented by simple means. N/A means that values were not observable or affected at all.

The second test was the performance test of a complete system, i.e. including SAN, active applications, etc. This test revealed some drawbacks in SAN's implementation. They are related to running many instances of short-run-time applications and capsules.

Although the tunneling results were not satisfying, they showed that the IP tunneling works and that it can be used for tunneling of applications like HTTP and SSH. Nevertheless, it became clear that we need to improve byte-code execution prior making any other substantial changes.

TABLE I. INITIAL SANINTERCEPTOR PERFORMANCE RESULTS ON LINUX TO LINUX

	CPU	Memory	Latency	Throughput
direct connection	N/A	N/A	<2ms	94 Mbps
Saninterceptor only	N/A	18kB	2ms	90 Mbps
SAN + saninterceptor	100%	N/A	>200ms	120 kbps

TABLE III. CODE EXECUTION TIMES ON WINDOWS

Environment	Average Time [sec]	First Time [sec]
SAN Java-in-Java	>> 1	>>1
SAN C++ JVM	1.37200	1.38000
1 st Sun JVM 1.6.0_17 64-bit	0.02350	0.02594
1.6.0_17 64-bit -Xcomp	0.02535	0.02605
2 nd Sun JVM 1.6.0_17 64-bit	0.01255	0.01295
1.6.0_17 64-bit -Xint	0.22561	0.22412
1.6.0_17 64-bit -g:none	0.01301	0.01363
VC2008 x64 Debug	0.07000	0.07000
VC2008 x64 Release	<0.00001 0.01000 occassionally	<0.00001

C. Execution Environment Benchmark

To compare performance of particular environments, and to estimate a minimum needed performance, we created a benchmark test. Since the C++ port is not finished yet, we cannot evaluate network-specific operations. Therefore, the benchmark computes a matrix determinant in such manner, that it uses memory, ALU and floating point instructions.

Table II gives results for Windows 7. To eliminate side effects of operating system, such as caching and program loading at random addresses, we ran the test for 10 times. Then, we ran the test for another 30 times and computed average execution time. The First Time column gives time just for the very first run. As the server has to execute a program code for a first time, as well as it may execute a particular code frequently, we give both – average and the first time. We collected the presented results on Intel64, family 6, model 23, stepping 10, frequency 2.40 GHz.

Table III gives results for Debian 4.3.2-1.1. The machine is part of Czech National Grid Project – MetaCentrum. This affects the software equipment. It runs on Intel Xeon, family 15, model 4, stepping 3, frequency 2.80 GHz. The benchmarking procedure was the same.

Using Sun JVM, we performed tests with non-standard switches. After the regular test, we run the test again, but with the non-standard -Xcomp switch. According to the documentation, everything should be optimized. Program loading took ~1 second on Windows, ~3 seconds on Linux. On Windows, the execution time did not change. On Linux, the execution time was longer. Then, we run the JVM again, but without the switch. On Linux, the execution time returned back to normal. On Windows, it was reduced by ~50%. Perhaps, there is some caching effect that causes such behavior.

Beside -Xcomp, we tested the -Xint switch as well. Accordingly to the documentation, the code should be interpreted only.

For comparison, we ran a C++ port of the benchmark with VisualC++ 2008 64-bit compiler, and with GCC 4.3.2 32-bit compiler.

SAN C++ JVM does not compile on Linux yet, so it is not included in Table III.

TABLE II. CODE EXECUTION TIMES ON LINUX

Environment	Average Time [sec]	First Time [sec]
SAN Java-in-Java	>> 1	>>1
SAN C++ JVM	N/A	N/A
1 st Sun JVM 1.5.0_10 32-bit	0.06137	0.06185
1.5.0_10 32-bit -Xcomp	0.09401	0.09388
2 nd Sun JVM 1.5.0_10 32-bit	0.06127	0.06138
1.5.0_10 32-bit -Xint	0.46654	0.46329
1.5.0_10 32-bit -g:none	0.06257	0.05964
GCC x86 -O0	0.02000	0.02000
GCC x86 -O3	<0.00001	<0.00001

D. Discussion

SAN's Java-in-Java interpreter is so slow that it has no point to measure the networking performance. With Java-in-Java, we test flow and logic correctness of program code.

Present JVM of SAN C++ performs much better. However, it is still significantly slower than Sun JVM. The performance results indicate that Sun JVM transforms byte-code into the native instruction set, based on some threshold given by code profiling. However, the optimization does not seem to be as good as it could be, on Windows and Linux x86 platforms.

If we would consider that the optimization does not make an intensive use of available processor registers to favor simpler-to-code utilization of stack, then it is a reasonable appeal to us to pursue the byte-code transformation, instead of developing the Java-in-Java byte-code interpreter further.

So, we already started to implement the byte-code transformation to the C++ port. SAN C++ will transform the byte-code, which is being executed frequently, into the native instruction set. Otherwise, it will execute the byte-code inside its JVM. We chose this rule to avoid program-code cache-trashing, and to reflect the very fact that the transformation takes some time as well. The threshold values of "executed frequently" and program-cache size are a subject to future research.

IX. CODE EXECUTION

To execute the active-networking code, we decided to support two code notations – the byte-code and processor-native code. The byte-code will be either interpreted, or transformed into the native instruction set. Any application can be executed in byte-code. For selected operating systems, the code can be supplied in a processor-native instruction set to support critical operations. Such code has to be signed digitally, and the server has to trust explicitly the particular code and the signer.

A well-written program in C has lower memory requirements than a byte-code equivalent. In addition, compilers such as GCC produce much more efficient native-instruction code than JVM does from byte-code. This is our motivation for allowing the possibility to supply the code in

native instruction set. We would like to have popular protocols to be handled as most efficiently as possible, with respect to their share in traffic composition.

A. Byte-Code Transformation

Instead of transforming byte-code instructions directly into processor-native instructions, we decided to generate C code. The generated code will use pointer arithmetic to manipulate operands of the byte-code instructions, which are stored in the stack. Then, we will rely on a C compiler for optimization.

For a complex byte-code instruction, such as a newarray or monitorenter, we will call a respective C-coded function. This way, we can handle methods such as System.arraycopy, as JVM already does [24].

B. Security

To enforce security measures at the program-code level, we need to forbid particular program constructs and to limit memory and processor-time utilization.

When we encounter an instruction such as anewarray, the byte-code interpreter asks security monitor. A C-transformed code will call a function that will do the same. The security monitor checks current memory allocation status and acts accordingly. If the amount of allocated memory is over a given threshold, the request is denied.

As byte-code interpreter runs, it counts number of executed instructions per interval. Scheduler will not plan the process, if it would overcome an imposed limit. Into the C code, we can insert such code blocks, which will check processor-time utilization and yield the processor eventually.

Alternatively, there are SetThreadContext and GetThreadContext functions on Windows. On Linux, there is the ptrace function. With these functions, we can suspend thread execution and get/set its context. Then, there will be no need to include those code blocks into the generated C-code.

A programmer may desire to call a certain, possibly dangerous method like System.exit. In SAN, calls are checked and possibly denied with black-lists. The configuration may look like this fragment:

```
<roles>
<role name="anonymous"
  resourceProfile="anonymousProfile">
  <permissions>
    <access name="java.lang.System.exit"
      type="method" allow="forbidden"/>
  </permissions>
</role>
</roles>

<resourceProfiles maxRunningCapsules="20"
  maxActiveCodes="50">
<profile name="anonymousProfile"
  priority="onIdle">
  <cpu type="percent" maxValue="5" />
  <memory type="percent" maxValue="5" />
  <bandwidth type="percent" maxValue="5" />
  <activeCodes maxValue="100" />
  <createdCapsules maxValue="5" />
</profile>
</resourceProfiles>
```

The concept of roles serves as an additional protection. For example, a time protocol cannot modify routing table, and routing protocol cannot set system time. Uncategorized protocols are most restricted with the anonymous role.

X. RELATED WORK

A number of papers were published on Active Networks in earlier years. Also, some recent works are relevant, although they do not address Active Networks exactly.

A. Google Chrome

Taking a closer look on the concept of the Google Chrome [17] operating system, we see a resemblance with the active networks concept. There is a simple, underlying operating system that provides hard application isolation – sandboxing [18]. API and the definition of web services define the Execution Environment. Also, it features a security manager that prevents running a malware.

B. Google Native Client

Although the Native Client [19] is not primarily designed for a use in active networks, there are ideas valuable to a high performance execution environment.

C. AntNet QoS

Another implementation of QoS that is based on a programmable approach is an adaptation of the AntNet routing algorithm for QoS [20]. The implementation was tested in a heterogeneous network as a part of a multimedia transcoding system. As a server receives a request, it generates ants, which search for a best transcoding path and service, based on desired QoS.

By having the AntNet algorithm implemented, we can continue to implement the QoS capability.

D. Security

Reference [21] gives an overview on CSANE active network concept, which aims for security and scalability. CSANE goes for cluster processing. It builds on ANTS, JanOS and Linux.

With rendez-vous, we take a preemptive counter-measure to deal with a possibility of attack, which is based on sharing a memory with another process. There was a similar attack, on the HyperThreading platform. One process obtained data of another process, particularly RSA key. We consider principle of this attack to possibly apply to monitor calls. Reference [22] provides attack details and gives suggestions to designers of operating systems.

E. Performance Testing

Reference [23] gives a recent, comparative study of JVM benchmarking – Sun JVM and Oracle JRockit. It concludes that JRockit runs usually 19% to 27% faster. Such numbers support the decision to perform byte-code transformation with SAN C++ internal means, instead of switching to another JVM.

F. Native Code

Reference [25] presents a load-redistribution method for distributed applications. As a proof of concept, it uses a special active-network server with no security, but utilizing active programs coded in processor-native instruction set. Execution overhead of this approach is insignificant.

XI. CONCLUSION

Adding program logic to passive IP packets may lead to a significant increase of network's efficiency [4, 5]. A network flow can adapt to current conditions automatically, as its units of transmission traverse the network. This is paid with increased overhead, as there is an executing code.

SAN is a universal active node server that is capable of IP tunneling to enhance performance of IP applications, as well as supporting newly created, active networking applications. Both can be accomplished in standard operating systems, so that no "overnight" revolution is needed to start benefiting from the active-networking concepts.

The related work shows that the concept of active networks is usable for the future – although, not in a way it was supposed to happen originally. With respect to the advances on the original work [1], we can try to evaluate history of active networks' development. Looking back at the history of active networks from the point of view of SAN development, it seems that the magnitude of initial support was driven more by expectations and possibilities, than it was corrected by development costs and requirements.

Active networks appeared with a proof-of-concept that was implemented with Java. While specialized languages appeared for active programs, a vast majority of well-accepted papers on active networking used Java. This gave the impression that Java is a good choice for development of active-networking server. And, we do not agree.

Speed and security were the major disadvantages, which prevented adoption of active networks. Our results suggest that the use of JVM is the cause. JVM cannot compete with an optimized code from a C compiler like GCC. Comparing Java and JVM with a well-written C program, Java loose, when it comes to memory requirements and code execution speed of both, active program and associated security checks. JVM overhead seems to be too great for software like active-networking server.

We still consider Java as a good choice for using the byte-code as active program notation, across different operating systems and processors. Also, Java benefits from a number of programmers and some language characteristics. For example, we consider it to be easier to implement security measures with references rather than pointers. Next, the garbage collector reduces the risk of memory leaks and segmentation faults.

On the other hand, references and garbage collector lead to increased memory demands and processor time spent in finalizing and freeing unreferenced objects. In addition, memory fragmentation boosts incurred speed penalty. Therefore, the server must be written in a C-like language, as well as frequently executed active program code. GCC-like

optimized code in processor-native instruction set is essential.

Java was an adequate choice for creating the proof of the concept for particular aspects of active networking. However, the situation has changed. To prove the concept of active networks as feasible for a productive use, we have to come close to performance of today IP stack implementations.

We succeeded with implementation of key components, while addressing shortcomings of preceding active-network implementations. Now, we need to finish the C++ port and to subsequently improve the code-execution speed with byte-code transformation.

ACKNOWLEDGMENT

The following members of SAN team implemented AntNet, Rendez-Vous, security and Windows IP Tunneling features: Zdenek Vacek, Miroslav Hendrych, Vladimir Aubrecht, Vaclav Papez and Petr Jaros {zdvacek|picard|aubrechv|vpapez|jarosp@students.zcu.cz}.

REFERENCES

- [1] J. Sykora and T. Koutny, "Enhancing Performance of Networking Applications by IP Tunneling through Active Networks", Proceedings of the Ninth International Conference on Networks, Les Menuires, France, 2010
- [2] K. Calvert, "Reflections on Network Architecture: an Active Networking Perspective", In ACM SIGCOMM Computer Communication Review, Volume 36, 2006, pp. 27-30, doi: 10.1145/1129582.1129590
- [3] D. L. Tenenhouse and D. J. Wetherall, "Towards an Active Network Architecture", Proceedings of DARPA Active Networks Conference and Exposition (DANCE.02), San Francisco, California, USA, 2002
- [4] M. Maimour and C. D. Pham, "AMCA: An Active-based Multicast Congestion Avoidance Algorithm," Proceedings of Eighth IEEE Symposium on Computers and Communications, Antalya, Turkey, 2003
- [5] V. Ferreria, A. Rudenko, K. Eustice, R. Guy, V. Ramakrishna and P. Reiher, "PANDA: Middleware to Provide the Benefits of Active Networks to Legacy Applications, Proceedings of DARPA Active Networks Conference and Exposition, San Francisco, California, USA, 2002
- [6] D. J. Wetherall, J. Guttag and D. Tennenhouse ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, IEEE Open Architectures and Network Programming 1998, San Francisco, California, USA, 1998
- [7] T. Koutny et al., "Smart Active Node", <http://www.san.zcu.cz/> Last Accessed on January 12, 2011
- [8] S. F. Bush and A. B. Kulkarni, "Active Networks and Active Network Management – A Proactive Management Framework", Kluwer Academic/Plenum Publishers, 2001
- [9] R. Rusty and W. Harald, "Linux Netfilter Hacking HOWTO", 2002, <http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.html> Last Accessed on January 12, 2011
- [10] G. di Caro and M. Dorigo, "An Adaptive Multi-Agent Routing Algorithm Inspired by Ants Behavior", Proceedings of PART98 - Fifth Annual Australasian Conference on Parallel and Real-Time Systems, Adelaide, Australia, 1998
- [11] S. Chandra, U. Shrivastava, R. Vaish, S. Dixit, M. Rana, "Improved-AntNet: ACO Routing Algorithm in Practice", Proceedings of UKSim 2009: 11th International Conference on Computer Modelling and Simulation, Cambridge, England, 2009

- [12] L. Zhang and L. Xiaoping, "The Research and Improvement of AntNet Algorithm", Proceedings of 2nd International Asia Conference on Informatics in Control, Automation and Robotics, Wuhan, China, 2010
- [13] M. Hicks, J.T. Moore, D.S. Alexander, C.A. Gunter and S.M. Nettles, "PLANet: an active internetwork", Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 1999
- [14] P. Menage, "RCANE: A Resource Controlled Framework for Active Network Services", Proceedings of the First International Working Conference on Active Networks, Berlin, Germany, 1999
- [15] D.S. Alexander, P.B. Menage, A.D. Keromytis, W.A. Arbaugh, K.G. Anagnostakis and J.M. Smith, "The Price of Safety in an Active Network", Journal of Communications and Networks, Special Issue on Programmable Switches and Routers, Volume 3, Number 1, March 2001
- [16] W. Eaves, L. Cheng, A. Galis, T. Becker, T. Suzuki, S. Denazis, C. Kitahara, "SNAP Based Resource Control for Active Networks", Proceedings of IEEE Global Telecommunications Conference, Taipei, Taiwan, 2002
- [17] J. Gray, "Google Chrome: The Making of a Cross-Platform Browser", In Linux Journal, Volume 2009, 2009
- [18] Ch. Reis, A. Barth and Ch. Pizano, "Browser Security: Lessons from Google Chrome", In Communications of the ACM, Volume 52, 2009, pp. 45-49, doi: 10.1145/1536616.1536634
- [19] B. Yee, D. Sehr, G. Dardyk, J. B. Chen, R. Muth, T. Ormandy, S. Okasaka, N. Narula, and N. Fullagar, "Native Client: A Sandbox for Portable, Untrusted x86 Native Code", Proceedings of 2009 IEEE Symposium on Security and Privacy, Oakland, California, USA, 2009
- [20] M. S. Hossain, and A. El Saddik, "QoS Requirement in the Multimedia Transcoding Service Selection Process", IEEE Transactions on Instrumentation And Measurement, Volume 59, Number 6, June 2010
- [21] C. Xiao-lin, Z. Jing-yang, D. Han, L. Sang-lu and C. Gui-hai, "A Cluster-Based Secure Active Network Environment", In Wuhan University Journal of Natural Sciences, Volume 10, Number 1, 2005, pp. 142 – 146, doi: 10.1007/BF02828636
- [22] C. Percival, "Cache Missing for Fun and Profit", Proceedings of BSDCan 2005, Ottawa, Canada, 2005
- [23] H. Oi, "A Comparative Study of JVM Implementations with SPECjvm2008", Proceedings of 2010 Second International Conference on Computer Engineering and Applications (ICCEA), Bali Island, Indonesia, 2010
- [24] Sun Microsystems, Inc., "Java SE 6 Performance White Paper", http://java.sun.com/performance/reference/whitepapers/6_performance.html#2 Last Accessed on January 12, 2011
- [25] T. Koutny and J. Safarik, "Load Redistribution in Heterogeneous Systems", Proceedings of the Third International Conference on Autonomic and Autonomous Systems, Athens, Greece, 2007

An Automated Framework for Mining Reviews from Blogosphere

Arzu Baloglu

Marmara University, Engineering Faculty
Computer Engineering Department
Istanbul, Turkey
E-mail: arzu.baloglu@marmara.edu.tr

Mehmet S. Aktas

Tubitak, The Center Of Research For Advanced
Technologies Of Informatics and Information Security,
Information Technologies Institute, Kocaeli, Turkey
E-mail: mehmet.aktas@bte.tubitak.gov.tr

Abstract— As usage of the Blogosphere increases, more and more Internet users have begun to share their experiences and opinions about products or services on the World Wide Web. Web logs (also known as blogs) have thus become an important source of information. In turn, great interest in blog mining has arisen, specifically due to its potential applications, such as collecting opinions regarding products, or reviewing search engine applications for their ability to collect and analyze data. In this study, we introduce an architecture, implementation, and evaluation of a Web log mining application, called the BlogMiner, which extracts and classifies opinions and emotions (or sentiment) from the contents of weblogs.

Keywords - blog mining, opinion mining, blog crawler, web blog mining

I. INTRODUCTION

The world's biggest library, the World Wide Web, is increasingly populated with data contributed by every Internet user around the world. People share their ideas, interests, emotions, experiences, and knowledge with others, in the form of opinions and reviews, via the Internet every day. Thus, mining opinions on the Web is a rich and important area for research [2].

Sociologists have used many different ways to recognize natural interests, aims, and preferences. In order to collect ideas from people's sharing over the Web, the most efficient way is to mine their Internet diaries, their blogs, which are their own direct, personal accounts of their ideas and opinions. This study introduces a system that is designed to mine ideas to understand the views of a web community.

In the last few years, blogs have emerged as widely known personal Web pages. Blogs began as online diaries. They are designed for regular updating. Each blog consists of a sequence of blog entries. A blog entry consists of a title, a textual content, and the time it was posted. Some blog entries may have comments by the blog readers. Some blogs are dedicated to a particular area of interest such as entertainment or business. Easy-to-use blogging tools have led to an explosion in the number of blogs. Especially with increasing usage of internet, blogging and number of blog pages are growing rapidly. Blog pages have become the most popular means to express one's opinions. By the end of 2008, there were 133 million blogs on the global Internet, as indexed by Technorati [3].

Mining opinions from Web pages involves several challenges. For example, these opinions, or review data, have to be crawled from Web sites and then separated from non-review data [9].

As an experiment, system extracts movie review data from blogs. As a result, we introduce an architecture and we describe the implementation of the system in detail. We also explain a classification of review data.

The organization of this paper is as follows. Section 2 discusses the literature. Sections 3-4 outline the proposed approach and the system architecture. Section 5 addresses the evaluation study. Section 6 concludes the paper with a summary and analysis of results.

II. LITERATURE SURVEY

In recent years, there has been a huge burst of research activity in the areas of sentiment analysis and opinion mining. Earlier studies focused mostly on interpretation of narrative points of view in text [6-11]. The widespread awareness of the research problems in sentiment analysis and opinion mining has increased with the rise of machine learning methods in natural language processing and information retrieval; of the availability of datasets for machine learning algorithms to be trained on (due to the blossoming of the World Wide Web); and, specifically, of the development of review-aggregation Web sites.

Zhongchao Fei et al. [4] describe a sentiment classification application that uses phrase patterns to classify opinions. In this study, at the document classification phase, the authors add tags to certain words in the text, and then match the tags within a sentence with predefined phrase patterns to find the sentiment orientation of the sentence under consideration. Next, they take into account the sentiment orientation of each sentence and classify the text according to the most repeated sentiment.

Jeonghee Yi et al. [5] describe a sentiment miner that extracts sentiment (or opinions) that people express about a subject, such as a company, brand, or product name. In this study, the authors design the sentiment miner with the following challenge in mind: Not only does it try to capture the overall opinion about a topic, but it is also the sentiment regarding individual aspects of the topic, thus capturing essential information of interest. The reason for this is that document-level sentiment classification fails to detect sentiment about individual aspects of the topic. Thus in the author's study, the sentiment miner analyzes grammatical sentence structures and phrases based on natural language processing (NLP) techniques, and detects, for each occurrence of a known topic spot, the sentiment about a

specific topic. With these characteristics, the proposed NLP-based sentiment system [5] achieved high quality results (~90% of accuracy) on various datasets, including online review articles and the general Web pages and news articles. The feature extraction algorithm, proposed by Jeonghee Yi et al. [5], successfully identified topic related feature terms from online review articles, enabling sentiment analysis at finer granularity.

Jian Liu et al. [6] describe an application that completes sentiment classification with review extraction. This approach extracts the review expressions on specific subjects and attaches a sentiment tag and weight to each expression. Then, it calculates the sentiment indicator of each tag by accumulating the weights of all the expressions corresponding to a tag. Next, it uses a classifier to predict the sentiment label of the text. In this study, the authors used online documents to test the performance of the proposed application. The experimental documents cover two domains: politics and religion. The experiments within those domains achieve accuracy between 85% and 95%.

Yun-Qing Xia et al. [7] describe a method of opinion mining to help e-learning systems note the users' opinions of the course-wares and e-learning teachers, and thus help improve the services. In this study, the authors develop an opinion mining system for e-learning reviews. The goal of this system is to extract and summarize the opinions and reviews, and determine whether these reviews and opinions are positive or negative. This study divides the whole task into four subtasks: expression identification, opinion determination, content-value pair identification, and sentiment analysis. The authors achieved the following precisions for these subtasks, respectively: 94%, 84.2%, 80.9% and 92.6%.

Qingliang Miao et al. [8] describe a sentiment mining and retrieval system called Amazing. The authors introduce a ranking mechanism, which is different from a general web search engine, since it utilizes the quality of each review rather than the link structures for generating review authorities. In this system, the most important aspect is that the authors incorporate the temporal dimension information into the ranking mechanism, and make use of temporal opinion quality and relevance in ranking review sentences. This study monitors the changing trends of customer reviews in time and visualizes the changing trends of positive and negative opinion respectively. It then generates a visual comparison between positive and negative evaluations of a particular feature in which potential customers are interested. The authors conducted experiments on the sentiment mining and retrieval system using the customer reviews of four kinds of electronic products, including digital cameras, cell phones, laptops, and MP3 players. The evaluation results indicate that the proposed approach achieves a precision of approximately 85%.

Li Zhuang et al. [10] describe a multi-knowledge-based approach that utilizes WordNet for statistical analysis and movie knowledge. WordNet is a large lexical database of English, developed under the direction of George A. Miller [11]. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct

concept. The proposed approach, described in [10], breaks down the problem of review mining and summarization into the following subtasks: identifying feature words and opinion words in a sentence; determining the class of feature word and the polarity of the opinion word; identifying the relevant opinion word(s) and then obtaining some valid feature-opinion pairs; and producing a summary using the discovered information. The authors use WordNet to generate a keyword list for finding features and opinions. Grammatical rules between feature words and opinion words are then applied to identify the valid feature-opinion pairs. Finally, the authors re-organize the sentences according to the extracted feature-opinion pairs to generate the summary. The objective of this study is to automatically generate a feature class-based summary for arbitrary online movie reviews. Experimental results show that this method has an average precision of approximately 65%. In addition, with this approach, it is easy to generate a summary with movie-related names as the sub-headlines.

In this study, we extend our previous work described in [1] and propose a project that is most similar to that described by Zhuang et al [10]. Our approach differs from this approach in the way we calculate sentiment orientation of the movie reviews from the blogs. The previous work focused on a constant dataset, while the proposed approach crawls the dataset from the blogs. In turn, this is used to calculate movie scores. We discuss our approach in detail in the next section.

III. APPROACH

A. Overview

In this section, we briefly describe problem definition, the techniques used in this study and what we aim to achieve as a result. This study is categorized into three phases. The first phase is the crawling phase, in which data is gathered from Web logs. The second phase is the analyzing phase, in which the data is parsed, processed and analyzed to extract useful information. The third phase is the visualization phase, in which the information is visualized to better understand the results. More details of the system architecture are explained in the system architecture section (IV).

B. Problem Definition

Web logs are full of un-indexed and unprocessed text that reflects opinions. Many people make choices by taking the suggestions of others into account. For example, one likes to buy a product that is most recommended by people who use that product. Thus, there is a need to crawl and process opinions, so that it can be used in decision-making processes of potential Web review applications.

C. Solution

In this study, we propose a blog mining system that will extract movie comments from Web logs and that will show Web log users what other people think about a particular

movie. Figure 1 shows the overall process model of the proposed system. As illustrated in this Figure 1, the blog mining process consists of following three main steps: Web crawling, sentiment analysis, and visualization.

Web crawling: A Web crawler (also known as a Web spider, or Web robot) is a program or automated script that browses the World Wide Web in a methodical, automated manner. A Web crawler is a type of software agent that takes a list of URLs, called seeds, to visit as input. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to a list of URLs, called the crawl frontier, to visit. URLs from the frontier are recursively visited according to a set of policies. The process of Web crawling is also known as spidering. Many sites, and search engines in particular, use spidering as a means of providing up-to-date data. Web crawlers (or spiders) are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam) or gathering text content.

In this study, we utilized two open source projects, OpenWebSpider [16] and Arachnode [21], for crawling the Web logs and collecting data for sentiment analysis.

Sentiment analysis: Sentiment analysis has three main tasks: determining subjectivity, determining sentiment orientation, and determining the strength of the sentiment orientation.

identifies whether the keyword has an adverb, which changes the degree of subjectivity. For determining the sentiment orientation the algorithm calculates the cumulative sentiment score for the review. If a keyword under consideration is found in the database, then the algorithm calculates the score.

Visualization: We utilize the Zed Graph [15] for visualization to present our findings. The Zed Graph provides an ASP web-accessible control for creating 2D line, bar, and pie graphs of arbitrary datasets. It is maintained as an open-source development project. We presented the results on the project website over a shared database.

IV. SYSTEM ARCHITECTURE

The proposed system architecture consists of several components: Blog Crawler, Sentiment Analyzer, and Web Usage Interfaces.

A. Blog Crawler

One of the most important parts of the proposed system is the blog crawler. The crawler needs to analyze as much data as possible to provide accurate results. If the analysis has not been conducted with enough data, the results will only indicate the opinions of a restricted group of people. Although one needs to crawl as many blogs as possible to obtain good results, the blogosphere contains huge amounts of data. The storage capacity is limited, and limitations also exist related to the computation and memory capabilities necessary to crawl all of the blogosphere.

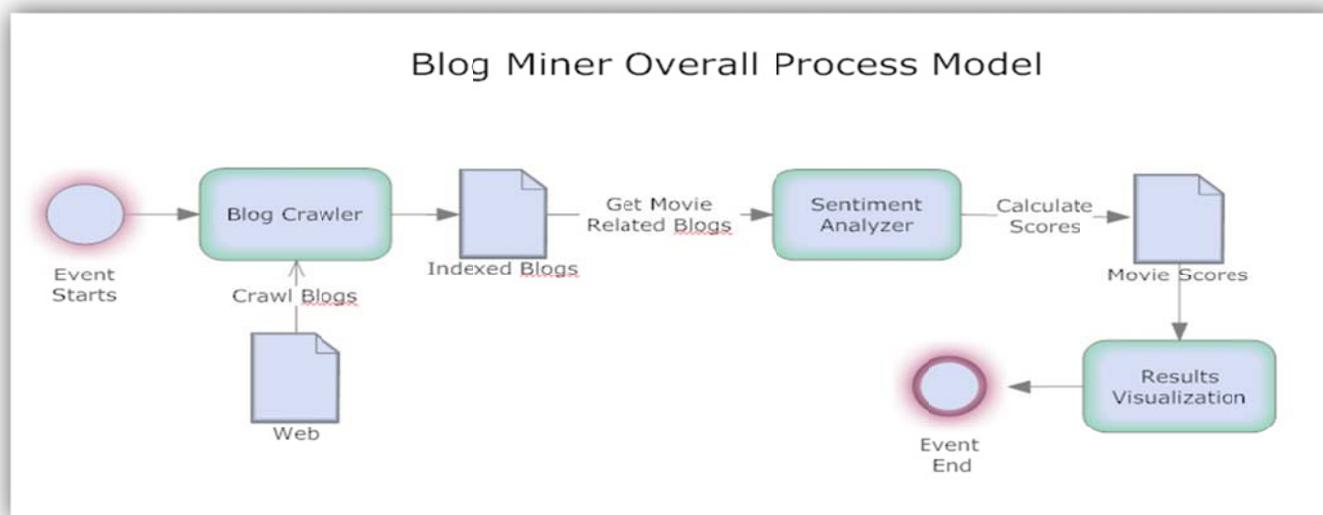


Figure 1 Blog Miner Overall Process Model

In this study, we use an unsupervised approach to sentiment analysis. For determining subjectivity, we use a keyword algorithm, which searches the pre-defined keywords in the text and then calculates their sentiment scores. For determining sentiment orientation, the algorithm

identifies whether the keyword has an adverb, which changes the degree of subjectivity. For determining the sentiment orientation, the algorithm calculates the cumulative sentiment score for the review. If a keyword under consideration is found in the database, then the algorithm calculates the score.

increased, the proposed application will produce better results.

keywords by mining the comments from blog pages. In order to calculate the sentiment scores, the analyzer first

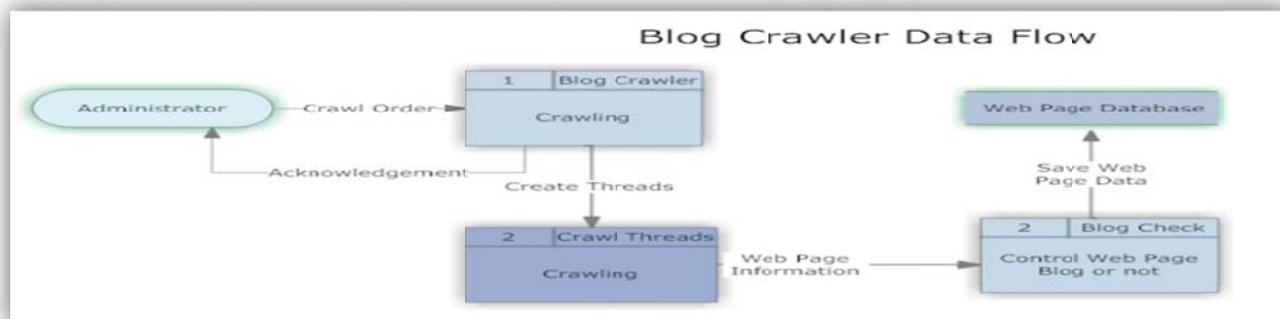


Figure 2 Blog Crawler Data Flow

We used Arachnode.Net to crawl the Web logs. Arachnode.net is an open source Web crawler for downloading, indexing, and storing Internet content including e-mail addresses, files, hyperlinks, images, and Web pages. Arachnode.net is written in C# and uses SQL Server 2005.

Arahnade.net uses the Lucene.Net library for indexing and searching. Arachnode.Net is selected because it is very customizable and well written. We start with seed lists like www.blogpulse.com and www.technorati.com, because these Web sites contain many links to Web logs. In turn, this improves the crawling performance. Figure 2 shows the data flow in the crawling process, while Figure 3 shows the main working process of the crawler.

As illustrated in Figure 3, the Web crawler starts by parsing a set of links that point to blog pages. The crawler then parses those pages for new links, and so on, recursively. After the new links are extracted, the system checks if they point to blog pages and inserts them into a queue of links to be processed by the crawler. The crawler resides on a single machine and sends HTTP requests for documents to other machines on the Internet, just as a web browser does when the user clicks on links. If the page is already fetched and resides in the cache, the crawler omits the link pointing to this page. All the crawler really does is to automate the process of following links for blog pages.

B. Sentiment Analyzer

The sentiment analyzer is a crucial component of the proposed system. If the analyzer finds a pre-defined keyword in a sentence of a given blog page for a specific movie, it looks for the sentiment words (such as an adjective or an adverb) that may be associated with that keyword. It calculates the sentiment scores for a movie for different

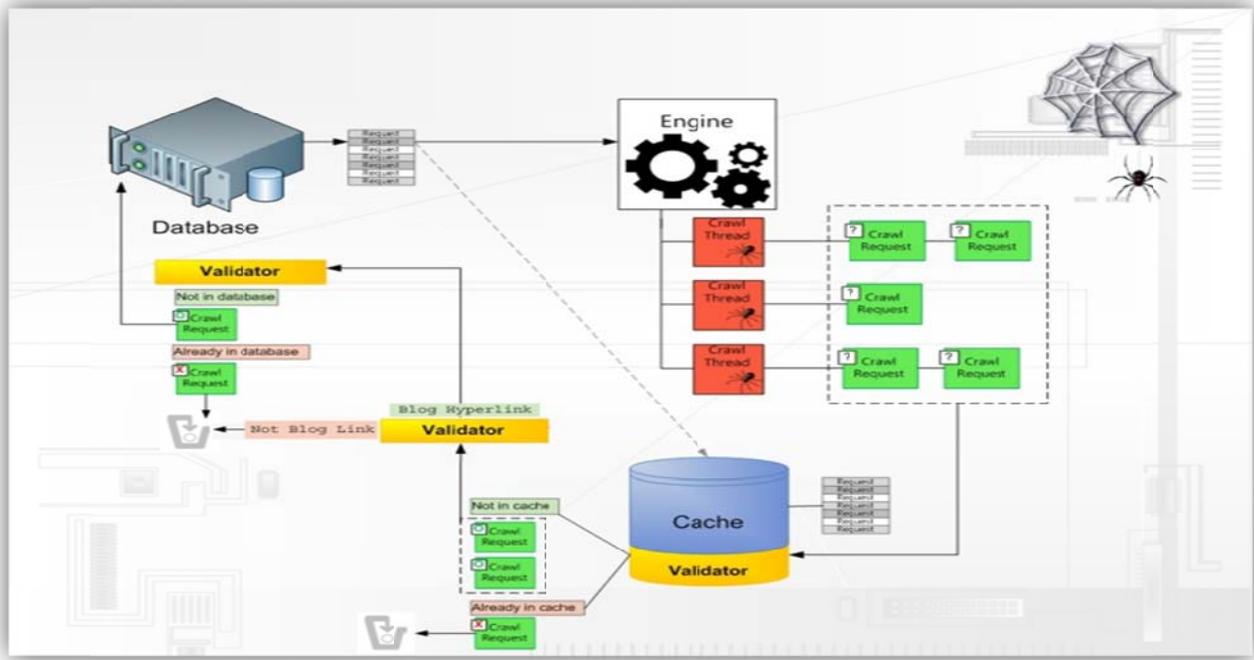
selects the blog pages that contain comments about a specific movie. Then, it parses each blog text and processes it in order to calculate sentiment scores for different keywords related to the movie under consideration.

The sentiment analyzer utilizes the aforementioned keyword algorithm in order to calculate sentiment scores. In this algorithm, the sentiment analyzer processes every sentence of a blog page for keywords such as “Screenplay,” “Director” and “Producer” that are related to the movie domain.

The analyzer utilizes the SentiWordNet [12] to obtain the sentiment scores. The SentiWordNet is a lexical resource, where each WordNet [11] synset s is associated with three numerical scores $Obj(s)$, $Pos(s)$ and $Neg(s)$, describing how objective, positive, and negative the terms contained in the synsets are. Figure 3 shows some adjectives and their scores according to the SentiWordNet.

If it finds a sentiment word, it obtains its score from SentiWord. It uses the obtained score as the keyword’s score and adds that to the total sentiment score of the blog page.

If the analyzer finds an adjective in a sentence of a given blog for a specific movie, it also looks for an adverb that modifies the degree of the adjective. Here, the adverbs are separated into two main categories, degree-adverbs and reversing-adverbs. If the analyzer finds a degree-adverb such as “less” or “more” in front of an adjective, then it multiplies the adjective’s score by the degree-adverb’s score and uses the result as the keyword’s score. If the analyzer finds a reversing-adverb such as “not” in front of an adjective, it simply reverses the score of that adjective and uses the result as keyword’s score.



id	wordid	type	posscore	negscore	objectivity	word
241669	800506	a	0.25	0.375	0.375	trenchant
241670	800669	a	0.125	0.25	0.625	impelling
241672	80096	a	0.375	0	0.625	accepting
241673	800997	a	0	0	1	rough-and-ready
241678	801906	a	0	0.875	0.125	ineffectual
241679	802074	a	0	0.625	0.375	effortful
241681	802754	a	0	0.625	0.375	difficult
241684	80310	a	0.125	0.625	0.25	rejective
241685	803227	a	0.125	0.625	0.25	labored
241686	803386	a	0	0.5	0.5	labor-intensive
241687	803568	a	0	0	1	leaden
241690	804036	a	0.25	0.5	0.25	effortless
241692	804388	a	0.5	0	0.5	unforced
241693	804585	a	0.5	0	0.5	efficacious
241694	80477	a	0	0.125	0.875	repudiative
241696	805086	a	0.25	0.125	0.625	inefficacious
241697	805276	a	0.375	0.375	0.25	efficient
241698	805616	a	0.25	0	0.75	businesslike
241699	805760	a	0	0	1	cost-efficient
241700	805869	a	0.125	0	0.875	economical

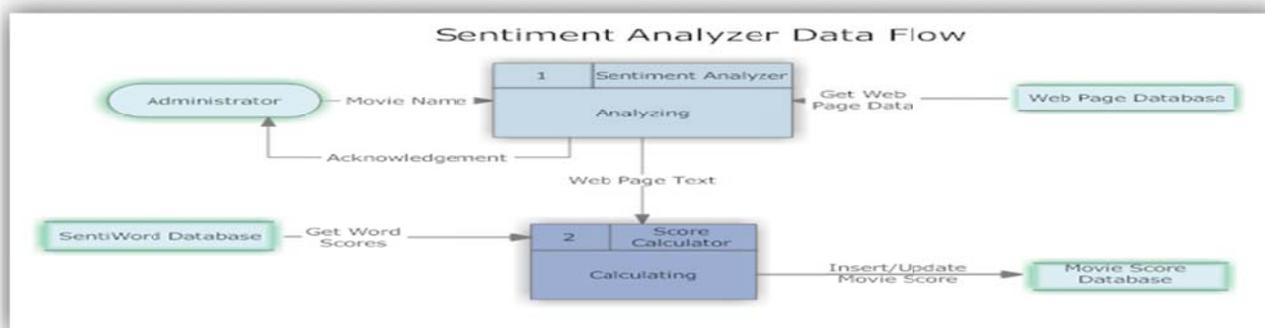


Figure 5 Sentiment Analyzer Data Flow

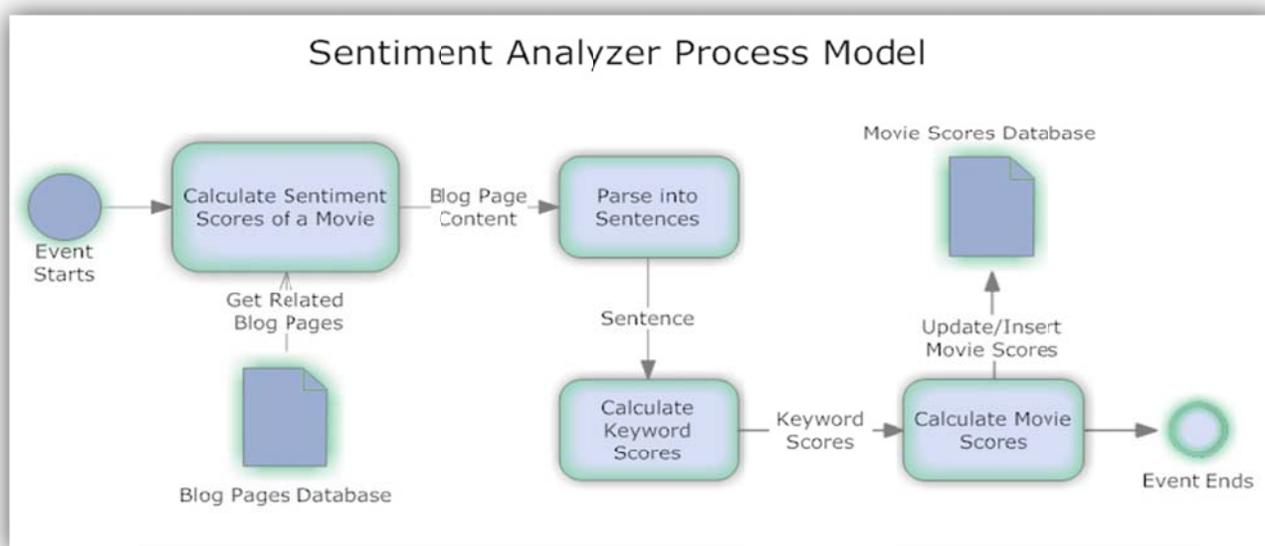


Figure 6 Sentiment Analyzer Process Model

In this way, the analyzer calculates the cumulative sentiment scores for all related blog pages for different pre-defined keywords and takes the average of these scores. In the end, the analyzer finds a sentiment score corresponding to each pre-defined keyword by mining the blogs for each particular movie. Figure 5 shows the data flow in the sentiment analyzer.

The reviews and comments in blogs may contain spelling errors, and these errors will decrease the accuracy of the application. To overcome this challenge, NetSpell [16] is used as a spelling library in our score calculation methodology.

The SentiWordNet database contains the stem of the words. In turn, this may affect the calculation of the sentiment scores and decrease accuracy of the application. Thus, in order to discover the sentiment score of a word, the analyzer must search its stem within the SentiWordNet. To overcome this problem, the analyzer utilizes the Porter Stemmer [14] to get the stem of a word. These text and word modifications improve the proposed application's accuracy. Figure 6 shows the process model of the sentiment analyzer.

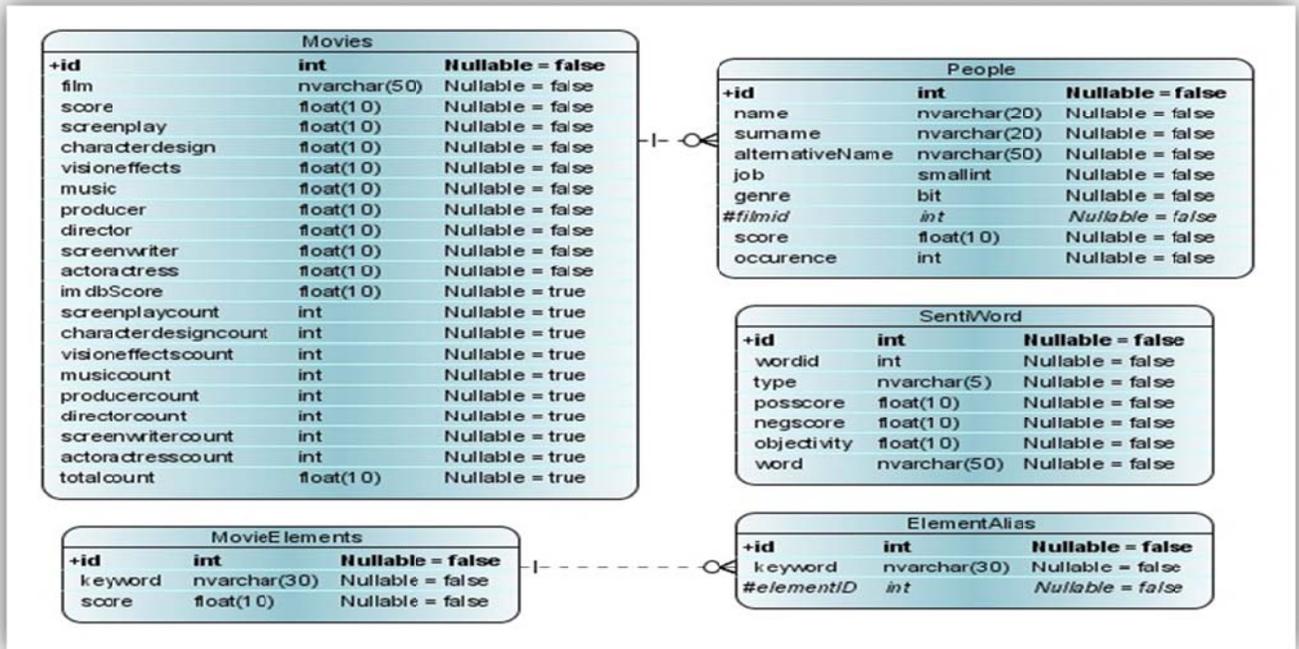


Figure 7 Blog Miner ER Diagram

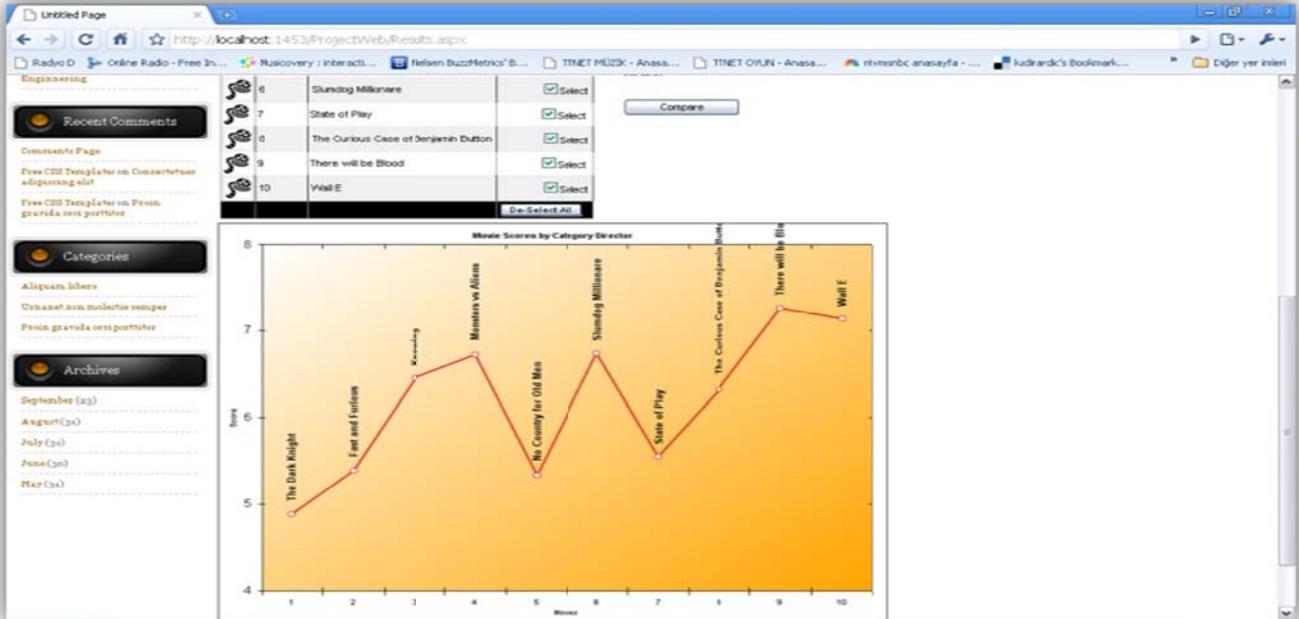


Figure 7 shows the Entity Relationship Diagram of the proposed application. Note that this diagram does not include the Arachnode.Net database, which is used to store blog pages. The database diagram of the Arachnode.Net is available at [12]. In Figure 7, the “Movies” table is used to store the score results of each movie under investigation. The “People” table is used to store all related information about the people involved in making movies, such as actors, actresses, and directors. The “SentiWord” table stores the sentiment dictionary, which was obtained from SentiWordNet [11]. The “Movie Elements” table stores the nine keyword categories from the movie domain, and the “Element Alias” table stores the keywords associated with these categories.

C. Web User Interfaces

We developed a Web user interface, as illustrated in Figure 6, to present the project evaluation and to give information about ongoing research. The web interface is used for two functions: The first category is the selection. There are two types of selection options. First is the selection of movies. Here, the system lets the user select a movie and then shows the sentiment score results corresponding to nine different keyword categories. Second is the selection of keyword categories. Here, the system lets the user specify only one category and shows the sentiment scores of different movies under the selected keyword category. In addition, the system also lets users select the movies they want to sketch and the category under which they want to do the analysis, and then shows the results in a graph.

The second category is the graphs. The system utilizes dynamic charts that are created each time users specify a selection as illustrated in Figure 7. Here, we utilize Zed Graph [15], which is an open-source library, written in C#, for creating 2D line and bar graphs of arbitrary datasets. This library provides a high degree of flexibility, i.e., almost every aspect of the graph can be user-modified.

Zed Graphs has two different libraries that can be used for creating Windows-based applications and Web-based applications. In this study, we use only some parts of the Zed Graph libraries to create a Web-based BlogMiner application.

V. EXPERIMENTS AND RESULTS

To evaluate the performance of the proposed application, we used reviews about several movies from The Internet Movie Database (IMDD) Web site [18] as the data set. For our analysis, we simply chose recent movies, since we want to analyze as many user comments as possible. Our assumption is that recent movies will attract more user comments, as they may have a larger audience.

Thus, we chose following 10 movies from the IMDB: “The Fast and Furious,” “Monsters vs. Aliens,” “State of Play,” “Knowing,” “The Dark Knight,” “Wall-E,” “Slumdog Millionaire,” “No Country for Old Men,” “There Will be Blood” and “The Curious Case of Benjamin Button.” For each movie, approximately 10 review pages are crawled by the Blog Crawler. In turn, this created approximately 1000 user reviews in total. These reviews are used for experiments to calculate the accuracy of the application. For the pre-defined keywords, we used the names of the three most important roles for each movie: actor/actress, director, and screenwriter. We include the names of these roles in the database in order to catch comments about actors, actresses, directors, and screenwriters. We refer the readers to Ardic and Enez [19] for extensive discussion on implementation and experiments.

We present our experimental study by showing the steps of the BlogMiner application for processing the raw data and calculating sentiment scores. Thus, the following sample user-review is chosen from IMDB to illustrate the steps.

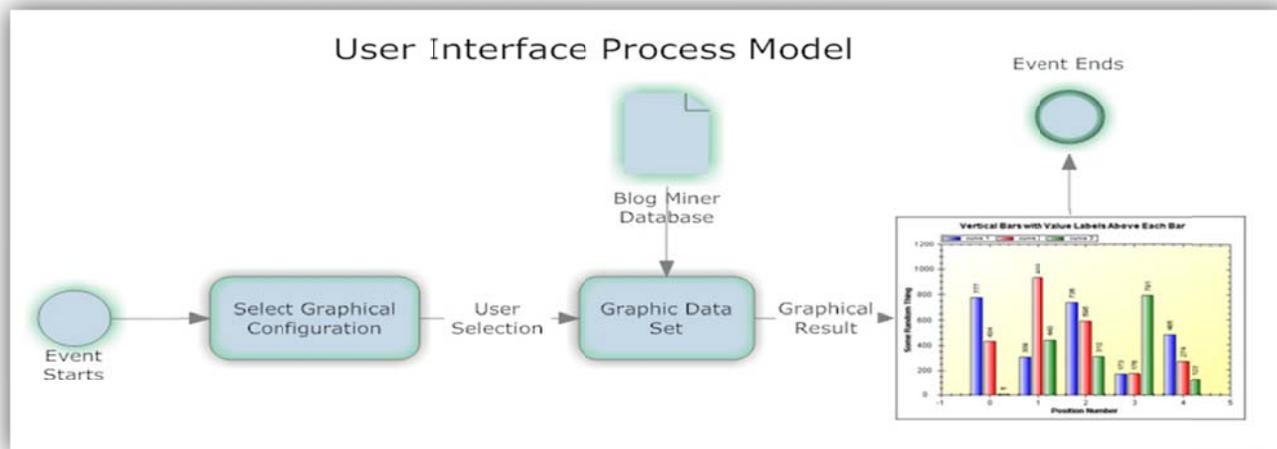


Figure 9 Web User Interface Process Model

Sample Review: "I thought it wouldn't be as good as it was, because thousands of people and reviews said it would suck! It was great, but what it missed was that it needed to be at-least an hour longer, because it missed a-little bit, but it still rocked! I loved it! I thought it was funny, and as did the person next to me, when John says: "I'll be back!""

First we split the text into sentences: In this step, the BlogMiner simply breaks down the text into sentences and makes the sentiment analysis at sentence level. Below, we illustrate this process as applied to the sample review after step 1.

~1~ I thought it wouldn't be as good as it was, because thousands of people and reviews said it would suck! ~1~
 ~2~ It was great, but what it missed was that it needed to be at-least an hour longer, because it missed a-little bit, but it still rocked! ~2~
 ~3~ I loved it! ~3~
 ~4~ I thought it was funny, and as did the person next to me, when John says: "I'll be back!" ~4~

Second, we tag the words in each sentence by their type. In this step, appropriate tags are added to the words to be able to understand their meanings more accurately. Figure 8 shows the tags that have been used and the meanings of these tags. The text below is the sample review after step 2.

I/PRP thought/VBD it/PRP would/MD not/RB be/VB as/RB good/JJ as/IN it/PRP was/VBD ./, because/IN thousands/NNS of/IN people/NNS and/CC reviews/NNS said/VBD it/PRP would/MD suck/VB !/.

It/PRP was/VBD great/JJ ./, but/CC what/WP it/PRP missed/VBD was/VBD that/IN it/PRP needed/VBD to/TO be/VB at-least/JJ an/DT hour/NN longer/RB ./, because/IN it/PRP missed/VBD a-little/JJ bit/NN ./, but/CC it/PRP still/RB rocked/VBD !/.

I/PRP loved/VBD it/PRP !/.

I/PRP thought/VBD it/PRP was/VBD funny/JJ, /, and/CC as/RB di/VBD the/DT person/NN next/JJ to/TO me/PRP, /, when/WRB John/NNP says/VBZ :/: "/^ I/PRP will/MD be/VB back/RB !/."/NN ./.

Third, we score the text using the keyword algorithm and calculate the scores. In this step, the system calculates the sentiment score for keywords and finds the accumulated scores for each sentence. Below, we illustrate the output of the sample review after step 3. The results of the experiments are illustrated in Figure 9 .

"I/PRP thought/VBD it/PRP would/MD not/RB<-1> be/VB as/RB good/JJ<0.844> as/IN it/PRP was/VBD ./, because/IN thousands/NNS of/IN people/NNS and/CC reviews/NNS said/VBD it/PRP would/MD suck/VB !/.

(sentence score = -0.844)

It/PRP was/VBD great/JJ<0.344> ./, but/CC what/WP it/PRP missed/VBD was/VBD that/IN it/PRP needed/VBD<-0.140625> to/TO be/VB at-least/JJ an/DT hour/NN longer/RB ./, because/IN it/PRP missed/VBD a-little/JJ bit/NN ./, but/CC it/PRP still/RB<-0.171> rocked/VBD !/.

(sentence score = 0.0104)

I/PRP loved/VBD<0.375> it/PRP !/.

(sentence score = 0.375)

I/PRP thought/VBD it/PRP was/VBD funny/JJ<-0.515> ./, and/CC as/RB did/VBD the/DT person/NN next/JJ to/TO me/PRP ./, when/WRB John/NNP says/VBZ :/: "/^ I/PRP will/MD be/VB back/RB !/."/NN !/.

(sentence score = -0.515)

As can be seen in this figure, the producer and screenwriter columns include rows with a score of 5.25. These scores are default values because no keywords were found for these movies.

The results of the experiment have been compared with each movie's IMDB score. On the IMDB page of each movie, the movie's general scores are listed. Thus, we can compare the IMDB score against the keyword algorithm's score.

For the producer and screenwriter categories, not enough comments were found to calculate a realistic score. As a result, most of the producer and screenwriter score columns are given the default value. When the results are compared against the IMDB scores, we observe a similar behavior. A movie with a low IMDB score also gets a low score in the proposed application. Similarly, a movie with high IMDB score gets a high score in the proposed application. We also observe two exceptions to this behavior. For example, the movies "Fast and Furious" and "State of Play" received high scores in our application; however, their IMDB scores are in a lower position than the proposed application calculated. We conclude that in the IMDB database, the comments and the score of the movie may not always be matched correctly.

VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Opinion mining is an important area of investigation. As Web 2.0 applications produce an enormous collection of meaningful information, mining such information has become an important task. In this study, we introduced an opinion mining application that is created for calculating movie scores from blog pages.

film	score	screenplay	characterdesign	visioneffe...	music	producer	director	screenwri...	actoractress	imdbScore	keywordscore
The Dark Knight	7.77	5.14	4.78	4.97	5.46	6.88	4.88	6.23	5.31	9	7.64
Fast and Furious	8.05	5.05	5.3	5.47	6.84	5.25	5.39	5.25	4.94	6.9	7.73
Knowing	7.6	5.26	7.5	4.98	2.81	1.11	6.45	5.25	5.84	6.8	7.41
Monsters vs Aliens	7.65	5	5.08	5.82	4.95	5.25	6.72	1.56	6.35	7	7.73
No Country for Old Men	7.69	5.37	5.47	6.12	5.63	5.25	5.34	5.25	5.55	8.3	7.75
Slumdog Millionaire	7.89	5.16	5.88	5.47	5.39	5.25	6.74	5.25	5.6	8.5	7.88
State of Play	8.28	4.98	5.25	6.68	5.49	5.25	5.55	5.25	5.35	7.9	7.83
The Curious Case of Be...	7.64	5.09	3.57	4.78	5.38	5.25	6.32	5.25	5.3	8.2	7.56
There will be Blooc	7.78	5.03	5.31	5.97	4.38	5.25	7.26	5.25	6.09	8.3	7.59
Wall E	8.14	5.43	5.92	5.53	5.31	5.25	7.14	4.32	6.38	8.6	7.82

Figure 10 Experiment Results

CC	Coordinating conjunction	RP	Particle
CD	Cardinal number	SYM	Symbol
DT	Determiner	TO	to
EX	Existential there	UH	Interjection
FW	Foreign word	VB	Verb, base form
IN	Preposition/subordinate conjunction	VBD	Verb, past tense
JJ	Adjective	VBG	Verb, gerund/present participle
JJR	Adjective, comparative	VBN	Verb, past participle
JJS	Adjective, superlative	VBP	Verb, non-3rd ps. sing. present
LS	List item marker	VBZ	Verb, 3rd ps. sing. present
MD	Modal	WDT	wh-determiner
NN	Noun, singular or mass	WP	wh-pronoun
NNP	Proper noun, singular	WP\$	Possessive wh-pronoun
NNPS	Proper noun, plural	WRB	wh-adverb
NNS	Noun, plural	'	Left open double quote
PDT	Predeterminer	,	Comma
POS	Possessive ending	'	Right close double quote
PRP	Personal pronoun	.	Sentence-final punctuation
PRP\$	Possessive pronoun	:	Colon, semi-colon
RB	Adverb	\$	Dollar sign
RBR	Adverb, comparative	#	Pound sign
RBS	Adverb, superlative	-LRB-	Left parenthesis *
		-RRB-	Right parenthesis *

Figure 11 Word Tags

Experimental results show that the proposed application produces accurate results close to IMDB result values. With this study, we introduced an unsupervised approach for sentiment analysis.

For future study, we want to further improve this application and investigate how clustering and self-organization methodologies can be used to improve the accuracy in the results. We will further improve the software so that the users are able to add their own keywords at runtime. We will also investigate the scalability of this approach by investigating the system performance under an increasing number of keywords.

Acknowledgement: We thank Kadir Ardic and Onur Enez for their contribution to the research presented in this paper. We also thank the Department of Computer Engineering in Marmara University for giving us permission to commence

this study and to do the necessary research work by utilizing departmental computer facilities.

REFERENCES

- [1] Baloglu, Arzu, Aktas, Mehmet, Mining Movie Reviews from Web Blogs: An approach to Automatic Review Mining, Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference, Barcelona, Spain, 9-15 May 2010
- [2] Bing Liu, Web Data Mining - Exploring Hyperlinks, Contents and Usage Data, Text Book, , Springer, December, 2006
- [3] Technorati Web Site is available at <http://technorati.com>, last accessed October 2009
- [4] Zhongchao Fei, et al., Sentiment Classification Using Phrase Patterns Proceedings of the Fourth International

- Conference on Computer and Information Technology (CIT'04), 2004.
- [5] Jeonghee Yi, et al., Sentiment Mining in WebFountain, Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), 2005
- [6] Jian Liu, et al., Super Parsing: Sentiment Classification with Review Extraction, Proceedings of the Fifth International Conference on Computer and Information Technology (CIT'05), 2005.
- [7] Yun-Qing Xia, et al., The Unified collocation Framework for Opinion Mining, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.
- [8] Qingliang Miao, et al., AMAZING: A sentiment mining and retrieval system, Expert Systems with Applications (2008) doi:10.1016/j.eswa.2008.09.035.
- [9] Qiang Ye, et al., Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Systems with Applications (2008) doi:10.1016/j.eswa.2008.07.035.
- [10] Li Zhuang, et al., Movie review mining and summarization, Proceedings of the 15th ACM international conference on Information and knowledge management, 2006.
- [11] WordNet Web site is available at <http://wordnet.princeton.edu>, Access Date: October 2009.
- [12] Andrea Esuli, et al., SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining, The fifth international conference on Language Resources and Evaluation, LREC 2006
- [13] Arachnode.Net Web site is available at <http://arachnode.net/media/g/databasediagrams/-default.aspx>, last accessed October, 2009
- [14] Porter Stemmer Web site is available at <http://tartarus.org/~martin/PorterStemmer>, Access data: October 2009
- [15] Zed Graphs Web site is available at http://zedgraph.org/wiki/index.php?title=Main_Page, Access date: October 2009
- [16] NetSpell Web site is available at <http://sourceforge.net/projects/netspell>, Access date: October 2009
- [17] OpenWebSpider Web site is available at <http://www.openwebspider.org>, Access date: October 2009
- [18] The Internet Movie Database (IMDB) Web site is available at <http://www.imdb.com>, Access date: October 2009
- [19] Kadir Ardic, Onur Enez, Blog Mining, Undergraduate graduation thesis is available at <http://www.scribd.com/doc/-16191423/Web-Blog-Miner-Licence-Thesis>, last accessed October 2009

Towards Effort Estimation for Web Service Compositions using Classification Matrix

Zheng Li
NICTA and UNSW
School of CSE
Sydney, Australia
Zheng.Li@nicta.com.au

Liam O'Brien
CSIRO and ANU
School of CS
Canberra, Australia
Liam.O'Brien@csiro.au

Abstract — Within the service-oriented computing domain, Web service composition is an effective realization to satisfy the rapidly changing requirements of business. Although the research into Web service composition has unfolded broadly, little work has been published towards composition effort estimation. Since examining all of the related work in this area becomes a mission next to impossible, the classification of composition approaches can be used to facilitate multiple research tasks. However, the current attempts to classify Web service composition are not suitable for the research into effort estimation. For example, the contexts and technologies of composition approaches are confused in the existing classifications. This paper firstly proposes an effort-oriented classification matrix for Web service composition, which distinguishes between the context and technology dimension. The context dimension is aimed at analyzing the environmental influence on the effort of Web service composition, while the technology dimension focuses on the technical influence on the effort. Therefore, different context types and technology categories can be treated as different effort factors. Based on the classification matrix, this paper also builds an effort-estimation-checklist table by applying a set of qualitative effort estimation hypotheses to those effort factors. The table can then be used to facilitate comparing the qualitatively estimated effort between different composition approaches.

Keywords - *service-oriented architecture (SOA); classification matrix; Web service composition; effort hypotheses; effort estimation*

I. INTRODUCTION

Web services have been widely accepted as the preferred standards-based way to implement Service-Oriented Architecture (SOA) in practice. Since “only when we reach the level of service composition can we realize all the benefits of SOA” [16], the research into composing Web services has grown significantly along with the increasing necessity of reusing existing resources. Over the past decade, numerous works for composing Web services have been developed and reported in the literature. However, little work can be found towards the cost and effort estimation for Web service compositions. Meanwhile, it is difficult to investigate different composition effort by exhausting all the published composition approaches. However, we can inductively classify the existing Web service composition works, and

thereby to facilitate the comprehension of related knowledge and the effort estimation work.

Existing classification work of Web service composition can be found in several survey papers [17, 19]. These classifications are either incomplete or ambiguous, which brings many issues when using them to categorize and analyze new composition approaches. Firstly, none of the existing classifications distinguishes between the composition technologies and the composition contexts. For example, Dustdar and Schreiner [17] list model-driven approaches as a separate composition category, while combining AI planning approaches with the automatic design process and ontology environment. Secondly, the terminology is vague in some composition classifications. For example, Rao and Su [19] use “static composition” to cover those approaches having manual workflow generation, even though the component Web service selection and binding are accomplished automatically. Finally, the lack of clear classification targets is the most significant weakness of existing classification work of Web service composition. Current classification work generally surveys composition types through subjective identification without objective constraints. The resulting classification is then hardly associated with other specific research topics such as software cost and effort estimation. For example, the declarative service composition class [17] focuses on its irregular composition architecture that is almost irrelevant to the composition effort and cost.

In this paper, we first present a novel classification matrix aimed at the influence on the effort of Web service composition. This matrix uses clarified terminology, and differentiates the classifications between the *Context* and *Technology* dimensions. The *Context* dimension includes major effort related contexts that are *Pattern*, *Semiotics*, *Mechanism*, *Design Time* and *Runtime*. When considering different composition *Patterns* for the same target, orchestration deals with a central mediator while choreography is a collaboration of all the participant Web services. Within the *Semiotics* context, semantic Web services have more descriptions than syntactic Web services, which can facilitate service discovery and matchmaking. *Mechanism* context comprises SOAP-based and RESTful composition. RESTful Web service compositions are relatively lightweight compared with SOAP-based

compositions. According to the manipulation procedure before generating a real composite Web service, there can be manual, semi-automatic, or automatic compositions at *Design Time*. During *Runtime*, the dynamic and static compositions are differentiated by the adaptability of Web service composition. On the other hand, the Technology dimension is divided into well defined *Workflow-based*, *Model-driven*, and *AI Planning* technology categories. In fact, one composition approach can be classified into one technology category and some context categories at the same time. For example, the approach in [5] uses model-driven technology and is under the contexts: Orchestration, Semantics, SOAP, Manual, and Static. Therefore, a matrix is suitable to represent this kind of cross-classification.

Considering the different influences on the composition effort, different context types and technology categories can be viewed as different effort factors of Web service compositions. After applying a set of effort estimation hypotheses to these factors, we can get a checklist that qualitatively defines their effort influences. By using several assistant symbols and rules, an effort score is further assigned to each factor to reflect its influence on composition effort. By associating the effort scores with the applied hypotheses, we can then build an effort-estimation-checklist table based on our previously proposed effort-oriented classification matrix of Web service composition [1]. Supposing the effort scores of different factors across two dimensions can be multiplied to reflect their combined influence on composition effort, the multiplied result are also specified in the corresponding cross area in this table. Eventually, the effort-estimation-checklist table facilitates comparing the qualitatively estimated effort of different composition approaches listed in the classification matrix.

The contributions of this research are manifold. Firstly, the complete classification matrix can help researchers explore the knowledge space in service composition domain, and help developers choose suitable techniques when composing Web services. Secondly, since different technology categories and context types can be regarded as different effort factors when composing Web services, a set of effort estimation hypotheses are proposed and a checklist is generated to qualitatively define these factors' influences on composition effort. Thirdly, an effort-estimation-checklist table is built, which can further help researcher and developers compare the qualitative effort between different composition approaches. Last but not least, new research opportunities could be revealed when comparing and analyzing those different composition approaches.

This paper is organized as follows. Section II justifies the necessity of the research into effort estimation for Web service composition. The two following sections try to identify effort factors of Web service composition by building up a classification matrix. Section III introduces the context-based classification through specifying every type of context. Section IV presents the technology-based classification, and explicitly defines different technology categories. In addition, a part of our work is demonstrated in Appendix I as an example of classification matrix of Web service composition. Section V introduces a set of effort

estimation hypotheses, and applies these hypotheses to different composition effort factors. The result then constitutes an effort-estimation-checklist table, as illustrated in Appendix II. The conclusion is drawn, and some potential research opportunities are identified in Section VI.

II. NECESSITY OF EFFORT ESTIMATION FOR WEB SERVICE COMPOSITION

As previously mentioned, service composition has increasingly become a significant type of SOA project. In SOA, composition of services is the concept with which we provide support for business processes in a flexible and feasible way. Through this way of business support, business processes in SOA are essentially a composition of service invocations in a certain order with rules that influence the execution and other constructs, such as parallel invocations, transformations of data, dependencies, and correlations. As organizations move to having more and more services, and business application software will increasingly rely on subscribing services [49], then the major problem in SOA implementation will be service composition and may be less on development of new services.

Consequently, we can concentrate on the service composition as a breakthrough in effort estimation for SOA implementations that is crucial for properly balancing the benefit and cost in SOA system investment or project bidding. In practice, contemporary SOA is intrinsically reliant on Web services, and meanwhile Web service concept and technology used to actualize service-orientation have influenced and contributed to a number of the common SOA characteristics [50]. Therefore, Web service can be viewed as the de facto implementation of service concept, and we can then focus on the effort estimation for Web service compositions.

To the best of our knowledge, unfortunately, there is little work published about estimating effort of composing Web services. Through literature review, we believe the challenges of effort estimation for Web service composition are mainly twofold:

- **The complexity of Web service composition.** Following general principles of SOA, composing Web services may comprise distributed processes because component Web services are loosely coupled and could scatter in different locations. Josuttis [46] has pointed out that distributed processing would be inevitably more complicated than non-distributed processing, and any form of loose coupling would increase complexity.
- **The diversity of Web service composition.** Existing works [1, 17, 19] have revealed that numerous solutions to Web service composition have been proposed during the past decade. Different techniques and contexts may result in different influence on the final effort of an instance of Web service composition.

Limited to these two challenges, it is nearly impossible to collect enough development data to estimate effort of various complex compositions quantitatively. For a particular Web

service composition project, nevertheless, qualitative effort comparison between different composition approaches can still facilitate developer's decision making. Therefore, this paper is to investigate such a method to realize the qualitative comparison between composition effort estimates.

III. CONTEX-BASED CLASSIFICATION OF WEB SERVICE COMPOSITION

The context discussed here refers to the environment and different stages involved in composing Web services. Through analyzing the lifecycle of Web service composition, we have identified several contexts: *Pattern*, *Semiotics*, *Mechanism*, *Design Time*, and *Runtime* that have the most influence on composition effort.

A. Pattern: Orchestration and Choreography

According to the methods of cooperation among component Web services, the Web service composition patterns can be distinguished between *orchestration* and *choreography*.

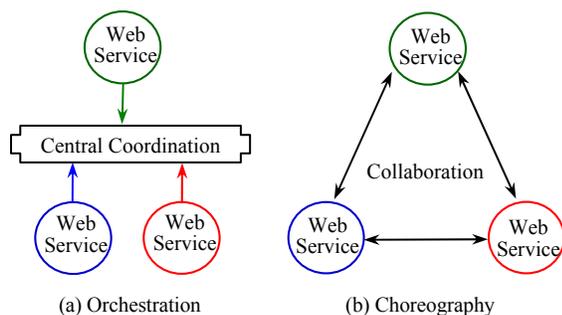


Figure 1. Web Service Orchestration and Choreography.

Orchestration, as shown in Figure 1(a), describes and executes a centralized process flow that normally acts as a coordinator to the involved Web services. The central coordinator explicitly specifies the business logic and controls the order of invocation of Web services. As a result, the coordination defines a long-term, cross-organization, transactional process. The involved Web services, on the other hand, need not be aware of their involvement in an orchestrated process. Orchestration represents coordination from the perspective of a single participant that can be another Web service.

Choreography, as shown in Figure 1(b), describes collaboration between web services that focuses on the peer-to-peer message exchange. The collaboration is decentralized where all participating Web services work equally and do not rely on a central controller. Each Web service involved in choreography understands its contribution to a business process: operation, timing of operation, and the interaction with other participants. Choreography represents collaboration from a global perspective.

In brief, orchestration and choreography describe two aspects of Web service composition for creating business processes [38]. Orchestration concentrates on the interactions of a single Web service with its environment, while choreography concentrates on the exchange of messages

among all the involved Web services. Consequently, an orchestration can be broken down into a series of primitive workflow logic activities, which invokes Web services following the determined execution sequence based on the central controller's enactment; whereas a choreography can be broken down into a series of message exchanges, which is not to control but to make autonomous participants cooperate based on their agreement.

In most cases, the pattern to which Web service composition belongs can be identified easily through the adopted standards or flow languages. For example, the current de facto standard for Web service orchestration is the Business Process Execution Language also known as BPEL. BPEL is an executable business process modeling language that can be used to describe the execution logic by defining the control flow and prescribing the rules for managing the non-observable data. The BPEL engine can then execute the description and orchestrate the pre-specified activities. Whereas one of the most widespread W3C recommended protocols for choreography is Web Services Choreography Description Language (WS-CDL). WS-CDL is designed to describe the common and collaborative observable behavior of multiple Web services that interact with each other to achieve their common goal. In other words, WS-CDL description offers the specification of collaborations between the participants involved in choreography.

Therefore, we can conveniently identify that the BPEL description related Web service compositions normally have orchestration context, e.g. [22], while WS-CDL description involved Web service compositions generally have choreography context, e.g. [23]. Nevertheless, the Web service composition pattern should not be judged merely through these keywords, because the technique can be adapted to satisfy different scenarios. For example, some people advocate the use of abstract BPEL as a choreography language. Consequently, the most reliable judgment should be still based on the understanding of the Web service composition process.

B. Semiotics: Syntactic and Semantic Compositions

The semiotic environment is becoming a more significant context for Web service composition as the Web evolves. Semiotics is the general science of signs, which studies both human language and formal languages. *Syntax* and *Semantics* are two of fundamental components of semiotics. Syntax relates to the formal or structural relations between signs and the production of new ones, while semantics deals with the relations between the sign combinations and their inherent meaning.

Currently, the World Wide Web can be mainly considered as syntactic Web that uses Hyper Text Markup Language (HTML) to compose documents and publish information. When it comes to Web services, the syntactic level XML standards, for example Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) have been used extensively to address corresponding e-business activities and research issues in industry and academia. By using human-oriented metadata,

SOAP is designed to provide descriptions of message transport mechanisms; WSDL is for describing the interfaces of Web services; while UDDI registers Web services by their physical attributes such as name, address and functional categorization. However, the syntactic Web was designed primarily for human interpretation and conveying information, a syntactic web page does not contain special tagging and the meaning of information is not readable by a computer program. The lack of machine-readable semantics then requires human intervention for Web service discovery and composition, and therefore hampers the usage of Web services in complex business environment.

To overcome the obstacles of interpretability and interoperability between traditional systems and applications, the semantic Web was proposed through incremental and information-added adjustments. These adjustments make the Web ontological. Ontology was originally developed to facilitate knowledge sharing and reuse [37]. Benefiting from ontology, greater ability of expression is provided for knowledge modeling and communicating knowledge between heterogeneous and distributed application systems. Therefore, the semantic Web can be viewed as a version of a Web of ontological contents and services, which includes machine-readable and human-transparent descriptions to the existing data and documents on the syntactic Web. In addition, the semantic Web supplies the necessary infrastructure and techniques for publishing, resolving and reasoning ontological descriptions of the contents and services.

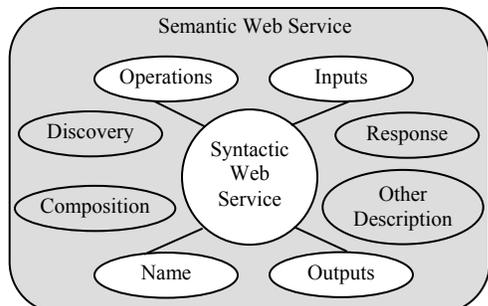


Figure 2. Syntactic Web Service and Semantic Web Service.

As for semantic Web services, besides the syntactic description, the information needed to select, compose, and respond to services are also encoded with semantic markup at the service Web sites. These efforts of service augmentation can then facilitate automated service discovery, composition, dynamic invocation and binding without human assistance or highly constrained agreements on protocols. Figure 2 illustrates the differences between syntactic and semantic Web services. Informally, a Web service can be characterized by its required inputs, the produced outputs, and the operations it will take [36]. The inputs and outputs may be further subject to pre-conditions and post-conditions respectively. With only descriptions in the syntactic level, as shown in the unfilled nodes of syntactic Web service in Figure 2, it is difficult for service providers and consumers to represent or interpret the

meaning of inputs, outputs and other applicable constraints. A semantic Web service relaxes such limitation by augmenting the service description with a rich set of formally semantic annotations of the service's capabilities, as shown in the grey nodes of semantic Web service in Figure 2. Accordingly, new standards and languages of semantic markup, like Web Ontology Language for Web Services (OWL-S) and Web Service Modeling Ontology (WSMO), should be investigated and used to give meaning to Web services.

Overall, the XML-based standards are for syntax, whilst the ontology-based standards are for semantics. Both share unified Web infrastructure and together provide capability for developing Web applications that deal with data and semantics. Nevertheless, one of the most important characteristics of ontology-based techniques is that they allow a richer integrability and interoperability of data in communications between domains. As previously mentioned, driven by the semantic markup and agent technologies, semantic Web service discovery, selection, composition, and execution are all supposed to be automatic tasks. Although fully automating these processes is still a challenge, accomplishing parts of this goal can still be achieved. For example, the semantic description is useful for the translation between Web service composition problems and AI-planning systems [13], while the semantic matchmaking can be used to facilitate the automatic Web service discovery [2]. Considering these outstanding characteristics, Web service compositions can be categorized according to syntactic and semantic context, while the context can be also identified through employed standards and techniques.

C. Mechanism: RESTful and SOAP-based Compositions

Concentrating on the technologies and architectures, nowadays there are two main mechanism paradigms of building composite Web services, namely *RESTful* composition and *SOAP-based* composition.

Basically, REpresentational State Transfer (REST) and Simple Object Access Protocol (SOAP) are not directly comparable with each other and not necessarily opposite. REST is an architectural style originally designed for building large-scale distributed hypermedia systems, whereas SOAP is a general protocol used as one foundation of numerous WS-* technologies. Within the REST environment, the Web is considered as a universal storage medium for publishing globally accessible information. In contrast, SOAP treats the Web as the universal transport mechanism for message exchange. When building Web services, traditional SOAP/WS-* environment requires relatively heavyweight open standards than that are being used in RESTful context. Although the SOAP vs. REST debate has been an ongoing discussion for some time, there is an implicit consensus that REST is more suitable for basic, ad-hoc, client-driven scenarios, while SOAP/WS-* are more suitable to address the quality of services requirements in highly interactive Web applications.

However, RESTful and SOAP-based Web services are indeed comparable. We can identify the differences between RESTful and SOAP-based Web services mainly through

their interfaces, the operations and Message Exchange Patterns (MEPs) behind interfaces, and their QoS support techniques.

1) *Interface differences.* The interface of a RESTful Web service comprises a variable set of Uniform Resource Identifiers (URIs). Each URI uses a globally unique address to identify a specific resource. Unfortunately, to the best of our knowledge, there is no standard and machine-processable way of describing RESTful interfaces. Using WSDL 2.0 description to wrap the RESTful Web services has been revealed as a burden for service consumers [32]. The Web Application Description Language (WADL) and other dedicated interface definition languages for RESTful services like RESTful Interface Definition and Declaration Language (RIDDL) [33] are not yet widely employed. Consequently, most of the time the interfaces of RESTful Web services are described through natural, informal, and more human-oriented documentations. When it comes to SOAP-based Web services, as mentioned previously, WSDL has gained widespread adoption to syntactically define the service interfaces. In a WSDL document, SOAP-based Web services are described as collections of network endpoints, or ports. A port associates a network address with a reusable binding. The reusable WSDL binding contains the concrete transport protocol and data format specifications for a particular port type. A port type is a set of abstract operations that are related to some abstract messages representing the data for exchange. Benefiting from the abstract interfaces described by WSDL, technical details of SOAP-based Web services can be concealed, for example, the implementation language, deployment platform and underlying communication protocol.

2) *Operation differences.* Since “REST is in many ways a retrospective abstracting of the principles that make the World Wide Web scalable” [34], RESTful Web services requires little technology support apart from well accepted HTTP and XML infrastructures. As a result, the manipulations of resources are completely constrained in the RESTful environment through a fixed set of four operations associated with HTTP: GET, PUT, DELETE, and POST. GET is used to retrieve a representation of the current state of a resource. PUT can either update the state of existing resource or create a new resource with the request URI if it does not previously exist. DELETE is used to delete a URI-identified resource and also invalidate the URI itself. POST creates subordinate resources to which new URIs are assigned by service provider. In contrast to the standard operations among RESTful Web services, the operations provided by SOAP-based Web services are ad hoc. Various APIs defined in different WSDL documentations represent different sets of operations for communication and interaction between service providers and consumers. The operations of SOAP-based Web services essentially are

functional components that are located on remote machines and can be invoked through APIs over the network.

3) *MEPs differences.* MEPs are patterns or templates that abstract the sequences of message transmission in the Web service context. Since REST is associated closely with HTTP, and HTTP is stateless request-response application protocol, RESTful Web services only have the synchronous request-response pattern under the HTTP mechanism. SOAP-based Web services allow rich patterns ranging from traditional request-response to broadcasting and sophisticated message exchanges. The latest WSDL 2.0 has been published with supporting eight MEPs [35]. Each MEP describes a bilateral message exchange between two involved services from a service point’s perspective.

- *In-Only* – The service receives a message.
- *Robust In-Only* – The service receives a message and will return a fault message only when meeting a fault.
- *In-Out* – The service receives a message and returns a response message.
- *In-Optional-Out* – The service receives a message and optionally returns a response message.
- *Out-Only* – The service sends a message.
- *Robust-Out-Only* – The service sends a message and will receive a fault message only when its partner service meets a fault.
- *Out-In* – The service sends a message and receives a response message.
- *Out-Optional-In* – The service sends a message and optionally receives a response message.

4) *QoS support technique differences.* Quality of Service (QoS) indicates a certain performance level of services that will be delivered to consumers, and can be evaluated through corresponding parameters like response time, throughput, cost, etc. As REST is usually used in conjunction with HTTP, the QoS of RESTful services are supported generally through basic protocols and techniques. For example, services’ interactions can be secured at the transport layer using the Secure Sockets Layer (SSL) protocol, while the security of messages can be guaranteed by encryption and digital signatures. On the contrary, SOAP-based Web services adopt more complicated mechanisms to cover QoS features. On the one hand, the header of an SOAP document contains message-layer infrastructure information that can be used for QoS configurations. On the other hand, the WS-* technology stack is employed to satisfy the large scope of QoS requirements such as transactions, security, and reliability. Benefiting from SOAP and WS-* technologies, QoS aspects of SOAP-based Web services are protocol transparent and independent. In other words, the QoS of Web service can be provided end to end without taking into account the variety of middleware systems transported.

All these differences between RESTful and SOAP-based Web services make the problem of RESTful Web service

composition fundamentally different from the composition problem of SOAP-based Web service. SOAP-based Web service composition is a collection of related, structured activities or tasks that produce a specific service or product for a particular customer. Within the relatively complex SOAP-based environment, a large number of standards and tools have been developed to facilitate the service composition activities. Dissimilarly, RESTful Web service composition integrates normally disparate Web resources to create a new application. These resources can be the exposure of pure data or traditional application functionality. With the constraint of lightweight technologies adopted in RESTful environment, service compositions mainly focus on the Web 2.0 Mashups that usually imply simple and fast integration of data/content from different sources on the Internet.

D. Design Time: Manual, Semi-Automatic and Automatic Compositions

Generally, there are four fundamental activities when composing a Web service, namely *Planning*, *Discovery*, *Selection*, and *Execution* [18]. *Planning* is to determine a composition plan including the execution sequence of tasks. Each task corresponds to either the functionality or activity of a service. *Discovery* is to find all the candidate services that can satisfy the tasks in the plan. The aim of *Selection* is to choose optimal subset from all the discovered services by using non-functional attributes. *Execution* builds a real composite Web service. In practice, the sequence of *Planning*, *Discovery*, and *Selection* can be diverse. For example, the theorem proving approach in [13] is based on the pre-determined Web services to generate the composition plan. Moreover, during the service composition procedure, the network configurations and non-functional factors may change, and existing Web services may be updated or terminated. As a result, some pre-identified services may not be available, and the new ones need to be re-selected or re-discovered. In other words, *Discovery* and *Selection* can also take place during or even after *Execution*. Therefore, we can define a potential *Adaptation* activity at the end of the procedure of Web service composition.

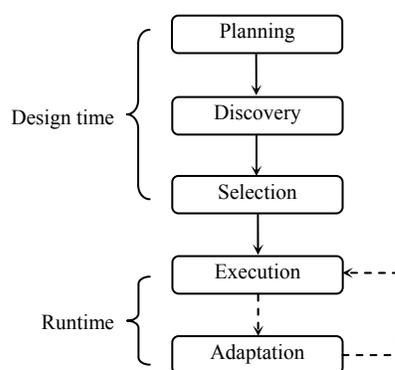


Figure 3. Stages of a Web Service Composition Scenario.

Based on the previous analysis, the process of Web service composition can be separated into design time and

runtime stages. Figure 3 shows one of the possible composition scenarios. Depending on the real practices, the design time stage comprises various activities from only *Planning* to the combination of *Planning*, *Discovery*, and *Selection*. According to the extent to which human intervention is involved, the design time procedure can be *manual*, *semi-automatic*, and *automatic*. Considering that there is still a long way to realize the complete automation of Web service composition even at design time, we mainly concentrate on the *Planning* activity when unfolding classification. Therefore, we can draw the outline of these three types of composition approaches during design time as:

1) *Manual approach*. In general, the manual *Planning* activity implies manual Web service composition. Two different scenarios of manual approaches can be further identified respectively as primitive level and abstract level respectively. In primitive manual composition approaches, developers have to specify every detailed activity in the composition processes. The resulting specifications are executable composition programs. For example, we can use BPEL to describe the procedure of Web service composition following the logic of corresponding business process, and the finalized description is executable with the support of BPEL engine. As for the manual composition approaches at an abstract level, the Web service composition plans are usually drawn into abstract workflows or models instead of specific programs. In such approaches the manual planning results cannot be executed directly, but can be transformed into executable specifications and finally executed by some tools or engines. Examples can be found in most of the UML related model-driven approaches.

2) *Automatic approach*. In general, the automatic *Planning* activity implies automatic Web service composition. In manual approaches discussed above, although we can decrease the effort of Web service composition through abstraction rather than programming, the planning phase still has to be realized manually. How to automatically generate the composition model or workflow then becomes a subsequent research topic. The current trend is to use Artificial Intelligence (AI) planning to satisfy the automation of the generation of a Web service composition plan. Benefiting from existing AI planning systems, the prerequisite effort of Web service composition is only to encode the requirements into dedicated, formal, and mathematical expressions.

3) *Semi-automatic approach*. We treat an instance of Web service composition as semi-automatic approach, if one of the following cases is met: (1) there are specifically automatic *Discovery/Selection* activities to facilitate manual *Planning*; or (2) there are specifically manual *Discovery/Selection* activities that constrain automatic *Planning*. Taking [2] as an example of the former case, semantic matchmaking technique is used to realize the semi-automatic approach by automatically filtering and presenting matching services to the user at each step of a

composition. An example of the latter case can be found in [13], the theorem proving technique requires manually pre-determining Web services before automatically generating the composition plan.

E. Runtime: Static and Dynamic Compositions

The *Execution* and potential *Adaptation* activities remain at the runtime stage of Web service composition. By focusing on the *Adaptation* activity, we can define that the Web service composition is *dynamic* at runtime if it is adaptive with minimal user intervention, otherwise it is *static*. In detail, static Web service composition re-discovers and re-selects new services manually when adapting the environment. In the worst case, static composition does not have adaptability at all. On the contrary, dynamic composition can re-discover and re-select new services at runtime without requiring any human assistance. Moreover, we also define a dynamic Web service composition if services can be discovered and selected during *Execution* activity, for instance eFlow [3].

Benefiting from the division between the design time and runtime of Web service compositions, we can clearly distinguish the two concepts: automatic and dynamic compositions that are confusing in the existing literature. Furthermore, it can be found that there is no relationship between automatic composition at design time and dynamic composition at runtime. On the one hand, automatic composition does not imply dynamic composition, for example, most of the AI planning approaches only concentrate on the automatic *Planning* process while leaving the planning result executed statically. On the other hand, static composition does not require automatic composition, for example, the visual language UML Profile for Web Service Composition (UML-WSC) [7] supports dynamically composing Web services although the composition model is still built manually.

IV. TECHNOLOGY-BASED CLASSIFICATION OF WEB SERVICE COMPOSITION

Technology refers to the techniques used in the approaches to implement Web service composition. It is difficult to enumerate all kinds of composition techniques, although different technique can contribute different composition effort. However, we can identify three groups of techniques: Workflow-based, Model-driven, and AI planning techniques.

A. Workflow-based Techniques

Workflow is a virtual representation of actual work including a sequence of operations. Workflow-based Web service composition uses the workflow perspective to describe the normally complex collaboration among Web services and implement the composition procedure. There are two ways to describe the Web service composition workflow:

- *To program the executable workflow directly:* Obviously, the composition process can be programmed from scratch by using traditional

languages and standards. However, the current universal technique is to use the dedicated, process-oriented language, for example the current de facto executable business process modeling language BPEL, to specify the transition interactions among Web services at a macro-level state.

- *To draw the abstract workflow without programming:* Supported by some tools and engines, the workload of Web service composition can be relieved by drawing the abstract workflow without programming. For example, the semantic matchmaking based approach [2] uses the GUI panel of composer to construct an abstract flow, while eFlow [3] adopts a graph-oriented method to define the interaction and order of execution among the nodes in an abstract composition process.

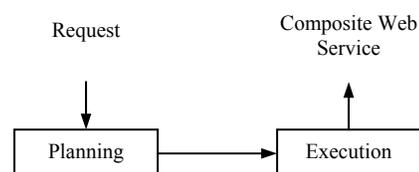


Figure 4. One-Stop Process of Web Service Composition.

If we only focus on the two main activities (*Planning* and *Execution*) in the Web service composition approaches, workflow-based techniques generally follow the One-Stop process, as shown in Figure 4. In the One-Stop process, the *Planning* activity happens just after receiving the composition requirement, and delivers the executable composition specification directly. In most cases of One-Stop based approaches, during the planning stage the user must provide inputs at choice points, decide the interoperation among component Web services, and specify the composition procedure.

B. Model-driven Techniques

In model-driven approaches of Web service composition, models are used to describe user requirements, information structures, abstract business processes, component services and component service interactions. The models are independent of, but can be transformed into, executable composition specifications. Generally, there is also modeling work in several workflow-based techniques. Whereas the model-driven techniques discussed here merely follow the standards provided by the Object Management Group (OMG). The standards mainly refer to the Unified Modeling Language (UML) and Model-Driven Architecture (MDA).

Numerous discussions related to UML-based modeling of Web service composition can be found in the literature. Through analysis and abstraction, we can further identify two basic scenarios of model-driven approaches for composing Web services.

- *To build executable composition model.* A typical example of this particular scenario is the UML-WSC profile [7]. The UML-WSC profile is a well-defined UML extension, which uses a static model and

extended variant of activity diagrams to define the process-oriented Web service composition. The static model describes the available Web services and components, while the extended variant of activity diagrams describes the composition processes. The composition model specified through UML-WSC profile can be executed automatically by a process engine. Therefore, the UML-WSC profile is also considered as an alternative to non-visualized languages like BPEL.

- *To build transformable composition model.* This generic scenario is to use UML class diagrams to represent the state parts of compositions, while the behaviour parts are represented through UML activity diagrams. The state parts can be Web service interface [4], the structure of composite Web service [5] and QoS characteristics [6]. On the other hand, the behaviour parts describe the composition operations, interactions of component Web services, and control and data flow. Furthermore, since BPEL is widely accepted for composing Web services, UML has been designedly to extend BPEL to include common aspects of Web service composition. Therefore, the modeling results can be conveniently transformed into executable BPEL specifications to eventually realize Web service compositions.

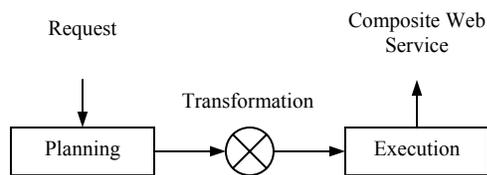


Figure 5. Bridge Process of Web Service Composition.

Although the former, particular scenario of model-driven approach still employs the One-Stop process for Web service composition, most of the existing modeling techniques adopt the Bridge process when composing Web services, as illustrated in Figure 5. The Bridge process can be viewed as an evolution from the One-Stop process, which describes such approaches that plan Web service compositions at an abstract level, while the planning results cannot be directly executed and have to be transformed into executable specifications. Therefore, unlike the first scenario of model-driven approaches employing the One-Stop process, any Web service composition approach adopting the Bridge process uses a transformation procedure for the mapping between the planning result and executable specification. The notion of the Bridge process is that the planning phase of Web service composition does not need to be tied to any particular composition language and execution engine, and thereby the same planning result can be transformed into more than one executable description.

C. AI Planning Techniques

AI planning seeks to use intelligent systems to generate a plan that can be one possible solution to a specified problem, while a plan is an organized collection of operators within the given application domain. AI planning is essentially a search problem. The underlying basis of planning relies on state transition system with states, actions and observations. Benefiting from the state transition system, the planner explores a potentially large search space and produces a plan that is applicable to bridge the gap between the initial state and goal when run. AI planning in Web service composition normally comprises of five attributes, they are (1) all the available services, (2) the initial state, (3) the state change functions, (4) all the possible states, and (5) the final goal. The initial state and final goal are specified in the requirements for composing Web service. The state change functions define the preconditions and effects when invoking Web services.

A large amount of research has been reported about the AI planning related Web service composition. These works apply techniques ranging from Situation Calculus [8], Automata Theory [9], Rule-based Planning [10], Query Planning [12], Theorem Proving [13], Petri Nets [14], to Model Checking [15]. Generally, these techniques convert the problems of composition into generating execution workflows using the dedicated expression. The workflows can then be transformed into executable specifications like BPEL documents or other XML-based descriptions, and executed through the corresponding engines.

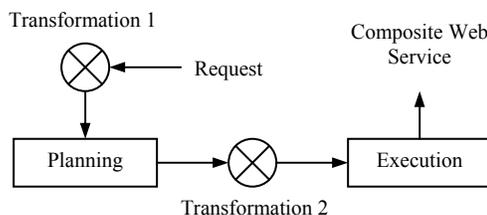


Figure 6. Double-Bridge Process of Web Service Composition.

Therefore, we can find that the Web service composition approaches using AI planning techniques normally contain the Double-Bridge process, as shown in Figure 6. The Double-Bridge process can be treated as further evolution from the Bridge process. The *Planning* activity is settled between two transformation procedures in a Double-Bridge process. In detail, since AI planning systems generally adopt dedicated, formal, and mathematical techniques, the initial information and composition requirement must be transformed for input into a planning system, and the planning result should be transformed again into an executable specification to build a composite Web service.

V. QUALITATIVE DISCUSSION ABOUT EFFORT ESTIMATION FOR WEB SERVICE COMPOSITION

Through categorizing Web service composition approaches along Context and Technology dimensions, a classification matrix can be established, as demonstrated in

Appendix I. Considering the different influences of different contexts and techniques on the composition effort, those technology categories and context types in the classification matrix can be viewed as effort factors when composing Web services. Therefore, we can use the classification matrix to facilitate the cost and effort estimation for different Web service composition approaches. Since the data we collected here are all based on qualitative descriptions, it is not suitable to do quantitative work for composition effort estimation. Through analyzing these qualitative descriptions, however, we can further build a checklist for experts to judge qualitatively the effort when implementing Web service compositions. Before building the qualitative effort estimation checklist, some effort related hypotheses should be investigated.

A. Qualitative Effort Estimation Hypotheses

In the context of software engineering, effort of a task is generally accounted by calculating how long and how many workers are needed to finish the task, and the unit can be person-day, person-month, or person-year. In brief, the amount of human activities in a project is proportional to the amount of effort required to finish the project. Therefore, for a certain software project, we can hypothesize:

- H1. The increase of human activities in a project will have a proportional impact on the final effort.

Human activities include both physical and mental activities. Since software engineering is a knowledge-intensive domain, the effort of a software project is mainly composed of mental activities. Unfortunately, within a given time span people have limited mental capability to deal with information [39]. For every single person, the increased amount of information beyond a certain point may even defeat his/her mental ability, and hence result in errors [41]. As a result, the more information that exists in a project, the more people and human activities will be required to perform accurate manipulations. Together with H1, therefore, we can hypothesize:

- H2. The increase of information in a project will have a proportional impact on the required human activities.

- H2'. The increase of information in a project will have a proportional impact on the final effort.

Moreover, complexity has been proved to be a significant and non-negligible factor that influences software development and maintenance [42]. Meanwhile, the more complexity involved in a system, the more difficulty the designers or engineers have to understand the implementation process and thus the system itself [40], and hence the greater mental effort people have to exert to solve the complexity [39]. To summarize, we can further hypothesize:

- H3. The increase of complexity in a project will have a proportional impact on the final effort.

When it comes to project complexity, one of the main contributors is the complexity of the methods that regard achieving the project goals [43]. The methods mentioned herein generally consist of processes, tools, and techniques that are used to complete the corresponding project [44]. In particular, processes and techniques have been viewed as internal environment of a system (organization), while the system's complexity is considered a response to the environmental complexity [45]. Consequently, the complexity of processes and techniques involved in a software project will positively influence the complexity of the project. As for the tools, although the adoption of sophisticated tools usually implies a complex project, tools are essentially developed and used to save human activities. For a certain project, the more work the tools can fulfill, the less human activities the project will require. Overall, we can also hypothesize:

- H4. The increase of process complexity in a project will have a proportional impact on the project complexity.

- H4'. The increase of process complexity in a project will have a proportional impact on the final effort.

- H5. The increase of difficulty of techniques in a project will have a proportional impact on the project complexity.

- H5'. The increase of difficulty of techniques in a project will have a proportional impact on the final effort.

- H6. The increase of work that tools can fulfill in a project will have an inversely proportional impact on the human activities.

- H6'. The increase of work that tools can fulfill in a project will have an inversely proportional impact on the final effort.

B. Qualitative Effort Estimation Checklist for Web service composition approaches

As mentioned earlier, we treat technology categories and context types in the classification matrix as effort factors of Web service composition approaches. After applying different effort estimation hypotheses to different but comparable factors, a set of qualitative effort estimation statements will be generated. These statements can then constitute a checklist for developers and engineers to qualitatively judge and compare the effort and cost of different composition strategies. In fact, using a checklist has been considered a simple way of utilizing experience and advocated as an efficient method of improving expert judgment processes when doing estimation [48]. To facilitate

building this qualitative effort estimation checklist, some symbols and rules are also proposed:

For one certain task of Web service composition, we use E_{F-H} to represent the effort E determined by factor F when applying hypothesis H . Moreover, a score S will be set for E_{F-H} to flag different effort determined by different but comparable factors when applying some hypothesis. For convenience of calculation, the rules of score setting can be:

$$\begin{cases} S(E_{F1-H}) = 2, S(E_{F2-H}) = 1 & \text{if } E_{F1-H} > E_{F2-H} \\ S(E_{F1-H}) = 1, S(E_{F2-H}) = 1 & \text{if } E_{F1-H} \approx E_{F2-H} \\ S(E_{F1-H}) = 1, S(E_{F2-H}) = 2 & \text{if } E_{F1-H} < E_{F2-H} \end{cases} \quad (1)$$

Note that if we use E_F to represent the effort E determined by factor F under all the different but applicable hypotheses, then all the scores for E_F under corresponding hypotheses can be summed up and represented as $S(E_F)$.

We can hereby build the effort estimation checklist following the sequence of building the classification matrix.

1) *For Orchestration and Choreography:* As analyzed previously, orchestration stands for a central coordination, while choreography represents multiparty collaborations. Since distributed processing would be inevitably more complicated than non-distributed processing [46], for a same Web service composition project choreography requires more effort than orchestration if applying H3. Meanwhile, as the current de facto standard of orchestrating Web services, BPEL stemmed from existing languages and tools and has been widely accepted, whereas the choreography language WS-CDL was developed without any prior implementation and is still far from maturity [47]. Considering this technical influence, the implementation of choreography will be more difficult than that of orchestration. By using *For* for representing the effort factor Orchestration and *Fch* for Choreography, the effort compare and scores can be listed in Table I.

TABLE I. EFFORT COMPARE BETWEEN ORCHESTRATION AND CHOREOGRAPHY

Applied Hypotheses	Compare	Scores
H3	$E_{For-H3} < E_{Fch-H3}$	$S(E_{For-H3})=1, S(E_{Fch-H3})=2$
H5'	$E_{For-H5'} < E_{Fch-H5'}$	$S(E_{For-H5'})=1, S(E_{Fch-H5'})=2$
Total	$E_{For} < E_{Fch}$	$S(E_{For})=2, S(E_{Fch})=4$

2) *For Syntactic and Semantic Compositions:* Since semantic Web and semantic Web services are proposed to automate service discovery, selection, composition and execution by adding the inherent meanings, human activities within semantic compositions will be decreased while the involved information will be increased. Considering the increased information is for machine interpretation rather than human intervention, however, hypothesis H2 is not applicable here. Meanwhile, syntactic and semantic Web

services share the unified Web infrastructure and both use markup language based techniques to describe information. It can then be stated that the difficulty levels of techniques adopted in both syntactic and semantic service compositions are similar. Therefore, by using *Fsy* for representing the effort factor Syntax and *Fse* for Semantics, the effort compare and scores can be listed in Table II.

TABLE II. EFFORT COMPARE BETWEEN SYNTACTIC AND SEMANTIC COMPOSITION APPROACHES

Applied Hypotheses	Compare	Scores
H1	$E_{Fsy-H1} < E_{Fse-H1}$	$S(E_{Fsy-H1})=1, S(E_{Fse-H1})=2$
H5'	$E_{Fsy-H5'} \approx E_{Fse-H5'}$	$S(E_{Fsy-H5'})=1, S(E_{Fse-H5'})=1$
Total	$E_{Fsy} < E_{Fse}$	$S(E_{Fsy})=2, S(E_{Fse})=3$

3) *For SOAP-based and RESTful Compositions:* Compared with RESTful Web service compositions, SOAP-based compositions employ more sophisticated techniques including heavyweight protocols, a set of WS-* stack, and more MEPs, which can satisfy more QoS requirements while also deal with more information. Therefore, the hypotheses H2' and H5' are both applicable. Incidentally, although the SOAP/WS-* related techniques indeed are complex, they should still be adopted when addressing advanced requirements especially in the enterprise computing scenarios. However, here we only focus on the implementation effort without considering other tradeoffs. By using *Fso* for representing the effort factor SOAP and *Fre* for REST, the effort compare and scores can be listed in Table III.

TABLE III. EFFORT COMPARE BETWEEN SOAP-BASED AND RESTFUL COMPOSITION APPROACHES

Applied Hypotheses	Compare	Scores
H2'	$E_{Fso-H2'} > E_{Fre-H2'}$	$S(E_{Fso-H2'})=2, S(E_{Fre-H2'})=1$
H5'	$E_{Fso-H5'} > E_{Fre-H5'}$	$S(E_{Fso-H5'})=2, S(E_{Fre-H5'})=1$
Total	$E_{Fso} > E_{Fre}$	$S(E_{Fso})=4, S(E_{Fre})=2$

4) *For Manual, Semi-Automatic, and Automatic Compositions:* During the design time of Web service compositions, the more automated the design processes are, the less human activities the compositions will require, and the less detailed information developers need be concerned with. Considering the realization of automation usually requires assistant tools and more techniques, for example the Semantic Matching approach [2], the hypotheses H5' and H6' are both applicable together with H1 and H2'. By using *Fma* for representing the effort factor Manual, *Fsa* for Semi-Auto and *Fau* for Auto, the effort compare and scores can be listed in Table IV.

TABLE IV. EFFORT COMPARE BETWEEN MANUAL, SEMI-AUTOMATIC AND AUTOMATIC COMPOSITION APPROACHES

Applied Hypotheses	Compare	Scores
H1	$E_{Fma-H1} > E_{Fsa-H1}$ $E_{Fma-H1} > E_{Fau-H1}$ $E_{Fsa-H1} > E_{Fau-H1}$	$S(E_{Fma-H1})=2+2=4$ $S(E_{Fsa-H1})=1+2=3$ $S(E_{Fau-H1})=1+1=2$
H2'	$E_{Fma-H2'} > E_{Fsa-H2'}$ $E_{Fma-H2'} > E_{Fau-H2'}$ $E_{Fsa-H2'} > E_{Fau-H2'}$	$S(E_{Fma-H2'})=2+2=4$ $S(E_{Fsa-H2'})=1+2=3$ $S(E_{Fau-H2'})=1+1=2$
H5'	$E_{Fma-H5'} < E_{Fsa-H5'}$ $E_{Fma-H5'} < E_{Fau-H5'}$ $E_{Fsa-H5'} < E_{Fau-H5'}$	$S(E_{Fma-H5'})=1+1=2$ $S(E_{Fsa-H5'})=2+1=3$ $S(E_{Fau-H5'})=2+2=4$
H6'	$E_{Fma-H6'} > E_{Fsa-H6'}$ $E_{Fma-H6'} > E_{Fau-H6'}$ $E_{Fsa-H6'} > E_{Fau-H6'}$	$S(E_{Fma-H6'})=2+2=4$ $S(E_{Fsa-H6'})=1+2=3$ $S(E_{Fau-H6'})=1+1=2$
Total	$E_{Fma} > E_{Fsa} > E_{Fau}$	$S(E_{Fma})=14, S(E_{Fsa})=12,$ $S(E_{Fau})=10$

5) *For Static and Dynamic Compositions:* If we emphasize the adaptation in both static and dynamic compositions during runtime, we can draw the same conclusions through the similar analysis as above. Therefore, by using *Fst* for representing the effort factor Static and *Fdy* for Dynamic, the effort compare and scores can be listed in Table V.

TABLE V. EFFORT COMPARE BETWEEN STATIC AND DYNAMIC COMPOSITION APPROACHES

Applied Hypotheses	Compare	Scores
H1	$E_{Fst-H1} > E_{Fdy-H1}$	$S(E_{Fst-H1})=2, S(E_{Fdy-H1})=1$
H2'	$E_{Fst-H2'} > E_{Fdy-H2'}$	$S(E_{Fst-H2'})=2, S(E_{Fdy-H2'})=1$
H5'	$E_{Fst-H5'} < E_{Fdy-H5'}$	$S(E_{Fst-H5'})=1, S(E_{Fdy-H5'})=2$
H6'	$E_{Fst-H6'} > E_{Fdy-H6'}$	$S(E_{Fst-H6'})=2, S(E_{Fdy-H6'})=1$
Total	$E_{Fst} > E_{Fdy}$	$S(E_{Fst})=7, S(E_{Fdy})=5$

6) *For Workflow-based, Model-driven and AI Planning Compositions:* To simplify the effort analysis in the Technology dimension, we constrain that workflow-based approaches strictly follow the One-Stop process, model-driven approaches strictly follow the Bridge process, and AI planning approaches strictly follow the Double-Bridge process. Considering that the One-Stop process delivers executable specifications, the Bridge process focuses on the abstract modeling, and the Double-Bridge process focuses on the composition requirement, workflow-based approaches have to deal with the most information while AI planning approaches deal with the least information for one certain task of Web service composition. Meanwhile, AI planning approaches have the longest processes while workflow-based approaches have the shortest. However, we can imagine that both One-Stop and Bridge processes also contain two transformation procedures as well as the Double-Bridge process does. The intangible transformation

procedures essentially take place as mental activities, while the tangible ones can be supported by tools. Therefore, it can be found that AI planning approaches require less human activities and use more tools, workflow-based approaches require more human activities and use fewer tools, while model-driven approaches are in the middle. When it comes to techniques, it is nearly impossible to compare the difficulty levels of workflow, modeling and AI planning with each other. Consequently, here we simply treat their difficulties similarly. After applying all the suitable hypotheses and using *Fwf* for representing the effort factor Workflow-based, *Fmd* for Model-Driven and *Fai* for AI Planning, the effort compare and scores can be listed in Table VI.

TABLE VI. EFFORT COMPARE BETWEEN WORKFLOW-BASED, MODEL-DRIVEN AND AI PLANNING COMPOSITION APPROACHES

Applied Hypotheses	Compare	Scores
H1	$E_{Fwf-H1} > E_{Fmd-H1}$ $E_{Fwf-H1} > E_{Fai-H1}$ $E_{Fmd-H1} > E_{Fai-H1}$	$S(E_{Fwf-H1})=2+2=4$ $S(E_{Fmd-H1})=1+2=3$ $S(E_{Fai-H1})=1+1=2$
H2'	$E_{Fwf-H2'} > E_{Fmd-H2'}$ $E_{Fwf-H2'} > E_{Fai-H2'}$ $E_{Fmd-H2'} > E_{Fai-H2'}$	$S(E_{Fwf-H2'})=2+2=4$ $S(E_{Fmd-H2'})=1+2=3$ $S(E_{Fai-H2'})=1+1=2$
H4'	$E_{Fwf-H4'} < E_{Fmd-H4'}$ $E_{Fwf-H4'} < E_{Fai-H4'}$ $E_{Fmd-H4'} < E_{Fai-H4'}$	$S(E_{Fwf-H4'})=1+1=2$ $S(E_{Fmd-H4'})=2+1=3$ $S(E_{Fai-H4'})=2+2=4$
H5'	$E_{Fwf-H5'} \approx E_{Fmd-H5'}$ $E_{Fwf-H5'} \approx E_{Fai-H5'}$ $E_{Fmd-H5'} \approx E_{Fai-H5'}$	$S(E_{Fwf-H5'})=1+1=2$ $S(E_{Fmd-H5'})=1+1=2$ $S(E_{Fai-H5'})=1+1=2$
H6'	$E_{Fwf-H6'} > E_{Fmd-H6'}$ $E_{Fwf-H6'} > E_{Fai-H6'}$ $E_{Fmd-H6'} > E_{Fai-H6'}$	$S(E_{Fwf-H6'})=2+2=4$ $S(E_{Fmd-H6'})=1+2=3$ $S(E_{Fai-H6'})=1+1=2$
Total	$E_{Fwf} > E_{Fmd} > E_{Fai}$	$S(E_{Fwf})=16, S(E_{Fmd})=14,$ $S(E_{Fai})=12$

To reflect the combined influences of different factors on the composition effort, we further define that the scores for different effort factors are accumulable in the same dimension, while they are multipliable across different dimensions. After filling the applicable hypotheses and scores to the classification matrix, we can achieve an effort-estimation-checklist table, as shown in Appendix II. Note that the numbers do NOT indicate any count of the amount of effort. These quantitative scores are only used to facilitate qualitatively contrasting the effort of different composition approaches, as demonstrated in Table VII.

Through Table VII, we can conveniently compare the estimated effort between different Web service composition approaches: one composition approach requires more effort than another does if the former's effort score is bigger than the latter's. Moreover, by investigating the result and procedure of calculation of the effort scores, we can find that the amount of applicable hypotheses implies the times of comparisons, while the times of consistent comparisons is proportional to the resulting effort score. Here we regard different comparisons are consistent when the same conclusion can be drawn in these comparisons by applying

different hypotheses. For example, there are two consistent comparisons when applying hypotheses H3 and H5' to the compare between Orchestration and Choreography in Table I. Since the consistent comparisons can help to confirm and reinforce the comparison result, the effort scores also reflect the extent of our confidence in the effort estimation result. Therefore, the larger difference between two approach effort scores, the more confidence we will have in the comparison result.

TABLE VII. EFFORT COMPARE BETWEEN DIFFERENT COMPOSITION APPROACHES

Composition Approaches	Approach Effort Scores
BPEL Programming	$S(E_{Fwf}) \times (S(E_{For}) + S(E_{Fsy}) + S(E_{Fso}) + S(E_{Fma}) + S(E_{Fst}))$ = $16 \times 29 = 464$
Semantic Matching [2]	$S(E_{Fwf}) \times (S(E_{Fch}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fsa}) + S(E_{Fst}))$ = $16 \times 30 = 480$
SA-REST + Smashup [21]	$S(E_{Fwf}) \times (S(E_{For}) + S(E_{Fse}) + S(E_{Fre}) + S(E_{Fsa}) + S(E_{Fst}))$ = $16 \times 26 = 416$
RESTfulBP [28]	$S(E_{Fwf}) \times (S(E_{Fch}) + S(E_{Fsy}) + S(E_{Fre}) + S(E_{Fma}) + S(E_{Fst}))$ = $16 \times 29 = 464$
UML + MDA [4]	$S(E_{Fmd}) \times (S(E_{For}) + S(E_{Fsy}) + S(E_{Fso}) + S(E_{Fma}) + S(E_{Fst}))$ = $14 \times 29 = 406$
UML + OCL [5]	$S(E_{Fmd}) \times (S(E_{For}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fma}) + S(E_{Fdy}))$ = $14 \times 28 = 392$
UML + QoS Support [6]	$S(E_{Fmd}) \times (S(E_{For}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fsa}) + S(E_{Fst}))$ = $14 \times 28 = 392$
UML + IHE framework [22]	$S(E_{Fmd}) \times (S(E_{For}) + S(E_{Fsy}) + S(E_{Fso}) + S(E_{Fma}) + S(E_{Fdy}))$ = $14 \times 27 = 378$
Petri Net [23]	$S(E_{Fai}) \times (S(E_{Fch}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fau}) + S(E_{Fst}))$ = $12 \times 28 = 336$
Interface Automata [11]	$S(E_{Fai}) \times (S(E_{For}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fau}) + S(E_{Fst}))$ = $12 \times 26 = 312$
AIMO [24]	$S(E_{Fai}) \times (S(E_{Fch}) + S(E_{Fse}) + S(E_{Fso}) + S(E_{Fau}) + S(E_{Fdy}))$ = $12 \times 26 = 312$
...	...

In fact, the calculation rule here for counting the effort scores of different Web service composition approaches are mainly inspired by the Addition and Multiplication principles in Combinatorics: (1) We apply an Addition-principle-like method to the effort factors in the Context dimension of the classification matrix, considering that different partial efforts of one Web service composition within different contexts are mutually exclusive, while different contexts are accumulable. (2) We apply a Multiplication-principle-like method to the effort factors across those two dimensions of the classification matrix, considering that the Technology dimension is independent of the Context dimension, and one technique can be used to compose Web services within any combination of contexts. However, this calculation rule still suffers from intuition, and will be further validated and revised through empirical study in our future work.

VI. CONCLUSION

The territory of Web service composition has been researched so broadly that it becomes difficult to analyze and

estimate the composition effort by exploring every existing composition approach. However, we are able to deliver a general classification of Web service composition to facilitate the effort estimation work through investigating limited approaches inductively. Unlike existing classification work, this paper proposes an effort-oriented classification matrix of Web service composition through a systematic review. Some of the reviewed composition approaches are then classified according to their published descriptions, as demonstrated in Appendix I. The matrix uses two dimensions, Context and Technology, to classify different compositions. Several pairs of effort-related contexts are selected in the Context dimension, while three technology categories are paralleled in the Technology dimension. Moreover, this paper also builds an effort-estimation-checklist table by applying a set of effort estimation hypotheses to different context types and technology categories that are viewed as different composition effort factors. The combined influences of factor pairs across Context dimension and Technology dimension on the composition effort are also represented in this table. The effort-oriented classification matrix can be used to facilitate exploration and comprehension in the research area of Web service composition, while the effort-estimation-checklist table can be used to facilitate the qualitative effort compare between different composition approaches. Furthermore, based on our current work, some new research opportunities in the Web service composition area can also be identified. For example, the gap between automatic composition at design time and dynamic composition at runtime should be bridged.

Overall, the work described in this paper not only brings a new perspective of classification of Web service composition, but also introduces a new method to compare the qualitatively estimated effort between different composition approaches. The prominent characteristic of the proposed classification matrix is of our primary objective - aiming at the influence on software development effort required for different Web service compositions. As such, the classification matrix is eventually developed into an effort-estimation-checklist table, while the effort-estimation-checklist table should be applied closely with the classification matrix. Our future work is to continue filling this classification matrix and to use the effort-estimation-checklist table to establish the basis of the research into cost and effort estimation for Web service composition.

ACKNOWLEDGMENT

This paper is based on our previous work in collaboration with Jacky Keung and Xiwei Xu. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

REFERENCES

- [1] Z. Li, L. O'Brien, J. Keung, and X. Xu, "Effort-Oriented Classification Matrix of Web Service Composition," Proc. the Fifth International Conference on Internet and Web Applications and

- Services (ICIW 2010), IEEE Computer Society, June 2010, pp. 357-362, doi: 10.1109/ICIW.2010.59.
- [2] E. Sirin, J. Hendler, and B. Parsia, "Semi-automatic Composition of Web Services using Semantic Descriptions," *Web Services: Modeling, Architecture and Infrastructure workshop in ICEIS 2003*, Apr. 2003.
 - [3] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, and M. Shan, "Adaptive and Dynamic Service Composition in EFlow," *Proc. 12th International Conference on Advanced Information Systems Engineering (CaiSE*00)*, Springer, Jun. 2000, pp. 13-31, doi: 10.1007/3-540-45140-4_3.
 - [4] D. Skogan, R. Groenmo, and I. Solheim, "Web Service Composition in UML," *Proc. Eighth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004)*, IEEE Computer Society, Sept. 2004, pp. 47-57, doi: 10.1109/EDOC.2004.1342504.
 - [5] J. T. E. Timm and G. C. Gannod, "Specifying Semantic Web Service Compositions using UML and OCL," *Proc. 2007 IEEE International Conference on Web Services (ICWS 2007)*, IEEE Computer Society, Jul. 2007, pp. 521-528, doi: 10.1109/ICWS.2007.168.
 - [6] R. Grønmo and M. C. Jaeger, "Model-driven Semantic Web Service Composition," *Proc. 12th Asia-Pacific Software Engineering Conference (APSEC '05)*, IEEE Computer Society, Dec. 2005, pp. 15-17, doi: 10.1109/APSEC.2005.81.
 - [7] S. Thone, R. Depke, and G. Engels, "Process-Oriented, Flexible Composition of Web Services with UML," *Proc. Third International Joint Workshop on Conceptual Modeling Approaches for E-business: A Web Service Perspective (eCOMO 2002)*, Springer, Oct. 2002, pp. 390-401, doi: 10.1007/b12013.
 - [8] V. R. Chifu, I. Salomie, and E. St. Chifu, "Fluent Calculus-based Web Service Composition — From OWL-S to Fluent Calculus," *Proc. 4th International Conference on Intelligent Computer Communication and Processing (ICCP 2008)*, IEEE Computer Society, Aug. 2008, pp. 161-168, doi: 10.1109/ICCP.2008.4648368.
 - [9] S. Mitra, R. Kumar, and S. Basu, "Automated Choreographer Synthesis for Web Services Composition Using I/O Automata," *Proc. IEEE International Conference on Web Services (ICWS 2007)*, IEEE Computer Society, Jul. 2007, pp. 364-371, doi: 10.1109/ICWS.2007.47.
 - [10] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid, "Composing Web services on the Semantic Web," *The VLDB Journal*, vol. 12, Sept. 2003, pp. 333-351, doi: 10.1007/s00778-003-0101-5.
 - [11] S. V. Hashemian and F. Mavaddat, "A Graph-based Approach to Web Services Composition," *Proc. The 2005 Symposium on Applications and the Internet*, IEEE Computer Society, Jan.-Feb. 2005, pp. 183-189, doi: 10.1109/SAINT.2005.4.
 - [12] S. Thakkar, C. Knoblock, and J. Ambite, "A View Integration Approach to Dynamic Composition of Web Services," *Proc. 2003 ICAPS Workshop on Planning for Web Services*, AAAI Press, 2003.
 - [13] J. Rao, P. Küngas, and M. Matskin, "Composition of Semantic Web Services using Linear Logic Theorem Proving," *Information Systems*, vol. 31, Jun.-Jul. 2006, pp. 340-360, doi: 10.1016/j.is.2005.02.005.
 - [14] V. Gehlot and K. Edupuganti, "Use of Colored Petri Nets to Model, Analyze, and Evaluate Service Composition and Orchestration," *Proc. 42nd Hawaii International Conference on System Sciences (HICSS'09)*, IEEE Computer Society, Jan. 2009, pp. 1-8, doi: 10.1109/HICSS.2009.487.
 - [15] P. Traverso and M. Pistore, "Automated Composition of Semantic Web Services into Executable Processes," *Proc. Third International Semantic Web Conference (ISWC'04)*, Nov. 2004, pp. 380-394.
 - [16] P. Sarang, F. Jennings, M. Juric, and R. Loganathan, *SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects*. Birmingham: Packt Publishing, 2007.
 - [17] S. Dustdar and W. Schreiner, "A Survey on Web Services Composition," *International Journal of Web and Grid Services*, vol. 1, Aug. 2005, pp. 1-30, doi: 10.1504/IJWGS.2005.007545.
 - [18] J. Cardoso and A. P. Sheth, *Semantic Web Services, Processes and Applications*. New York: Springer, 2006.
 - [19] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," *Lecture Notes in Computer Science*, vol. 3387/2005, Jan. 2005, pp. 43-54, doi: 10.1007/b105145.
 - [20] F. Rosenberg, F. Curbera, M. J. Duftler, and R. Khalaf, "Composing RESTful Services and Collaborative Workflows: A Lightweight Approach," *IEEE Internet Computing*, vol. 12, Sept.-Oct. 2008, pp. 24-31, doi: 10.1109/MIC.2008.98.
 - [21] J. Lathem, K. Gomadam, and A. P. Sheth, "SA-REST and (S)mashups : Adding Semantics to RESTful Services," *Proc. First IEEE International Conference on Semantic Computing (ICSC 2007)*, IEEE Computer Society, Sept. 2007, pp. 469-476, doi: 10.1109/ICSC.2007.94.
 - [22] R. Anzboeck and S. Dustdar, "Semi-Automatic Generation of Web Services and BPEL Processes - A Model-Driven Approach," *Lecture Notes in Computer Science*, vol. 3649/2005, Sept. 2005, pp. 64-79, doi: 10.1007/11538394_5.
 - [23] V. Valero, M. E. Cambronero, G. Díaz, and H. Macià, "A Petri Net Approach for the Design and Analysis of Web Services Choreographies," *Journal of Logic and Algebraic Programming*, vol. 78, May-Jun. 2009, pp. 359-380, doi: 10.1016/j.jlap.2008.09.002.
 - [24] S. G. H. Tabatabaei, W. M. N. Kadir, and S. Ibrahim, "Semantic Web Service Discovery and Composition Based on AI Planning and Web Service Modeling Ontology," *Proc. IEEE Asia-Pacific Services Computing Conference (APSCC '08)*, IEEE Computer Society, Dec. 2008, pp. 397-403, doi: 10.1109/APSCC.2008.126.
 - [25] 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, Oct. 2004, pp. 377-396, doi: 10.1016/j.websem.2004.06.005.
 - [26] H. Zhao and P. Doshi, "Towards Automated RESTful Web Service Composition," *Proc. IEEE International Conference on Web Services (ICWS 2009)*, IEEE Computer Society, Jul. 2009, pp. 189-196, doi: 10.1109/ICWS.2009.111.
 - [27] F. Casati, M. Sayal, and M. Shan, "Developing E-Services for Composing E-Services," *Lecture Notes in Computer Science*, vol. 2068/2001, Jan. 2001, pp. 171-186, doi: 10.1007/3-540-45341-5_12.
 - [28] X. Xu, L. Zhu, Y. Liu, and M. Staples, "Resource-Oriented Architecture for Business Processes," *Proc. 15th Asia Pacific Software Engineering Conference (APSEC 2008)*, IEEE Computer Society, Dec. 2008, pp. 395-402, doi: 10.1109/APSEC.2008.52.
 - [29] S. Mosser, "Web Services Composition: Mashups Driven Orchestration Definition," *Proc. 2008 International Conference Computational Intelligence for Modeling Control & Automation*, IEEE Computer Society, Dec. 2008, pp. 284-289, doi: 10.1109/CIMCA.2008.96.
 - [30] Y. Xu, S. Tang, Y. Xu, and Z. Tang, "Towards Aspect Oriented Web Service Composition with UML," *Proc. 6th Int'l. Conf. Computer and Information Science (ICIS 2007)*, IEEE Computer Society, Jun. 2007, pp. 279-284, doi: 10.1109/ICIS.2007.185.
 - [31] J. Pathak, S. Basu, R. Lutz, and V. Honavar, "MoSCoE: A Framework for Modeling Web Service Composition and Execution," *Proc. 22nd International Conference on Data Engineering Workshops*, IEEE Computer Society, Apr. 2006, pp. x143, doi: 10.1109/ICDEW.2006.96.
 - [32] C. Pautasso, "RESTful Web Service Composition with BPEL for REST," *Data and Knowledge Engineering*, vol. 68, no. 9, Mar. 2009, pp. 851-866, doi: 10.1016/J.DATAK.2009.02.016.
 - [33] J. Mangler, E. Schikuta, and C. Witzany, "Quo Vadis Interface Definition Languages? Towards a Interface Definition Language for RESTful Services," *Proc. 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA '09)*, IEEE Computer Society, Dec. 2009, pp. 1-4, doi: 10.1109/SOCA.2009.5410459.
 - [34] M. zur Muehlen, J. V. Nickerson, and K. D. Swenson, "Developing Web Services Choreography Standards – the Case of REST vs. SOAP," *Decision Support Systems*, vol. 40, no. 1, July 2005, pp. 9-29, doi: 10.1016/j.dss.2004.04.008.

- [35] A. A. Lewis, "Web Services Description Language (WSDL) Version 2.0: Additional MEPs," W3C Working Group Note, June 2007, <http://www.w3.org/TR/wsd20-additional-meps/>.
- [36] S. Kona, A. Bansal, M. B. Blake, and G. Gupta, "Generalized Semantic-based Service Composition," Proc. IEEE 2008 International Conference on Web Services (ICWS'08), IEEE Computer Society, Sept. 2008, pp. 219-227, doi: 10.1109/ICWS.2008.118.
- [37] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition, vol. 5, no. 2, June 1993, pp. 199-220, doi: 10.1006/knac.1993.1008.
- [38] L. Liu and M. T. Özsu, Encyclopedia of Database Systems. New York: Springer, 2010.
- [39] T. Globerson, "Mental Capacity, Mental Effort, and Cognitive Style," Developmental Review, vol. 3, no. 3, Sept. 1983, pp. 292-302, doi: 10.1016/0273-2297(83)90017-5.
- [40] J. Cardoso, "How to Measure the Control-Flow Complexity of Web Processes and Workflows," Workflow Handbook 2005, Lighthouse Point: Layna Fischer, Apr. 2005, pp. 199-212.
- [41] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological Review, vol. 63, no. 2, Mar. 1956, pp. 81-97, doi: 10.1037/h0043158.
- [42] C. Francalanci and F. Merlo, "The Impact of Complexity on Software Design Quality and Costs: An Exploratory Empirical Analysis of Open Source Applications," Proc. 16th European Conference on Information Systems (ECIS 2008), June 2008, pp. 1442-1453, Galway, Ireland.
- [43] J. R. Turner and R. A. Cochrane, "Goals-and-Methods Matrix: Coping with Projects with Ill-defined Goals and/or Methods of Achieving them," International Journal of Project Management, vol. 11, no. 2, May 1993, pp. 93-102, doi: 10.1016/0263-7863(93)90017-H.
- [44] A. Camci and T. Kotnour, "Technology Complexity in Projects: Does Classical Project Management Work?," Proc. Technology Management for the Global Future (PICMET 2006), IEEE Computer Society, vol. 5, July 2006, pp. 2181-2186, doi: 10.1109/PICMET.2006.296806.
- [45] K. Dooley, "Organizational Complexity," International Encyclopedia of Business and Management, M. Warner (ed.), London: Thompson Learning, Oct. 2001, pp. 5013-5022.
- [46] N. M. Josuttis, SOA in Practice: The Art of Distributed System Design. Sebastopol: O'Reilly Media, Inc., 2007.
- [47] A. Barros, M. Dumas, and P. Oaks, "Standards for Web Service Choreography and Orchestration: Status and Perspectives," Proc. Business Process Management Workshops, Sept. 2005, pp. 61-74, doi: 10.1007/11678564_7.
- [48] K. M. Furulund and K. Moløkken-Østfold, "Increasing Software Effort Estimation Accuracy Using Experience Data, Estimation Models and Checklists," Proc. Seventh International Conference on Quality Software (QSIC '07), IEEE Computer Society, Oct. 2007, pp. 342-347, doi: 10.1109/QSIC.2007.4385518.
- [49] H. Demirkan, R. J. Kauffman, J. A. Vayghan, H. G. Fill, D. Karagiannis, and P. P. Maglio, "Service-Oriented Technology and Management: Perspectives on Research and Practice for the Coming Decade," Electronic Commerce Research and Applications, vol. 7, no. 4, Dec. 2008, pp. 356-376, doi: 10.1016/j.elerap.2008.07.002.
- [50] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Crawfordsville: Prentice Hall PTR, 2005.

APPENDIX I: A SAMPLE OF CLASSIFICATION MATRIX OF WEB SERVICE COMPOSITION

Technology		Context											
Category	Detailed Technique	Pattern		Semiotics		Mechanism		Design Time		Runtime			
		Orchestration	Choreography	Syntax	Semantics	SOAP	REST	Manual	Semi-Auto	Auto	Static	Dynamic	
Workflow-based	BPEL Programming	✓		✓		✓						✓	
	Semantic Matching [2]		✓		✓				✓			✓	
	eFlow [3]	✓		✓		✓							✓
	Bite [20]		✓	✓			✓					✓	
	SA-REST + Smashup [21]	✓			✓		✓					✓	
Model-driven	CSDL [27]	✓		✓		✓						✓	
	RESTfulBP [28]		✓	✓			✓					✓	
	UML + MDA [4]	✓		✓			✓					✓	
	UML + OCL [5]	✓			✓		✓						✓
	UML + QoS Support [6]	✓			✓		✓					✓	
AI planning	UML-WSC [7]	✓		✓			✓						✓
	UML + IHE framework [22]	✓		✓			✓						✓
	MD Mashup [29]		✓	✓			✓					✓	
	UML-AOWSC [30]	✓		✓			✓						✓
	MoSCoE [31]	✓			✓		✓					✓	
	SHOP2 [25]	✓			✓		✓					✓	
	Petri Net [23]		✓		✓		✓					✓	
	Situation Calculus [8]	✓			✓		✓					✓	
	I/O Automata [9] *		✓	✓		✓		✓				✓	
	Rule-based Planning [10]	✓			✓		✓					✓	
AI planning	Interface Automata [11]	✓			✓		✓					✓	
	Query Planning [12] *	✓		✓		✓						✓	
	Linear Logic Theorem Proving [13]	✓			✓		✓					✓	
	Colored Petri Net [14]	✓		✓		✓						✓	
	Model Checking [15]	✓			✓		✓					✓	
	AIMO [24]		✓		✓		✓					✓	
Situation Calculus for REST [26]	✓			✓		✓					✓		

* The approaches in [9] and [12] are independent of the Semiotics context.

APPENDIX II: EFFORT-ESTIMATION-CHECKLIST TABLE FOR WEB SERVICE COMPOSITION

Technology	Context													
	Category	Pattern		Semiotics			Mechanism			Design Time			Runtime	
		Orchestration	Choreography	Syntax	Semantics	SOAP	REST	Manual	Semi-Auto	Auto	Static	Dynamic		
		H3, H5 S(E _{Por})=2	H3, H5 S(E _{Chg})=4	H1, H5 S(E _{Sys})=2	H1, H5 S(E _{Fac})=3	H2, H5 S(E _{Soa})=4	H2, H5 S(E _{Res})=2	H1, H2, H5, H6 S(E _{Man})=14	H1, H2, H5, H6 S(E _{Ssa})=12	H1, H2, H5, H6 S(E _{Aut})=10	H1, H2, H5, H6 S(E _{Sta})=7	H1, H2, H5, H6 S(E _{Dyn})=5		
Workflow-based	H1, H2, H4, H5, H6 S(E _{Por})=16	H1, H2, H3, H4, H5, H6 S(E _{Por}) × S(E _{Chg}) = 32	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sys}) = 32	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Fac}) = 48	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Soa}) = 64	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Res}) = 32	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Man}) = 224	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Ssa}) = 192	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Aut}) = 160	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sta}) = 112	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Dyn}) = 80			
Model-driven	H1, H2, H4, H5, H6 S(E _{Por})=14	H1, H2, H3, H4, H5, H6 S(E _{Por}) × S(E _{Chg}) = 56	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sys}) = 28	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Fac}) = 42	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Soa}) = 56	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Res}) = 28	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Man}) = 196	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Ssa}) = 168	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Aut}) = 140	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sta}) = 98	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Dyn}) = 70			
AI planning	H1, H2, H4, H5, H6 S(E _{Por})=12	H1, H2, H3, H4, H5, H6 S(E _{Por}) × S(E _{Chg}) = 48	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sys}) = 24	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Fac}) = 36	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Soa}) = 48	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Res}) = 24	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Man}) = 168	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Ssa}) = 144	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Aut}) = 120	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Sta}) = 84	H1, H2, H4, H5, H6 S(E _{Por}) × S(E _{Dyn}) = 60			

Supporting Mobile Web Service Provisioning with Cloud Computing

Satish Narayana Srirama, Vladimir Šor and Eero Vainikko
Distributed Systems group
Institute of Computer Science, University of Tartu
J. Liivi 2, Tartu, Estonia
 {srirama, eero, volli}@ut.ee

Matthias Jarke
Information Systems and Databases Group
RWTH Aachen University
Ahornstr. 55, 52056 Aachen, Germany
 jarke@dbis.rwth-aachen.de

Abstract—Web services are going mobile. A Mobile Enterprise can be established in a cellular network by participating Mobile Hosts, which act as web service providers, and their clients. Mobile Hosts enable seamless integration of user-specific services to the enterprise, by following web service standards, also on the radio link and via resource constrained smart phones. However, establishing such a Mobile Enterprise poses several technical challenges, like the quality of service (QoS) and discovery aspects, for the network and as well as for mobile phone users. The paper summarizes the challenges and research in this domain, along with our developed mobile web service mediation framework (MWSMF). However, to scale Mobile Enterprise to the loads possible in cellular networks, we shifted some of its components to the new utility computing paradigm, cloud computing. The cloud based load balancing for the Mobile Enterprise can be provided at the middleware framework level or at the individual services level. This paper described both the approaches, with two Mobile Host application scenarios, in collaborative m-learning and multimedia services domains. The analysis concludes that MWSMF and its components are horizontally scalable, thus allowing to utilize elasticity of cloud platform to meet load requirements of Mobile Enterprise in an easy and quick manner.

Keywords-Mobile web services, Mobile Host, Mobile Enterprise, cloud computing, QoS and enterprise service bus

I. INTRODUCTION

* This journal paper is an extension to our prior work published at [1].

Mobile data services in tandem with web services [2] are seemingly the path breaking domain in current information systems research. In mobile web services domain, the resource constrained smart phones are used as both web service clients and providers (Mobile Host). Mobile terminals accessing the web services cater for anytime and anywhere access to services. Some interesting mobile web service applications are the provisioning of services like information search, language translation, company news etc. for employees who travel regularly. There are also many public web services like the weather forecast, stock quotes etc. accessible from smart phones. Mobile web service clients are also significant in the geospatial and location based services [3]. While mobile web service clients are common, the scope of mobile web service provisioning (MWSP) was studied at RWTH Aachen University since

2003 [4], where Mobile Hosts were developed, capable of providing basic web services from smart phones. Mobile web service clients and the Mobile Hosts in a cellular network, together form a Mobile Enterprise.

Mobile Hosts enable seamless integration of user-specific services to the enterprise, by following standard web service interfaces and standards also on the radio link. Moreover, services provided by the Mobile Host can be integrated with larger enterprise services bringing added value to these services. For example, services can be provided to the mobile user based on his up-to-date user context. Context details like device and network capabilities, location details etc. can be obtained from the mobile at runtime and can be used in providing most relevant services like maps specific to devices and location information. Besides, Mobile Hosts can collaborate among themselves in scenarios like Collaborative Journalism and Mobile Host Co-learn System and bring value to the enterprise. [5]

Once the Mobile Host was developed, an extensive performance analysis was conducted to prove its technical feasibility [4]. While service delivery and management from Mobile Host were thus shown technically feasible, the ability to provide proper quality of service (QoS), especially in terms of security and reasonable scalability, for the Mobile Host is observed to be very critical. Similarly, huge number of web services possible, with each Mobile Host providing some services in the wireless network, makes the discovery of these services quite complex. Proper QoS and discovery mechanisms are required for successful adoption of mobile web services into commercial environments. Moreover, the QoS and discovery analysis of mobile web services has raised the necessity for intermediary nodes helping in the integration of Mobile Hosts with the enterprise. Based on these requirements a Mobile Web Services Mediation Framework (MWSMF) [6] is designed as an intermediary between the web service clients and the Mobile Hosts within the Mobile Enterprise, using the Enterprise Service Bus (ESB) technology.

While we were successful in establishing MWSMF on standard servers, the scale of the Mobile Enterprise is leading us to the new utility computing paradigm, cloud computing. We also have observed that load balancing is

the key in successful deployment of Mobile Enterprise in commercial environments. So, we established the mediation framework on a public cloud infrastructure so that the framework can adapt itself to the loads of the mobile operator proprietary networks, thus mainly helping in horizontal scaling and load balancing the MWSMF and its components and consequently the Mobile Enterprise. The remaining sections of the paper are ordered as follows.

Section II discusses the details of providing services from smart phones. Section III discusses the challenges associated with establishing a Mobile Enterprise. Section IV discusses the details of the MWSMF. Section V discusses cloud computing and load handling issues of the MWSMF along with the analysis and results. Section VI discusses the details and approach of dragging Mobile Enterprise to the cloud with detailed analysis. Section VII concludes the paper with future research options.

II. MOBILE WEB SERVICE PROVISIONING

The quest for enabling open XML web service interfaces and standardized protocols also on the radio link, with the latest developments in cellular domain, lead to a new domain of applications, mobile web services. The developments in cellular world are two folded; firstly there is a significant improvement in device capabilities like better memory and processing power and secondly with the latest developments in mobile communication technologies with 3G and 4G technologies, higher data transmission rates in the order of few mbs were achieved. In the mobile web services domain, the resource constrained mobile devices are used as both web service clients and providers, still preserving the basic web services architecture in the wireless environments. While mobile web service clients are quite common these days [3], the research with providing web services from smart phones is still sparse.

The main benefit with Mobile Host is the achieved integration and interoperability for the mobile devices. It allows applications written in different languages and deployed on different platforms to communicate with Mobile Hosts over the cellular network. Moreover, the paradigm shift of smart phones from the role of service consumer to the service provider is a step toward practical realization of various computing paradigms such as pervasive computing, ubiquitous computing, ambient computing and context-aware computing. For example, the applications hosted on a mobile device provide information about the associated user (e.g. location, agenda) as well as the surrounding environment (e.g. signal strength, bandwidth). Mobile devices also support multiple integrated devices (e.g. camera) and auxiliary devices (e.g. Global Positioning Systems (GPS) receivers, printers). For the hosted services, they provide a gateway to make available their functionality to the outside world (e.g. providing paramedics assistance). In the absence of such provisioning functionality the mobile user has to regularly

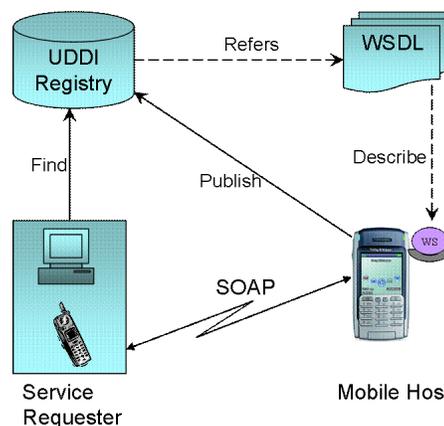


Figure 1. Basic mobile web services framework with the Mobile Host

update the contents to a standard server, with each update of the device's state.

With the intent, in our mobile web service provisioning project one such Mobile Host [4] was developed, proving the feasibility of concept. Figure 1 shows the basic mobile web services framework with web services being provided from the Mobile Host. Mobile Host is a light weight web service provider built for resource constrained devices like cellular phones. It has been developed as a web service handler built on top of a normal Web server. The SOAP based web service requests sent by HTTP tunneling are diverted and handled by the web service handler component. The Mobile Host was developed in PersonalJava on a SonyEricsson P800 smart phone. The footprint of the fully functional prototype is only 130 KB. Open source kSOAP2 [7] was used for creating and handling the SOAP messages. The key challenges addressed in Mobile Host's development are threefold: to keep the Mobile Host fully compatible with the usual web service interfaces such that clients will not notice the difference; to design the Mobile Host with a very small footprint that is acceptable in the smart phone world; and to limit the performance overhead of the web service functionality such that neither the services themselves nor the normal functioning of the smart phone for the user is seriously impeded.

The detailed performance evaluation of this Mobile Host clearly showed that service delivery as well as service administration can be done with reasonable ergonomic quality by normal mobile phone users. As the most important result, it turns out that the total web service processing time at the Mobile Host is only a small fraction of the total request-response time (< 10%) and rest all being transmission delay. This makes the performance of the Mobile Host directly proportional to achievable higher data transmission rates. Further, the regression analysis of the Mobile Host showed that the Mobile Host can handle up to 8 concurrent requests for reasonable services of message sizes approximately 2

Kb. Mobile Host is also possible with other Java variants like Java 2 Micro Edition (J2ME) [8], for smart phones. We also have developed a J2ME based Mobile Host and its performance was observed to be not so significantly different from that of the PersonalJava version.

Mobile Host opens up a new set of applications and it finds its use in several domains like mobile community support, collaborative learning, social systems etc. Primarily, the smart phone can act as a multi-user device without additional manual effort on part of the mobile carrier. Several applications were developed and demonstrated with the Mobile Host, for example in a remote patient tele-monitoring scenario, the Mobile Host can collect remote patient's vital signs like blood pressure, heart rate, temperature etc. from different sensors and provide them to the doctors in real time. In the absence of such Mobile Host the details are to be regularly updated to a server, where from the doctor can access the details. The latter scenario causes problems with stale details and increased network loads. A second example is that in case of a distress call; the mobile terminal can provide a geographical description of its location (as pictures) along with location details. Another interesting application scenario involves the smooth co-ordination between journalists and their respective organizations while covering events like Olympics. [5]

III. MOBILE ENTERPRISE

A Mobile Enterprise [5], [9] can be established in a cellular network by participating Mobile Hosts and their clients, where the hosts provide user-specific services to the clients as per the WS* standards. However, such a Mobile Enterprise established, poses many technical challenges, both to the service providers and to the mobile operator. Some of the critical challenges and associated research are addressed in this section.

A. Challenges for establishing Mobile Enterprise

Figure 2 shows the Mobile Enterprise and hints the critical challenges posed to the mobile phone users and the operators. As the Mobile Host provides services to the Internet, devices should be safe from malicious attacks. For this, the Mobile Host has to provide only secure and reliable communication in the vulnerable and volatile mobile ad-hoc topologies. In terms of scalability, the Mobile Host has to process reasonable number of clients, over long durations, without failure and without seriously impeding normal functioning of the smart phone for the user.

Similarly, huge number of available web services, with each Mobile Host providing some services in the wireless network, makes the discovery of the most relevant services quite complex. Proper discovery mechanisms are required for successful adoption of Mobile Enterprise. The discovery, moreover, poses some critical questions like: where to publish the services provided by the Mobile Hosts? Should

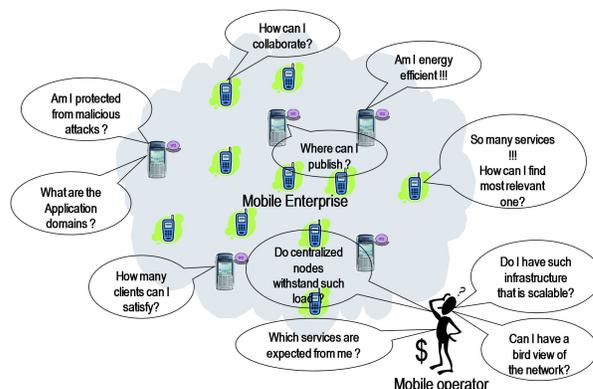


Figure 2. Mobile Enterprise and the critical challenges posed to the mobile phone users and the operator

they be published with the centralized Universal Description, Discovery, and Integration (UDDI) registries available in the Internet or the operator is going to offer some help? This also raises questions like whether centralized nodes can withstand such high loads or some alternatives are to be looked at?

From the mobile operator's perspective the Mobile Enterprise poses questions like: what are the services expected by the mobile users from the operator? Can the operator monitor the communication and have a bird view of the complete network, so that business scenarios can be drawn out of it? Do operators have such infrastructure that can scale and adapt to such huge oscillating requirements? What about the scalability of such infrastructure?

Our research in this domain focused on addressing most of these issues [5] and the remaining parts of this paper summarize the research and results.

B. QoS aspects of the Mobile Host

Providing proper QoS, especially, appropriate security and reasonable scalability, for mobile web service provisioning domain was observed to be very critical. The security analysis of the Mobile Host studied the adaptability of WS-Security specification to the MWSP domain and concludes that not all of the specification can be applied to the Mobile Host, mainly because of resource limitations. The results of our analysis suggest that the mobile web service messages of reasonable size, approximately 2-5kb, can be secured with web service security standard specifications. The security delays caused are approximately 3-5 seconds. We could also conclude from the analysis that the best way of securing messages in a Mobile Enterprise is to use AES (Advanced Encryption Standard) symmetric encryption with 256 bit key, and to exchange the keys with RSA 1024 bit asymmetric key exchange mechanism and signing the messages with RSAwithSHA1. But there are still high performance penalties when messages are both encrypted and signed. So we suggest encrypting only the parts of the message, which are critical in terms of security and signing the message. The signing on

top of the encryption can completely be avoided in specific applications with lower security requirements. [10]

In terms of scalability, the layered model of web service communication, introduces a lot of message overhead to the exchanged verbose XML based SOAP messages. This consumes a lot of resources, since all this additional information has to be exchanged over the radio link. Thus for improving scalability the messages are to be compressed without effecting the interoperability of the mobile web services. Message compression also improves the energy efficiency of the devices as there will be less data to transmit.

In the scalability analysis of the Most Host [11], we have adapted BinXML [12] for compressing the mobile web service messages. BinXML is a light-weight XML compression mechanism, which replaces each XML tag and attribute with a unique byte value and replaces each end tag with 0xFF. By using a state machine and 6 special byte values including 0xFF, any XML data with circa 245 tags can be represented in this format. The approach is specifically designed to target SOAP messages across radio links. So the mobile web service messages are exchanged in the BinXML format, and this has reduced the message of some of the services by 30%, drastically reducing the transmission delays of mobile web service invocation. The BinXML compression ratio is very significant where the SOAP message has repeated tags and deep structure. The binary encoding is also significant for the security analysis as there was a linear increase in the size of the message with the security incorporation. The variation in the WS-Security encrypted message size for a typical 5 Kb message is approximately 50%. [5]

C. Discovery aspects of the Mobile Enterprise

In a commercial Mobile Enterprise with Mobile Hosts, and with each Mobile Host providing some services for the Internet, expected number of services to be published could be quite high. Generally web services are published by advertising WSDL (Web Services Description Language) descriptions in a UDDI registry. But with huge number of services possible with Mobile Hosts, a centralized solution is not the best idea, as they can have bottlenecks and can introduce single points of failure. Besides, mobile networks are quite dynamic due to the node movement. Devices can join or leave network at any time and can switch from one operator to another operator. This makes the binding information in the WSDL documents, inappropriate. Hence the services are to be republished every time the Mobile Host changes the network.

Dynamic service discovery is one of the most extensively explored research topics in the recent times. Most of these service discovery protocols are based on the announce-listen model like in Jini. In this model periodic multicast mechanism is used for service announcement and discovery. But these mechanisms assume a service proxy object that acts as

the registry and it is always available. For dynamic ad hoc networks, assuming the existence of devices that are stable and powerful enough to play the role of the central service registries is inappropriate. Hence services distributed in the ad-hoc networks must be discovered without a centralized registry and should be able to support spontaneous peer to peer (P2P) connectivity. [13] proposes a distributed peer to peer Web service registry solution based on lightweight Web service profiles. They have developed VISR (View based Integration of Web Service Registries) as a peer to peer architecture for distributed Web service registry. Similarly Konark service discovery protocol [14] was designed for discovery and delivery of device independent services in ad hoc networks.

Considering these developments and our need for distributed registry and dynamic discovery, we have studied alternative means of mobile web service discovery and realized a discovery mechanism in the P2P network. In this solution, the virtual P2P network also called the mobile P2P network is established in the mobile operator network with one of the nodes in operator proprietary network, acting as a JXTA super peer. JXTA (Juxtapose) is an open source P2P protocol specification. Once the virtual P2P network is established, the services deployed on Mobile Host in the JXME virtual P2P network are to be published as JXTA advertisements, so that they can be sensed as JXTA services among other peers. JXTA specifies Modules as a generic abstraction that allows peers to describe and instantiate any type of implementation of behavior representing any piece of "code" in the JXTA world. So the mobile web services are published as JXTA modules in the virtual P2P network. Once published to the mobile P2P network, the services can later be discovered by using the keyword based search provided by JXTA. This approach also considered categorizing the services and the advanced features like context aware service discovery. We address the discovery solution as mobile P2P discovery mechanism. The evaluation of the discovery approach suggested that the smart phones are successful in identifying the services in the P2P network, with reasonable performance penalties for the Mobile Host. [15]

IV. MOBILE WEB SERVICES MEDIATION FRAMEWORK

Mobile Hosts with proper QoS and discovery mechanisms, enable seamless integration of user-specific services to the Mobile Enterprise. Moreover services provided by the Mobile Host can be integrated with larger enterprise services bringing added value to these services. However, enterprise networks deploy disparate applications, platforms, and business processes that need to communicate or exchange data with each other or in this specific scenario addressed by the paper, with the Mobile Hosts. The applications, platforms and processes of enterprise networks generally have non-compatible data formats and non-compatible communications protocols. Besides, within the domain of our

research, the QoS and discovery study of the Mobile Host offered solutions in disparate technologies like JXTA. This leads to serious integration problems within the networks. The integration problem extends further if two or more of such enterprise networks have to communicate among themselves. We generally address this research scope and domain, as the Enterprise Service Integration.

The mobile web services mediation framework (MWSMF) [6] is established as an intermediary between the web service clients and the Mobile Hosts in mobile enterprise. ESB is used as the background technology in realizing the mediation framework. Similar mediation mechanisms for mobile web services are addressed in [16]. Especially, [16] describes the status of research with provisioning services from resource constrained devices. When considering mediation within semantic web services, Web Service Modeling Ontology (WSMO) has significant contributions [17]. However, we went with the ESB approach, due to the availability of several open source implementations.

Figure 3 shows the Mobile Enterprise and the basic components of the mediation framework. For realizing the mediation framework we relied on ServiceMix [18], an open source implementation of ESB, based on the JBI specification [19]. JBI architecture supports two types of components Service Engines and Binding Components. Service engines are components responsible for implementing business logic and they can be service providers/consumers. Service engine components support content-based routing, orchestration, rules, data transformations etc. Service engines communicate with the system by exchanging normalized messages across the normalized message router (NMR). The normalized messaging model is based on WSDL specification. The service engine components are shown as straight lined rectangles in the figure. Binding components are used to send and receive messages across specific protocols and transports. The binding components marshal and unmarshal messages to and from protocol-specific data formats to normalized messages. The binding components are shown as dashed rectangles in the Figure 3.

The HttpReceiver component shown in figure 3 receives the web service requests (SOAP over HTTP) over a specific port and forward them to the Broker component via NMR. The main integration logic of the mediation framework is maintained at the Broker component. For example, in case of the scalability maintenance, the messages received by Broker are verified for mobile web service messages. If the messages are normal Http requests, they are handled by the HttpInvoker binding component. If they comprise mobile web service messages, the Broker component further ensures the QoS of the mobile web service messages and transforms them as and when necessary, using the QoSVerifier service engine component, and routes the messages, based on their content, to the respective Mobile Hosts. The framework also

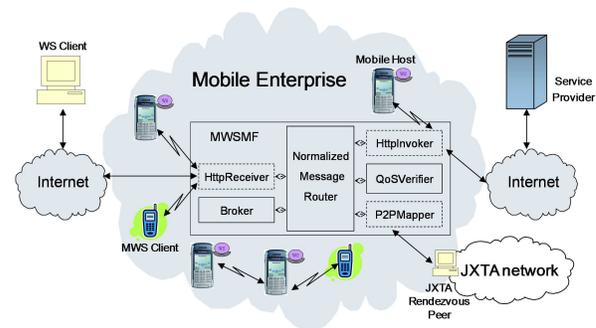


Figure 3. Mobile Enterprise setup with Mobile Hosts, MWS clients and MWSMF

ensures that once the mobile P2P network is established, the web service clients can discover the services using mobile P2P discovery mechanism and can access deployed services across MWSMF and JXTA network. [6]

Apart from security and improvements to the scalability, QoS provisioning features of the MWSMF also include message persistence, guaranteed delivery, failure handling and transaction support. External web service clients, that do not participate in the mobile P2P network, can also directly access the services deployed on the Mobile Hosts via MWSMF, as long as the web services are published with any public UDDI registry or the registry deployed at the mediation framework and the Mobile Hosts are provided with public IPs. This approach evades the JXME network completely. Thus the mediation framework acts as an external gateway from Internet to the Mobile Hosts and mobile P2P network. The framework also provides a bird view of the mobile enterprise to the cellular operator, so that business scenarios can be drawn out of it. Preliminary analysis of the mediation framework is available at [5].

V. MWSMF ON THE CLOUD

While the MWSMF was successful in achieving the integrational requirements of the Mobile Host and the Mobile Enterprise, a standalone framework again faces the troubles with heavy loads. The problems with scalability are quite relevant in such scenarios and the system should scale on demand. For example number of Mobile Hosts providing the services and the number of services provided by the Mobile Hosts can explode while some events are underway; like Olympics or national elections etc. Some of these application scenarios are addressed in [5]. This increases the number of MWS clients the framework has to support. Elasticity of the framework can be defined as its ability to adjust according to the varying number of requests, it has to support. As the study targets the scales of mobile operator proprietary networks, to achieve elasticity, horizontal scaling (scaling by adding more nodes to the cluster, rather than increasing performance of a single node) and load balancing for the

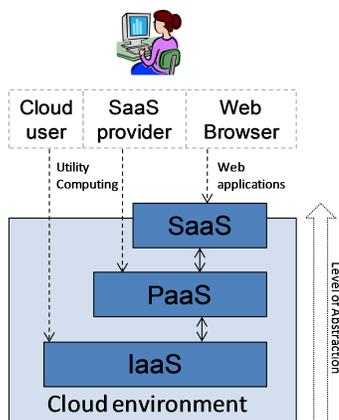


Figure 4. Level of abstraction and layers of cloud services

MWSMF, we tried to realize the mediation framework on a public cloud.

A. Cloud computing

Cloud computing is a style of computing in which, typically, resources scalable on demand are provided "as a service (aaS)" over the Internet to users who need not have knowledge of, expertise in, or control over the cloud infrastructure that supports them. Cloud computing mainly forwards the idea of utility computing along with virtualization. In the utility computing model, consumers pay based on their usage of computing resources, just like the traditional utility services e.g. water, electricity, gas etc. Just like any utility services model cloud computing benefits from economies of scale. On the other hand, virtualization technologies partition hardware and thus provide flexible and scalable computing platforms. Servers in the cloud can be physical machines or virtual machines. A cloud computing platform dynamically provisions, configures, reconfigures, and de-provisions servers as requested. [20], [21]

Cloud services are provided on demand and at different levels. Figure 4 shows the layers of cloud services, in terms of level of abstraction. The provisioning of services can be at the Infrastructural level (IaaS) or Platform level (PaaS) or at the Software level (SaaS). In the IaaS, commodity computers, distributed across Internet, are used to perform parallel processing, distributed storage, indexing and mining of data. IaaS provides complete control over the operating system and the clients benefit from the computing resources like processing power and storage, e.g. Amazon EC2 [22]. Virtualization is the key technology behind realization of these services. PaaS mainly provides hosting environments for other applications. Clients can deploy the domain specific applications on these platforms, e.g. Google App Engine [23]. These applications are in turn provided to the users as SaaS. SaaS are generally accessible from web browsers, e.g. Facebook. Web 2.0 is the main technology behind the

realization of SaaS. However, the abstraction between the layers is not concrete and several of the examples can be argued for other layers.

While there are several public clouds on the market, Google Apps (Google Mail, Docs, Sites, Calendar, etc), Google App Engine (provides elastic platform for Java and Python applications with some limitations) and Amazon EC2 are probably most known and widely used. Elastic Java Virtual Machine on Google App Engine allows developers to concentrate on creating functionality rather than bother about maintenance and system setup. Such sandboxing, however, places some restrictions on the allowed functionality [23]. Amazon EC2 on the other hand allows full control over virtual machine, starting from the operating system. It is possible to select a suitable operating system, and platform (32 and 64 bit) from many available Amazon Machine Images (AMI) and several possible virtual machines, which differ in CPU power, memory and disk space. This functionality allows to freely select suitable technologies for any particular task. In case of EC2, price for the service depends on machine size, its uptime, and used bandwidth in and out of the cloud. Flexibility of EC2 environment and our existing Mobile Enterprise implementation were some of the reasons why EC2 was chosen for most of our experiments.

Moreover, there are free implementations of EC2 compatible cloud infrastructure e.g. Eucalyptus [24], that help in creating private clouds. Thus the cloud computing applications can initially be developed at the private clouds and later can be scaled to the public clouds. The setup is of great help for the research and academic communities, as the initial expenses of experiments can be reduced by great extent. Our research group is in the process of setting up a scientific computing cloud (SciCloud) on our clusters, using Eucalyptus technology. With this SciCloud [25], students and researchers can efficiently use the already existing resources of university computer networks, in solving computationally intensive scientific, mathematical, and academic problems. The project mainly targets the development of a framework, including models and methods for establishment, proper selection, state management (managing running state and data), auto scaling and interoperability of the private clouds. The preliminary results can be found at the project site [26] and will be addressed by our future publications.

B. Load balancing the MWSMF from the cloud

To achieve the scalability for the mediation framework, the MWSMF was installed on the Amazon EC2 cloud. Once the Amazon Machine Images (AMI) are configured, stateless nature of the MWSMF allows, fairly easy horizontal scaling by adding more MWSMF nodes and distributing the load among them with the load balancer. Figure 5 shows the deployment scenario with the load balancer (LB) and the MWSMF worker AMI nodes (W-1, W-2, W-3, and W-n).

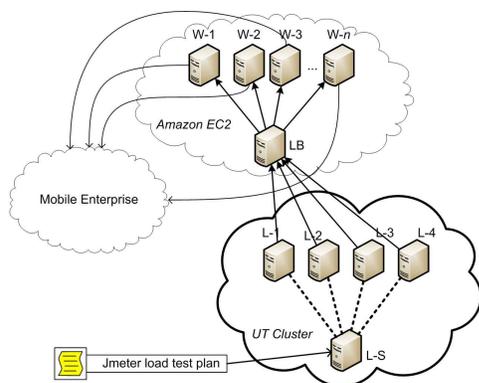


Figure 5. Load test setup for the MWSMF

There are several load balancing techniques that can be used in this scenario. One approach is to use DNS based load balancing, where each call to the DNS server will result in different IP address. This means that each MWSMF node will be accessed by certain subset of clients directly, without an intermediary load balancing proxy as discussed below. This approach is not fault tolerant in case the framework node would crash but its IP would be cached on the client's DNS cache. However, this approach is inevitable, if loads on the single proxy based load balancer will grow to a level that a single load balancer itself will become a bottleneck. Another approach is to use load balancing proxy server in front of MWSMF nodes. Among other options, Apache HTTPD server with `mod_proxy` and `mod_load_balancer` is probably most commonly used configuration. It has one major drawback in elastic environment, as it doesn't allow dynamic reconfiguration of worker nodes. If we add or remove some MWSMF nodes we are required to restart load balancer as well, which is not convenient and potentially introduces some failed requests during restart.

Alternative http proxy load balancer HAProxy [27] allows such dynamic behavior. However we used Apache HTTP server with `mod_proxy` and `mod_load_balancer` [28] as a load balancer for the MWSMF because it is more widespread and we had experience in configuring such setup, which was important, as the aim of this research was to show the horizontal scalability rather than achieve maximum automation. However we have considered HAProxy [27] in the analysis of scaling the Mobile Enterprise in total. The scenario is explained in the next section with a usage scenario.

C. Scalability of the MWSMF

Load testing of MWSMF on the cloud was performed in a distributed manner using JMeter - open source load testing software. Figure 5 shows the deployment setup in detail. JMeter was deployed on one of the clusters in the University of Tartu (UT Cluster). Deployment consisted of

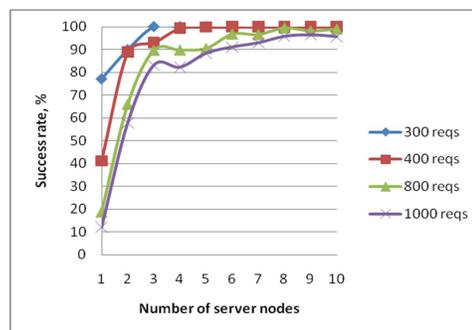


Figure 6. Success rate of concurrent requests over multiple server nodes

4 slave nodes (L-1, L-2, L-3, and L-4) and 1 master node (L-S). Load testing scenario (called a test plan in Jmeter) is loaded on the master node, which sends it to the slave nodes and initiates load testing. During the test run each slave node sends testing results back to the master node, where results are aggregated into a single report.

In our test scenario we performed 5 consequent requests by n concurrent threads, where n varied between 75 and 250 per slave node, which makes 300 to 1000 concurrent requests on a load balancer, thus simulating a large number of simultaneous clients for the MWSMF and the Mobile Host in Mobile Enterprise. Another important factor that impacts test results is a connection and response timeout on the client, in our test case - the slave node. Connection timeout is a time until connection to the server is established and response timeout is the time since call starts on the client side until response is received. If these timeouts are long enough, then observations showed, that even single MWSMF node can withstand large loads, due to the sufficient QoS of the ESB. However, in such scenario a call may last too long for a mobile client and the client may start retransmitting instead of waiting. In our tests we set connection-response timeout to 50-70 seconds. It must be also noted that, in the real life connection timeout on a client side is not a parameter that the service provider can affect nor predict. In case of interactive applications, where user interaction is involved, response should be preferably delivered in less than 10 seconds to keep user's attention [29].

On the cloud front a load balancer (LB) and up to 10 MWSMF worker nodes were set up. To show the elasticity of the cloud we increased the number of the server nodes from 1 to 10 after each test. All servers were running on Amazon EC2 infrastructure and all of them were using EC2 small instances. Small instance has 1.7 GB of memory, CPU power of 1 EC2 Compute unit, which is equivalent to CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor as of 07.12.2009 (CPU capacity of an EC2 compute unit do change in time). Both load balancer and worker nodes were running 32 bit Linux platforms. Apache HTTP server version 2.2.8 with `mod_proxy` and `mod_load_balancer` were

used as a load balancer. Load balancer was setup to use request based scheduling, which means that all worker nodes received equal amount of requests. However, it is possible to configure load balancer based on traffic or busyness. Busyness means how many concurrent requests a worker node has at the moment the new request arrives. In real-life situation best load balancing algorithm for a particular scenario should be chosen based on the services provided by the mobile enterprise and the nature of the request/response traffic. For more details on load balancing algorithm refer [28].

In the load test of the MWSMF, we measured how success rate of the requests depends on a number of worker nodes depending on a number of concurrent requests. Success means that a request will get a response before connection or response timeout occurs and success rate shows how many requests from all performed requests succeeded. The results of the experiment are shown in figure 6. From the diagram it can be clearly seen that the percentage of succeeded requests grows logarithmically with the number of nodes and degrades exponentially as load grows. Performance of a single node drops rapidly already after 300 concurrent requests and even with 300 concurrent requests success rate is only 77%, however 3 nodes can handle this load with 100% success rate. It can be also seen, that with current setup adding more nodes does not show any visible effect after 6 nodes and performance is improved by an insignificant fraction in contrast to difference between 1, 2 and 3 nodes.

In summary we observed that, with current MWSMF implementation one single node can handle around 100-130 concurrent MWS requests with 100% success rate. Adding an additional node adds roughly 100 new concurrent requests to the total capability until the load grows up to 800 concurrent requests, when load balancer itself becomes a bottleneck and adding any additional nodes do not give desired effect. This analysis showed mediation framework to be horizontally scalable. However, certain loads demand more advanced load balancing techniques. The elastic cloud environment helps to achieve this required setup very quickly.

VI. SCALING MOBILE ENTERPRISE

While our earlier analysis proved that MWSMF is horizontally scalable, scaling the Mobile Enterprise in total is a different issue. Our earlier analysis only considered the load balancing ahead of the MWSMF itself. However, individual services also can become a bottleneck and thus the load on them has to be balanced. So the load balancing for the Mobile Enterprise as a whole has to be extended further. To sum it up – cloud based load balancing for the Mobile Enterprise can be at the mediation framework level or at the individual service level. We try to address both the approaches in terms of two application scenarios,

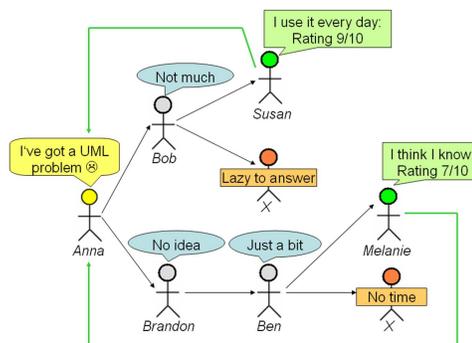


Figure 7. Expert finder scenario with the MobileHost CoLearn System

MobileHost CoLearn System and lightweight application server (LAS) services for mobiles.

A. MobileHost CoLearn System

The MobileHost CoLearn System studied the scope of Mobile Host in m-learning (mobile learning) domain [30]. The system presents a novel approach to expert finding within a truly mobile collaborative learning environment. It targets the framework of the learner's social network, along with the social networks of her acquaintances, and the social networks of the acquaintances of her acquaintances, and so forth. Such an expert finder flow usually leads to the discovery of more than one potential expert, and the learner's subjective decision who of them is the most knowledgeable one can be based either on the rating for the expert's level of expertise in the field, or on the path that the expert finder request has traveled before reaching the respective expert. An example scenario, using the system, is illustrated in the figure 7. In the expert finder scenario of the system, every participant should act as both provider and client for the messages being exchanged, which the Mobile Host technology has made feasible. After having found an expert, the learner is provided with all the necessary information in order to contact that expert for further assistance regarding specific issues.

Alongside the valuable knowledge that flows within the system from the experts to non-experienced learners, the system supports the retrieval of a variety of literature resources, such as articles, proceedings, pictures, audio or video lecture recordings, location details, and other learning services. Most often the resources are tagged by the learners. As tagging is something subjective, a three-level scale of *relevance of a tag to a resource* has been introduced. The system also has support for image and audio resources within photocasting and podcasting channels. The channels automatically distribute resources to all subscribers, as soon as they become available. MobileHost CoLearn system is the first of its kind that adapts mobile web services for collaborative learning, bringing the benefits of the latest technological developments to the learner. [30]

B. Scalable MobileHost CoLearn System

In the MobileHost CoLearn System, the main load for the Mobile Enterprise was at handling large number of clients and at providing the QoS services from the MWSMF. For example, MWSMF has to convert the incoming XML based messages to BinXML format so that the messages can be exchanged across the radio link. The process is taken care by the QoSVerifier component of the MWSMF (figure 8). So to increase the elasticity for the Mobile Enterprise we can establish a load balancer in front of the MWSMF running on several nodes in the public cloud, handling the mobile clients by accessing services from several Mobile Hosts. This is what was showed by our earlier analysis.

The next subsections discuss scaling the Mobile Enterprise with respect to load balancing at the individual services level.

C. Mobile access to LAS services

LAS is a lightweight application server (LAS) designed as a community middleware that is capable of managing users and multiple hierarchically structured communities along with their particular access rights as well as a set of services accessible to users. LAS mainly offers MPEG-7 (Moving Picture Experts Group) multimedia services to the users. MPEG-7 is a well-established and widely used standard in multimedia data management. However, due to its inherent complexity it was not used in mobile data management that often. With new initiatives like the application profiles the use of MPEG-7 has become much easier, also for mobile data management. A community application can make use of the offered services by simply connecting to the server and then remotely invoking service methods. Server functionality of the LAS is easily extensible by implementing and plugging in new services and respective components. Many community information systems have been built on top of this framework including MIST; a MPEG-7 based non-linear digital storytelling system, ACIS; a multimedia information system, and CAAS; a mobile application for context-aware search and retrieval of multimedia and community members. [31]

Even though, LAS is a reliable application server, it is not a pure web service architecture; it was not designed under the SOA paradigm and important aspects like scalability and distributed services were not taken into account. QoS and performance problems have been observed recently by LAS users. For many years, LAS has been used on top of traditional networks infrastructures for providing the services required by social software such as Virtual Campfire [32]. Virtual Campfire, is an advanced framework to create, search, and share multimedia artifacts with context awareness across communities.

Recently, the multimedia services are also being offered to the Mobile Hosts and mobile phone users. The multimedia services can be accessed from mobile phones in three modes:

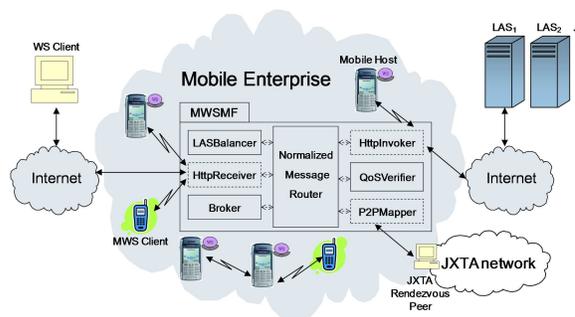


Figure 8. Mobile Enterprise setup with the LASBalancer and LAS server farm coming into the picture

- 1) Directly accessing the MPEG services through the Mobile Host.
- 2) Accessing the service through the mediation framework of the mobile enterprise.
- 3) Indirect way of using the Mobile Host to connect to the mediation framework.

Generally LAS services are extended and are provided as services from the Mobile Host. The extensions can be user specific. This entry of the Mobile Hosts into the LAS has further advanced the scalability problem. Load balancing and cluster support were observed to be the immediate requirements for improving the performance of the LAS.

D. Load balancing mobile access to LAS services

Contrary to the MobileHost CoLearn System, the QoS of the LAS can be improved either by changing the architecture of the LAS to have the cluster support or to employ a binding component on the MWSMF, taking care of the load balancing issues. In the first case, a hardware based load balancer can be deployed through specialized devices, like multilayer switches [33]. However, implementing, configuring and maintaining this solution is costly in terms of money and time.

Alternatively, we can deploy the LAS servers on the cloud and employ the same load balancer technique as in the case of the first scenario. As a third solution, we deployed the HAProxy node in the cloud and the requests are diverted to the respective LAS. If the load further increases, new LAS nodes can be deployed on the cloud. The main difference is that HAProxy allows dynamic behavior to the architecture and new LAS nodes can be added dynamically to the setup. This solution utilizes the elasticity, dynamic and on-demand provisioning features of the cloud, to the most. We are also studying the auto scaling of the cloud, as part of our SciCloud project. With this solution, the load balancing system can react to the sudden surges in usage patterns and can provision new nodes dynamically. The details will be addressed by our future publications.

For the third option, employing a binding component on the MWSMF, we adapted our knowledge from Mobile

Enterprise domain to the LAS. Moreover, since LAS services are also accessible to Mobile Host, we wanted to provide only a single entry to the LAS from the Mobile Enterprise. We developed components that provide web service interface to the LAS services. These integration components with the load balancer in front of them are designed to act as a cluster so that the requests are diverted to the less occupied server among a set of LASs. Connection to the LAS cluster is handled by the LASBalancer component at MWSMF. Modified Mobile Enterprise with the entry of the LASBalancer into the picture is shown in figure 8. You can see this component being present inside the MWSMF in the diagram. Inside LAS there are no necessary changes to do. Mobile users of LAS only need to connect to a single point, the MWSMF, in order to access any LAS server they are interested in. Without this solution, Mobile Hosts should have specific connection to the right LAS server based on the services offered by it. However, this architecture adds extra load to the mediation framework at LASBalancer level. QoS and fault tolerance features of ESB help to some extent, in handling this load. But, LAS requests don't need QoS transformation features of the MWSMF as the messages are sent via Internet. So the node that provided load balancing and web service interface for the LAS, is separated from the MWSMF, and we deployed it on the EC2 cloud. The HttpInvoker just diverts the LAS requests to this node. Now this node is horizontally scalable and we can apply business logic, fault tolerance and solution correctness to the cloud node without seriously affecting the performance of MWSMF. The results of the analysis are summarized in next subsection.

E. Testing the scalability of the Mobile Enterprise

In previous subsections we outlined requirements for scalability of the Mobile Enterprise and described the solution to integrate LAS and MWSMF in a scalable way. To verify our ideas, an experiment was conducted using Amazon EC2 services to scale the number of servers up and down. Deployment was made similarly to the MWSMF scalability experiment described in section V – we used Amazon EC2 public cloud infrastructure to deploy the Mobile Enterprise and the HPC (*High Performance Computing*) cluster of the University of Tartu to deploy JMeter in a distributed manner for load generation and measurement of the results. Deployment diagram is shown in Figure 9.

As we had limited access to the LAS installation, we substituted it with a mock application server. The server provides a web service that on request performs some image manipulation and sends the resulting image back to the client. As we are concerned only with the performance and scalability of the Mobile Enterprise, the functioning of the service and the application server do not affect the analysis and results. Servers hosting this image web service are shown as IS-1 ... IS-n on the figure 9. These servers

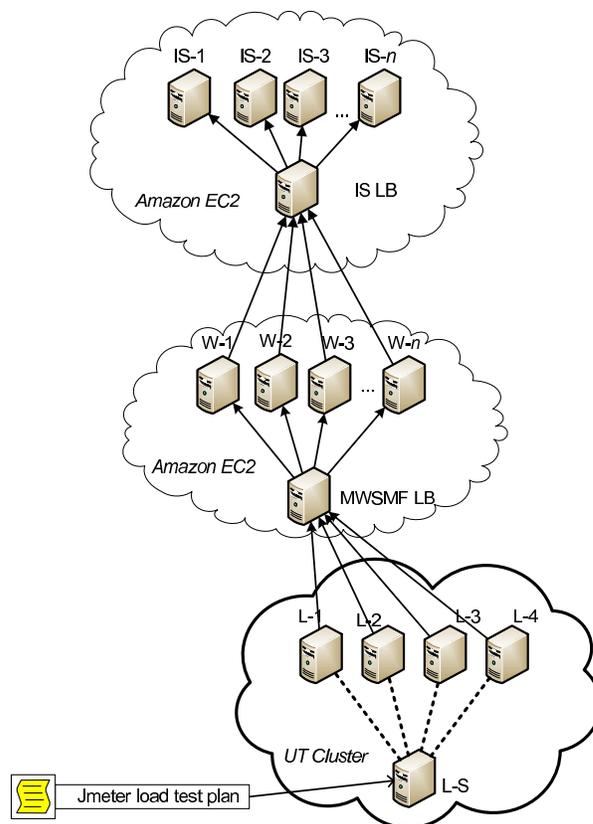


Figure 9. Deployment diagram of the scaling experiment.

constitute the cluster and the IS LB is the load balancer for it. HAProxy was used as a load balancer for this cluster.

MWSMF load balancing setup is the same used in the MWSMF scaling experiment – *MWSMF LB* is the load balancer and W-1 ... W-n are mediation framework's nodes. However, this time we used HAProxy instead of the Apache with *mod_proxy* to have a consistent deployment with image cluster and to compare it with the setup from the previous experiment [1]. HAProxy showed itself more suitable for such dynamic setup because it allows easily specify the configuration file location as a command line parameter, which is a lot more convenient when lot of changes have to be made (each time a cluster was changed, configuration file had to be changed). Also, HAProxy comes with a dynamic dashboard containing extensive statistics which show a lot more information compared to the default statistics web page of *mod_proxy*.

This time we used 5 Apache JMeter servers instead of 4 to generate the load of 200, 400, 600, 800 and 1000 concurrent requests (which makes respectively 40, 80, 120, 160 and 200 concurrent requests per one JMeter server). In the experiment we varied the number of MWSMF and Image Server nodes in the respective clusters and tested the setup with aforementioned loads. As we concluded from

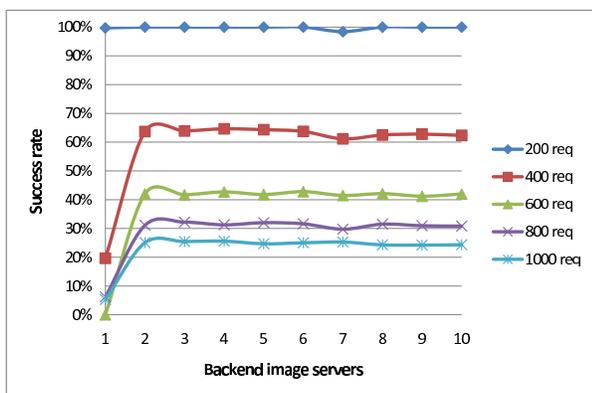


Figure 10. Scaling Image servers with 1 MWSMF node.

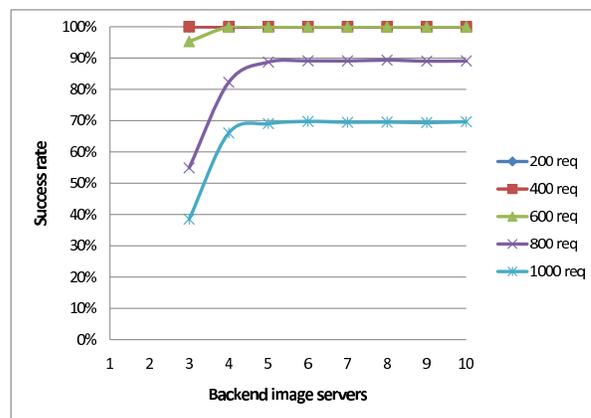


Figure 12. Scaling Image servers with 3 balanced MWSMF nodes.

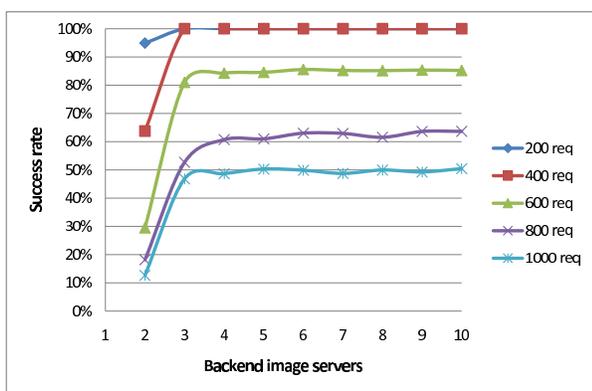


Figure 11. Scaling Image servers with 2 balanced MWSMF nodes.

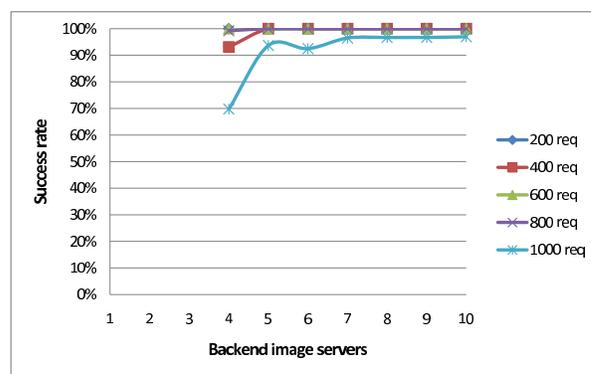


Figure 13. Scaling Image servers with 4 balanced MWSMF nodes.

the previous experiment, scaling of MWSMF makes some real impact only when scaled up to 4 nodes, after which difference in adding more servers is less visible. So this time we changed number of MWSMF nodes in the cluster from 1 to 4.

Number of servers providing Image Service was varied between 1 and 10. It must be noted, that the number of Image Service servers was always equal or bigger than the number of MWSMF nodes. The reasoning for this is an assumption, that when we model a contention of a particular service, then it shall be upscaled before mediation framework. This means that for this scenario the number of nodes for a particular service will always be bigger or equal to the number of mediation framework nodes.

Figures 10, 11, 12 and 13 summarize results of experiments. It can be seen, that increasing the number of servers for particular service results in the success rate growth and the tendency is closely similar to the characteristics observed during previous MWSMF scaling – large increase with first 3 nodes and after that difference is almost unnoticeable. Adding additional MWSMF nodes also adds to the success factor growth – it acts as a multiplier for the whole graph. This, however, also shows that mediation framework is a

major factor for the scalability of the Mobile Enterprise as a whole.

VII. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The developments in the web services domain, the improved device capabilities of the smart phones and the improved transmission capabilities of the cellular networks have lead to the mobile web services provisioning domain. With this paper, we summarized the challenges and research associated in this domain and establishing the Mobile Enterprise. The QoS aspects of the developed Mobile Host, like providing proper security and scalability, and the discovery of the provided services are addressed briefly. Further, the QoS and discovery analyses of the Mobile Host have raised the necessity for a middleware framework and the features and realization details of the MWSMF are discussed.

However to scale of Mobile Enterprise to the loads possible in mobile networks, we shifted some of its components to the cloud computing paradigm. The paper illustrated this categorical shift in terms of two application scenarios. It showed that MWSMF is horizontally scalable, thus allowing to utilize cloud's elasticity to meet load requirements in an easy and quick manner. It also illustrated different means to

scale the LAS based MPEG-7 services. Thus cloud computing is shown to scale the Mobile Enterprise dynamically.

Our future research in this domain will focus at surge computing and auto scaling so that Mobile Enterprise can scale according to the oscillating loads automatically. In the experiments discussed in this paper, we configured load balancer manually and one of our future research directions is to achieve more automation in scaling process. The planned framework detects loads automatically, dynamically adds more working nodes and automatically configures load balancer to accommodate new worker nodes. After loads drop, dynamically scalable MWSMF should shutdown unnecessary worker nodes. Another future research direction is to use Eucalyptus framework for cloud infrastructure instead of Amazon EC2, to show that public cloud's elasticity is achievable also in private clouds. We also want to extend this experience to our scientific computing cloud (SciCloud) project.

ACKNOWLEDGMENT

The research is supported by the European Social Fund through Mobilitas program, the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science and Eureka project "SITIO". The work was earlier supported by the Ultra High-Speed Mobile Information and Communication (UMIC) research cluster at RWTH Aachen University. The authors would also like to thank R. Levenshteyn and M. Gerdes of Ericsson Research for their help and support.

REFERENCES

- [1] S. N. Srirama, V. Šor, E. Vainikko, and M. Jarke, "Scalable mobile web services mediation framework," in *The Fifth International Conference on Internet and Web Applications and Services (ICIW 2010)*, 2010.
- [2] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to web services architecture," *IBM Systems Journal: New Developments in Web Services and E-commerce*, vol. 41(2), pp. 178–198, 2002. [Online]. Available: <http://researchweb.watson.ibm.com/journal/sj/412/gottschalk.html>
- [3] B. Benatallah and Z. Maamar, "Introduction to the special issue on m-services," *IEEE transactions on systems, man, and cybernetics - part a: systems and humans*, vol. 33, no. 6, pp. 665–666, November 2003.
- [4] S. N. Srirama, M. Jarke, and W. Prinz, "Mobile web service provisioning," in *AICT-ICIW '06: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*. IEEE Computer Society, 2006, p. 120.
- [5] S. Srirama and M. Jarke, "Mobile hosts in enterprise service integration," *International Journal of Web Engineering and Technology (IJWET)*, vol. 5, no. 2, pp. 187–213, 2009.
- [6] S. N. Srirama, M. Jarke, and W. Prinz, "Mobile web services mediation framework," in *Middleware for Service Oriented Computing (MW4SOC) Workshop @ 8th International Middleware Conference 2007*. ACM Press, 2007.
- [7] kSOAP2, "kSOAP2 - An efficient, lean, Java SOAP library for constrained devices," SourceForge.net, 2007. [Online]. Available: <http://sourceforge.net/projects/ksoap2>
- [8] Sun Microsystems, "JavaTM 2 Platform, Micro Edition (J2METM) Web Services Specification - Datasheet," Sun Microsystems, Inc., Tech. Rep., 2007.
- [9] S. N. Srirama and M. Jarke, "Mobile enterprise - a case study of enterprise service integration," in *3rd International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009)*. IEEE Computer Society, September 2009, pp. 101–107.
- [10] S. Srirama, M. Jarke, and W. Prinz, "Security analysis of mobile web service provisioning," *International Journal of Internet Technology and Secured Transactions (IJITST)*, vol. 1(1/2), pp. 151–171, 2007.
- [11] S. N. Srirama, M. Jarke, and W. Prinz, "MWSMF: A mediation framework realizing scalable mobile web service provisioning," in *International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications (Middleware 2008)*. ACM Press, 2008.
- [12] M. Ericsson and R. Levenshteyn, "On optimization of XML-based messaging," in *Second Nordic Conference on Web Services (NCWS 2003)*, November 2003, pp. 167–179.
- [13] S. Dustdar and M. Treiber, "Integration of transient web services into a virtual peer to peer web service registry," *Distributed and Parallel Databases*, vol. 20, pp. 91–115, 2006.
- [14] C. Lee, A. Helal, N. Desai, V. Verma, and B. Arslan, "Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services," *IEEE transactions on systems, man, and cybernetics - part a: systems and humans*, vol. 33, no. 6, pp. 682–696, November 2003.
- [15] S. N. Srirama, M. Jarke, W. Prinz, and H. Zhu, "Scalable mobile web service discovery in peer to peer networks," in *IEEE Third International Conference on Internet and Web Applications and Services (ICIW 2008)*. IEEE Computer Society, 2008, pp. 668–674.
- [16] Y. Kim and K. Lee, "A lightweight framework for mobile web services," *Journal on Computer Science - Research and Development*, vol. 24, no. 4, pp. 199–209, November 2009.
- [17] A. Mocan, E. Cimpian, M. Stollberg, F. Scharffe, and J. Scicluna, "Wsmo mediators," Online, December 2005, 10.12.2009. [Online]. Available: <http://www.wsmo.org/TR/d29/>
- [18] Apache Software Foundation, "Apache ServiceMix," 2007, 10.12.2009. [Online]. Available: <http://incubator.apache.org/servicemix/home.html>

- [19] R. Ten-Hove and P. Walker, "JavaTM Business Integration (JBI) 1.0 -JSR 208 Final Release," Sun Microsystems, Inc., Tech. Rep., August 2005.
- [20] M. Armbrust et al., "Above the clouds, a berkeley view of cloud computing," University of California, Tech. Rep., Feb 2009.
- [21] Dustin Amrhein et al., "Cloud computing use cases," A white paper produced by the Cloud Computing Use Case Discussion Group, Tech. Rep. Version 2.0, October 2009.
- [22] Amazon Inc., "Amazon elastic compute cloud (amazon ec2)," Online, 10.12.2009. [Online]. Available: <http://aws.amazon.com/ec2/>
- [23] Google Inc, "App engine java overview," 10.12.2009. [Online]. Available: <http://code.google.com/appengine/docs/java/overview.html>
- [24] Eucalyptus Systems Inc., "Eucalyptus," Online, 11.12.2009. [Online]. Available: <http://www.eucalyptus.com>
- [25] S. N. Srirama, O. Batrashev, and E. Vainikko, "Scicloud: Scientific computing on the cloud," in *10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010)*. IEEE Computer Society, 2010, p. 579.
- [26] S. N. Srirama, "Scientific computing on the cloud (scicloud)," Online, 10.12.2009. [Online]. Available: <http://ds.cs.ut.ee/research/scicloud>
- [27] W. Tarreau, "Haproxy architecture guide, version 1.1.34," Online, January 2006. [Online]. Available: <http://haproxy.1wt.eu/download/1.3/doc/architecture.txt>
- [28] Apache Software Foundation, "Apache module mod_proxy_balancer," Online, uRL last visited on 10th Dec 2009. [Online]. Available: http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html
- [29] J. Nielsen, *Usability Engineering*. San Francisco: Morgan Kaufmann, 1994.
- [30] M. A. Chatti, S. N. Srirama, I. Ivanova, and M. Jarke, "The mobilehost colearn system: Mobile social software for collaborative learning," *International Journal of Mobile Learning and Organisation (IJMLO), Special Issue on: "Developing Themes in Mobile Learning"*, vol. 4, no. 1, pp. 15–38, 2010.
- [31] M. Spaniol, R. Klamma, H. Janen, and D. Renzel, "LAS: A lightweight application server for mpeg-7 services in community engines," in *Int. Conf. on Knowledge Management (I-KNOW)*, 2006, p. 592.
- [32] Y. Cao, M. Spaniol, R. Klamma, and D. Renzel, "Virtual campfire - a mobile social software for cross-media communities," in *International Conference on New Media Technology and Semantic Systems (I-Media'07)*, September 2007.
- [33] W. Tarreau, "Making applications scalable with load balancing, revision 1.0," Sep 2006. [Online]. Available: http://1wt.eu/articles/2006_lb/

Developing Personalized Information Services for Mobile Commerce Location-Aware Applications

Christos K. Georgiadis

Department of Applied Informatics

University of Macedonia

Thessaloniki, GREECE

e-mail: geor@uom.gr

Abstract — The mobile setting adds unique characteristics to applications that can be used by mobile commerce client devices, such as ubiquity and location awareness. These devices are known to be limited in terms of computational power, input/output capabilities and memory, thus enhancing the mobile browsing user experience is realistic only if perceptual and contextual considerations are addressed. In this article, we attempt to define and analyze the issues of mobility, taking into consideration factors that would attract users to mobile commerce applications. We focus on how these issues may influence the user acceptance and the quality of personalized location-based services. To improve understanding of the mobile setting, a case study of a context-aware location-based application is designed and carried out. The application is capable to identify and to depict on the map user's current location, to search and detect routes, and to display various user personal points of interest/attractions, along user's current route. It is a personalized application, based on Microsoft MapPoint Web Service technology, in which each user receives information which is strictly related to his identity. Finally, to review certain personalized and user-friendly features of our approach, a typical application scenario is presented.

Keywords - location-based information; context-awareness; mobile commerce adoption; Web Services; mobile setting.

I. INTRODUCTION

Increased sophistication of mobile technology makes itself an ideal channel for offering context-aware personalized services to mobile users [1]. Such services actually, give pace to the rapid development of e-commerce conducted with portable devices, more commonly referred to as wireless or mobile commerce (m-commerce), as they are context-specific to each individual and thus, are capable to attract customer's attention [2]. The interface usability of mobile applications is a critical factor for the acceptance of m-commerce, as a good interface design allows users to achieve high performance [3]. Moreover, as 3G/UMTS services roll out, m-commerce is increasingly used to enable content delivery and payment for personalized and location-based services (LBS) such as image content (maps, photos, etc.), as well as video and audio content, including full length music tracks [4].

This work, undertakes a broad examination of the mobile setting in Sections II and III, to introduce significant

concepts that influence m-commerce user behavior, and to provide major considerations regarding m-commerce user acceptance and quality. In section IV we discuss about location-aware and personalized services to clarify personalization's connection to LBS. Section V entails a case study involving a modern approach to deal with personalized LBS in m-commerce. Here we first analyze the LBS user requirements and then we exploit Microsoft MapPoint Web Service technology to design search processes for addresses, attractions, routes, and creating processes of their respective maps. Section VI deals in detail with the major functional parts (classes and variable types) of our mobile application. Section VII contains an m-commerce information-oriented application scenario to demonstrate some of the major implementation concerns that must be taken under consideration to offer such personalized LBS.

II. ANALYZING MOBILE SETTING

System, environment, and user are actually three different viewpoints to classify the characteristics of the m-commerce applications [5]:

- System perspective - mobile applications present disadvantages, because they provide a lower level of available system resources [6].
- Environmental perspective - mobile applications enable users to access mobile Internet content anywhere and anytime (obviously, a big advantage).
- User perspective – is heavily influenced both by system/environmental characteristics and by certain aspects of the “mobility” concept [7][8][9]:
 - Spatial mobility, the extensive geographical movement of users.
 - Temporal mobility, the ability of users for mobile browsing while engaged in a peripheral task.
 - Contextual mobility, the dynamic conditions in which users employ mobile devices.

As it is depicted in Figure 1, we may distinguish three worth noticing attributes regarding mobile device usage [10]: first, users prefer to treat their mobile device in a quite **personal** way, and favor to access more personalized services (spatial mobility must be considered as the major reason behind this behavior). Second, users have usually **limited attention** as they manage their mobile devices (temporal mobility is the reason of this phenomenon). Finally, users manage their mobile devices in broadly mixed environments, appreciating all accommodations that may provide the **context-sensitive** mobile device **functionality**.

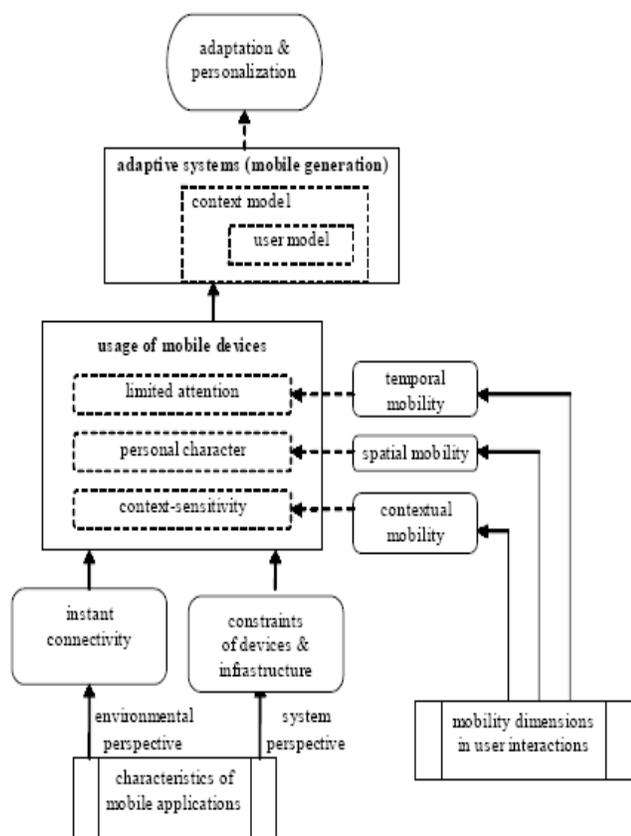


Figure 1. Analyzing mobile setting.

III. M-COMMERCE: USER ACCEPTANCE AND QUALITY

User requirements' satisfaction is among the most useful methods for measuring m-commerce application success. Undoubtedly, a key restraining factor concerning mobile services is financial: cost issues are raised due to confusing billing schemes and their potential dependency on versatile network connections [11]. However in this work, we will leave out cost related issues, and we will analyze the rest of interesting research results related with user satisfaction measuring [12], trying to identify how they can be used to improve the quality of m-commerce applications.

Wu and Wang in [12] revised effectively the technology acceptance model (TAM) [13][14] to cover the additional aspects of the m-commerce environments, as depicted in Figure 2. According to the main research results (which are consistent to a large extent with other researchers' work), various variables influence user adoption of m-commerce applications, as follows:

- M-commerce use can be predicted efficiently from the users' intentions, which are affected significantly by:
 - Perceived risk - the user's subjective expectation of suffering a loss in pursuit of the desired outcome of using m-commerce,

- Compatibility - the degree to which the innovation is perceived to be consistent with the potential users' previous experiences and needs, and
- Perceived usefulness - the degree to which a person believes that using a particular system would enhance his (or her) job performance.

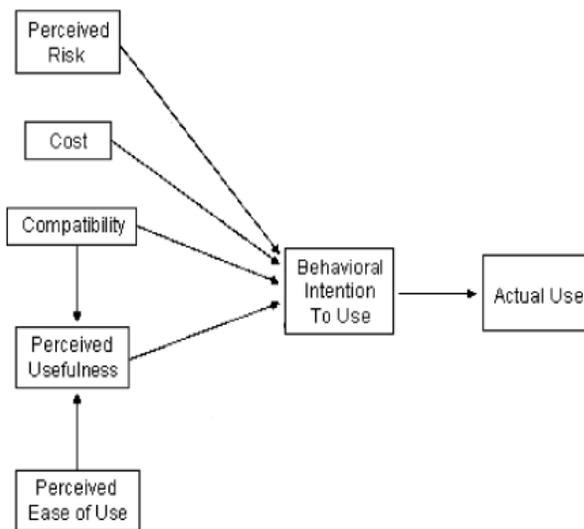


Figure 2. Mobile commerce acceptance model.

- User's possibility to engage in m-commerce transactions (behavioral intention to use), is affected also strongly, by:
 - Perceived risk,
 - Compatibility, which essentially has significant effect on the frequency of using m-commerce (actual use), and
 - Perceived ease of use - the degree to which a person believes that using a particular m-commerce application would be free of effort, which basically influences indirectly, through perceived usefulness variable.

Certain m-service quality criteria are directly or indirectly related with these user adoption variables. In detail, the software quality characteristics as defined by the ISO 9126 software quality standard [15] and adapted to m-commerce systems [16], are usability, reliability, functionality, and efficiency.

Usability aims at simplifying end users' actions and is associated with perceived ease of use. Functionality refers mainly to providing secure and suitable functions to end user. Thus, this quality characteristic is related both with perceived risk and compatibility. Reliability refers to systems tolerance on end users' actions, and consequently is associated also with perceived risk from a different perspective. Finally, efficiency's main attributes are response time and resource behavior, and therefore may be related with perceived usefulness.

The above analysis may provide constructive ideas to m-commerce application developers. It has certainly influenced

our work, to focus on quality issues by supporting user appreciated features satisfactory.

IV. PERSONALIZATION AND LOCATION-BASED SERVICES

A. Personalization

The ‘personal character’ emerges clearly from all quality characteristics in m-commerce context: it is the perception of the ease of use, the perception of risk, the perception of the usefulness and the perception of the consistency of the innovations with a particular user’s experiences. Personalization technology is therefore, one of the most prominent technologies in m-commerce systems: it is not just capable to get the most of the mobile devices’ limited resources. In addition, it provides m-commerce applications the efficiency of supporting users with the content they need in the most optimized approach. This is considered as an effecting way to holding users and to cultivate their loyalty [17].

B. Location-Based Services

LBS are, in general, informative services accessible with mobile devices through the mobile network; they are utilizing the ability to make use of the location of the mobile device [18]. There exist a broad range of different LBS. Figure 3 gives an overview on the main categories of LBS applications (the listing is certainly growing over time) [19].

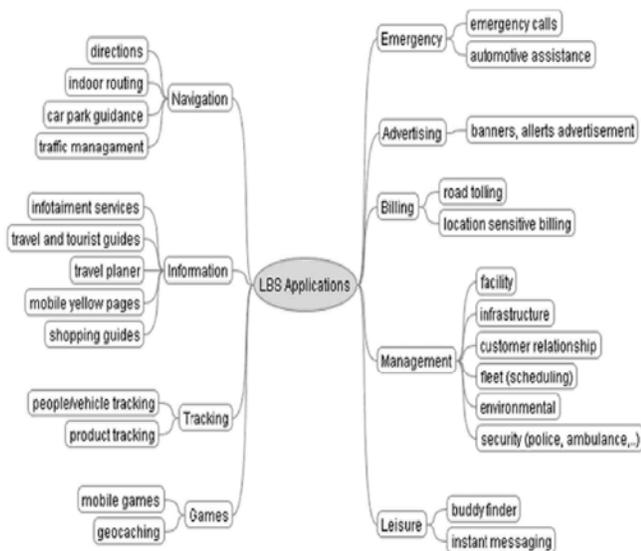


Figure 3. Location-based services application categories.

C. Related Work

In this work we are interested on certain navigation (e.g., providing users detailed directions about how to get to a desired destination starting from their current geographical location) and information LBS (e.g., providing a local street map and notifying about nearby places of interest).

This type of LBS is a characteristic example of mobile device’s context-sensitive functionality: it allows users to

obtain context-specific information to react upon. This makes mobile devices especially fit for providing time-critical and location-sensitive services. Several authors [17] [18] [20] have stressed this functional value of m-commerce, thereby acknowledged the social value that can be derived from mobile services. Moreover, Boll et al. in [21], presented an approach capable to integrate mobile LBS with personalized multimedia presentation. The mobile channel is indeed an extremely personal medium that users carry with them at all times. Consequently, it has become part of the user’s social context and everyday life, influencing all areas of society, such as businesses operation and employees work, advertising opportunities, personal communications, consumer purchases, location-aware services, information locating and retrieval [22].

According to Steiniger et al. in [19], LBS are quite different from more conventional internet based media (traditional guides, directories, maps etc.) because they are aware of the context in which they are being used and can adapt their contents and presentation accordingly. The relation between location and context concepts is very strong: user’s location is considered a major context parameter, because it can be used to characterize the user’s situation [17]. For example, knowing user’s location, the application can determine what other objects or people are near the user and even what kind of activity is occurring in that area. In addition, LBS may make use of a more detailed context. They actually give the possibility of a two way communication and interaction. Therefore the user tells the service provider (i.e., the m-commerce application) his actual context, like the kind of information he needs and his preferences (what), his time (when), and his position (where). This helps the provider of such location services to deliver information tailored to the user needs.

V. OUR CASE STUDY

A case study of a context-aware location-based application is designed and carried out. It is capable to identify the current location of the user and to provide the following information:

- depiction on the map of user’s current location,
- route search/detection and marking on the map of user’s preferred route while he is in move,
- displaying various user personal points of interest/attractions (such as his home, his office, his favorite restaurants, etc.), along user’s current route,
- depiction on the map of general interest type attractions (e.g. museums, subway stations and hospitals),
- editing of user personal attractions (adding new ones, deleting existing, etc.),
- editing of general interest attractions’ categories,
- dynamic update of user’s personal attractions and categories of attractions, based on advanced personalization techniques.

It is undoubtedly a customized application, since each user receives information which is strictly related to his identity. The personalization of services is accomplished by

exploiting identification information available from the mobile device's SIM card. In our case, obviously, there is no SIM card, as we use the .NET mobile device programming environment for simulating the mobile application's behavior.

Thus, in our case, users to enter the system should give some personal codes. That is, there is a form in which a registered user will present his credentials, while an additional form, can be used by a new user to create an account. As a result, a data base in the portable device carries user's data.

The existence of the device' database creates some additional problems. One has to do with the capacity of devices, which affects the size of the database. Since it cannot be very large, data of multiple users is difficult to be saved on the same device. In other words it is not feasible different users to use the same device, and their personal settings (passwords and attractions) to remain stored on the device. But this is obviously not so important, since the mobile device is used mainly as a personal (single user) client machine.

Another similar problem has to do with users' ability to connect to the system from different devices. They should create their account in any device from the beginning, and then adjust the parameters they want. A viable solution is to maintain a database on a server. The device will have to link to this server, and request and receive data whenever it wants (merge replication process).

Location-aware applications may base their operations not only on user's current location, but also on other information about the status of the user, such as time, and weather (context-aware wider character of location-based applications). Finding user's location is possible by various technologies, but as this application was developed and tested locally, finding position and movement-on-a-path operations are made by simulation. That is, a virtual movement of a certain user and its current position changes are defined through the application environment. Certainly, finding the virtual user's current location is accomplished through our application design.

The application consists of three separate units, which interact with each other:

- User interface.
- Mobile database on the device, which deletes its data when session ends.
- Server database, which stores and maintains all mobile users' data, and additionally sends the appropriate data to the proper mobile database (based on user identity information).

A. MapPoint Web Service Technology

MapPoint Web Service (MWS) technology [23] is used to design search processes for addresses, attractions, routes, and creating processes of their respective maps. The MWS is actually a Microsoft's Internet service, which is designed to work using the Graphical Information System (GIS) and responds to various scenarios mapping, which involve different types of applications, such as portals, web pages, but mainly mobile applications. In conjunction with

MapPoint Location Server (server which detects the location of a mobile device) they create a "package" particularly effective for implementing an integrated location-based application.

MWS' typical capabilities are to display maps (render), to find the coordinates of an address (geo-coding), to find the address through given coordinates (reverse geo-coding), to search for addresses within walking distance from user's current location, and to present route instructions.

MWS has four basic services:

- Find Service: finding addresses, attractions, coordinates of points, and points within a distance from the position of the user, depending on the type and how to do the search.
- Render Service: creating the image of the map, based on the data we want
- Route Service: creates a path between two locations we have stated. The type of the route and the points from which the path will pass, depends on the data we provide.
- Common Service: contains classes, which are common in all three previous services, such as the definition of DataSource.

The general categories for the attractions available from MWS are:

- Airport
- AncientSite (Archaeological sites)
- Hospital
- MetroStation
- Museum
- ShoppingCenter (Convention Centers)
- Stadium (Stages-Sports Facilities)

The MapPoint Location Server (MLS) detects the current user's location, identifying the mobile device (including simple mobile devices, Pocket PCs, PDAs, SmartPhones and all registered devices). To find the position, many methods are available, such as Cell-ID (Cell Identification), A-GPS (Assisted GPS), triangular methods, etc.

B. Mobile and Server Database

The mobile device's database contains three tables: ID, POI (Point Of Interest) and CAT_POIS (Category of POIs). The first one maintains user's personal credentials, while the last ones user's personal sights and categories, that is those points of interest and categories that user has added on his profile. Server database contains tables with corresponding names. They exchange data with each other, using Merge Replication process of the Internet Information Services (ISS) web server. The exchange of data is accomplished in both directions, meaning that mobile database tables cannot only send data to server, but also may receive data from it.

Microsoft SQL Server Management Studio is used to create server database. And mobile database tables are automatically generated and updated through synchronization.

Table ID has two columns, the username and password. Table POI contains the name, the coordinates and a brief description for locations that a certain user has personally

selected. Table CAT_POIS consists of two columns, the username and poi-catname. The second column stores the name of user's selected general category. Obviously, a certain username may appear in several records of the table, to represent user's multiple selections on the attractions' general categories.

VI. FUNCTIONAL COMPONENTS OF THE MOBILE APPLICATION UNDER STUDY

We will present in detail the major functional parts (classes and variable types) of our mobile application.

A. Application Classes

The application consists of seven forms and eight code classes. Six of the classes used to connect to the MWS, and perform a different type of work. The seventh class is used for synchronizing server and mobile databases. Finally, the eighth class is used for advanced personalization processes.

1) GetAddress Application Class

This class finds the coordinates of one or more addresses, based on user provided information, such as the address of point (street name and number), or the zip code. Without providing zip code, then most likely to be found more of one addresses, otherwise is found only one. The results are stored on a FindResults variable (MWS variable type).

The class is used during the storage of user's personal attractions. And its purpose is twofold. Firstly, to enable users to choose the desired location-point among others, although they have not entered the zip code and thus several addresses are found. Secondly, to store the coordinates of a point finally chosen by the user, in case that he asks to illustrate the point on the map. In such a case, location's coordinates are ready to be used, and there is no need to search them through the MWS.

2) Generalpois1 Application Class

This class finds all general attractions of a user selected category. Its parameters are:

- entityname: the category of attractions that user has selected,
- distance: the distance over which should be the results desired by the user,
- mylatlongs: holds the coordinates of the user's current location, which will be used if the user wants attractions within a specified distance.

Not setting the distance parameter means that user wants to see all the sights in that category, without distance limits. Another important issue is that the MWS SearchContext parameter cannot limit the search to specific region (e.g. Attica), but only to specific country (e.g. Greece). A viable solution to this issue is by using code statement like the following:

```
If fr.FoundLocation.Entity.DisplayName.IndexOf
("Attiki") <> -1 Then ...
```

This line of code checks over whether found locations are located in Attica. The check is made on whether the found address contains the word Attica.

The search process must be carried out twice, because the size of foundEntities MWS variable must be defined in order to be able to save the results that user desires.

The other case is when user wants the attractions at a specific distance from its current location to be displayed. The parameter distance is expressed in meter units. It should be converted into degrees, so that it can take part in calculations with the attractions' and current location's coordinates.

3) Generalpois2 Application Class

This class accepts the results found by the generalPois1 class and creates the map, with the representation of all the attractions found, plus the user's current location. Depending on the number of results (MWS variable entitiesnum) that have been already found, the size of both mylocations MWS variable (items that will appear) and pushpins MWS variable (the "pins" to be placed on the map and will represent the respective point of mylocations) is defined. Once given the coordinates, then the map is created by using the ViewByBoundingLocations class.

The label parameter of pushpins variable stores a number, which represents the series that the attraction has to the list of results. So, a ListBox user interface control, which stores the name of the attractions, along with the pushpins' label parameter is capable to inform users what exactly represents any numbers displayed on the map.

4) Viewpois Application Class

This class returns an image with the user's current location and the potential various attractions that he personally selected to be displayed. It works just as the generalpois2 class. An important differentiation is the zoom parameter's setting of the views MWS variable.

5) MakeRoute Class

This class creates a path between two points. Depending on the user's selections, various personal attractions can also be presented along this path.

Each time the MakeRoute class is called, it creates a new path, which is part of the journey up the first time that the class was called for this route. Each time the route is divided into segments (sections). In every move, the user's current location is moved to the next segment, to show that the user has moved on set route. In order to have a proper representation of the move, we should have kept the number of parts of the route that was created when the class was invited for the first time on this route.

The MWS variable kindofRoute sets the type of route to be used for the creation of the route. The SegmentPreference.Shortest value creates the shortest route (based on distance), between points, while the SegmentPreference.Quickest value creates the fastest route, based on the time of transition from the beginning to the end.

An example of code to create a certain path follows:

```
route = routeService.CalculateSimpleRoute(latlongs,
    "MapPoint.EU",
    SegmentPreference.Shortest)
```

Three different cases exist: to not display any attraction along the route, to display all the user's personal sights, and to display those user's personal sights, which are in a certain distance from the user. The choice of the case is based on MWS pois variable's value. In general, the movement is divided, for all three previous cases into three sections:

- the movement carried out from beginning to the last but one part,
- the movement carried out in the penultimate part,
- the movement as user has reached the end of the path.

6) MakeRoutePois Application Class

This class creates a path between the user's current location and a certain attraction which is selected by the user. The class is differentiated from the previous MakeRoute class in its way to present the map. The image's center is not the user's current location: what is important here is to display the whole path. For this reason, the previously mentioned MWS ViewByBoundingLocations class is not used. Instead, the MWS variable mapSpec is used to hold the path.

7) Replication Application Class

This class is used to synchronize server and mobile databases. The parameter InternetUrl holds the virtual address, set up on our server, to allow for communication between the bases:

```
replication.InternetUrl =
    "http://192.168.1.2/ReplSync/sqlcesa30.dll"
```

ReplSync is the folder that has been created for the exchange of data, while sqlcesa30.dll is an essential dynamic link library .NET file for exchanging data.

8) AdvPersonalization Application Class

This class is used to apply additional personalization mechanisms which are distinguished as:

- content-based considerations – personal attractions and categories of general attractions are dynamically updated in user profile, based on user interactions with the mobile application. This means that this class monitors user actions and evaluates the frequency of user requests on attractions not belonging in his profile. E.g., even if a particular user has not declared the White Tower attraction in his profile, it may be part of it, if for a certain period of time user frequently asked to see it on his mobile phone. A parameterized threshold is used for deciding the required value of frequency which allows the alteration of user profile.
- collaborative-filtering considerations – server database is queried for examining other users'

profiles. Like-minded users, namely users with many common selections (declared explicitly) on their profiles, are used as a simple recommendation engine: personal attractions and categories of general attractions are dynamically updated in a particular user profile, based on the preferences of other users with similar interests.

B. Application Variable Types

A number of variable types, mainly related to functions of MWS, are used in all classes. Their role in the design of mobile application is of critical importance.

1) FindServiceSoap, RouteServiceSoap Variable Types

These variable types, need to connect to the MWS, and thus before being used they had to be 'activated' by using user name/password combinations, (acquired during registration phase in MWS).

```
findService.Credentials = New
    System.Net.NetworkCredential (myUserName,
    myPassword)
```

For efficiency reasons, PreAuthenticate parameter should be set properly:

```
findService.PreAuthenticate = True
```

2) FindSpecification Variable Type

This object is used by the FindServiceSoap, to carry out searches for Attractions. The parameters that should be set are: the category name of the attractions that we are interested (EntityTypeNames), the map in which the search will be done (DataSourceName), and a number of Options, capable to reduce the volume of returned results. The main options are:

- *Range*: the maximum number of MWS provided results. By default, this number is set to 25. It may be set up to 500.
- *ThresholdScore*: indicates the degree of correlation with the search. Any result "returned" from MWS has such a value, which is stored in *ThresholdScore*. The default option value is set to 0.85. By reducing this value, we increase the amount of results, as we ask practically to reduce the degree of correlation.
- *SearchContext*: an integer that indicates the region, which would limit the search. For example, *DataSource* may specify the map of Europe (MapPoint.EU), but if we are interested only on the results concerning a particular country, then we must set the region option to the corresponding value (e.g., 98 is for Greece).

3) Pushpin() Variable Type

These variables are used to display 'tacks' (pushpins) on the map. A Pushpin represents a mark that we would like to see on the map. Related parameters are:

- *LatLong*: its coordinates.

- *IconDataSource*: a source-repository with images, capable to represent *Pushpins* on maps, as they can be fitted to the point defined by the coordinates. Each user may create his own images, and place them into his personal *IconDataSource*. Or, he may use the default parameter (*MapPoint.Icons*).
- *IconName*: the name of the Pushpin icon, contained in *IconDataSource*. In our application, there are two different icons (namely "0" and "1". Both depict a differently tinted tack ("0" for the blue and "1" for the red one).
- *Label*: the caption that appears above the Pushpin.

4) *ViewByHeightWidth, ViewByBoundingLocations Variable Types*

These types are used for describing two alternative map representations. *ViewByHeightWidth* defines a central point, based on which the map will be shown. There is no guarantee that all the defined Pushpins are being displayed. The map is created so that the center of the provided image is the desired central point (usually user's current location), and the attractions' presence on the map is depended on map's size.

On the other hand, *ViewByBoundingLocations* declares that image map will be created adapted to objects shown on the map. Thus, the map will present all the desired points/attractions.

5) *MapSpecification Variable Type*

This type of variables is used to create maps, based on the provided parameter values. The most significant of them are:

- *DataSource*: the map of MWS.
- *Views*: alternative map representations, using *ViewByHeightWidth*, and *ViewByBoundingLocations* variable types.
- *Options*: stores the size (height and width) of PictureBox control (form element), in which the map will be placed. Map creation is adjusted to the size of the PictureBox. Furthermore, very helpful is the Zoom parameter. Its default value is 1, while the lowest price we can get is 0. It must be noted that Options parameter should be initialized before use.
- *Pushpins*: the potential Pushpins that we are interested to be placed on the map.
- *Route*: in case that we have defined a route (via a specific *Route* MWS variable type), this parameter may cause route design and display on the map.

VII. USING THE LOCATION-BASED SERVICES

In this section, we will present a typical application scenario to demonstrate specific contextual, personalized and user-friendly features of our approach.

Let us suppose that user Alice wishes to see on her mobile the route from her current location to the Airport. Figure 4 depicts the navigation chart of the application, so Alice has to follow the selections' path:

'Map'--> 'Find Route' --> 'To General Attraction'

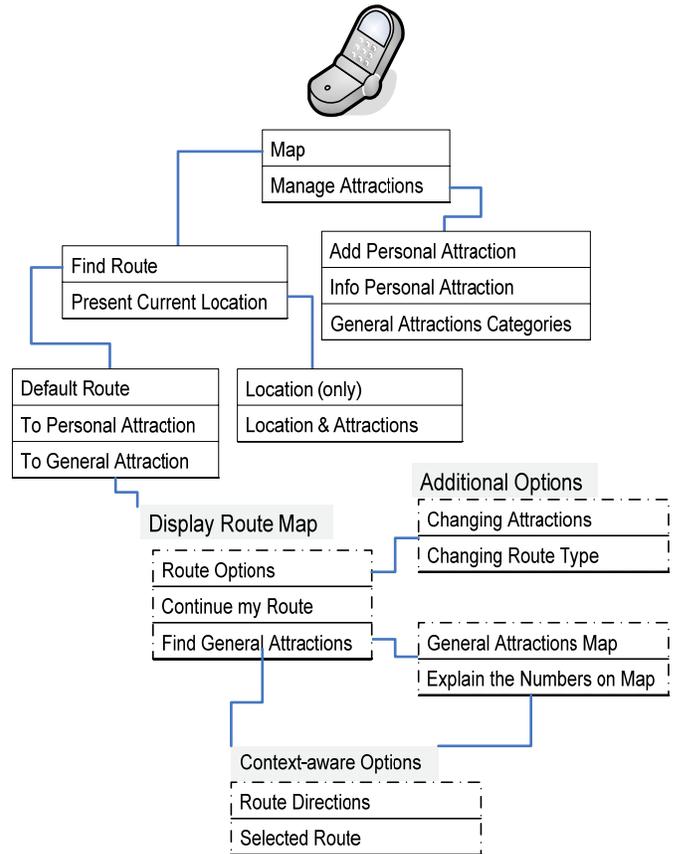


Figure 4. The GUI navigation chart of the application.

Certainly, all attractions belonging to the general categories available from MWS (which are mentioned previously in Section VI), may be used as the destination location of Alice's route. But not only that: Alice may easily choose Airport destination because all categories of attractions that are not of interest for her, are not provided as possible selections on her mobile device. This can be done, because in the past Alice has used the following option to declare her interests:

'Manage Attractions' --> 'General Attractions Categories'

Or, Alice in the past had too many times asked about Airport, and consequently her profile was dynamically updated by AdvPersonalization class, to include this attraction. It must be noticed that, this was possible because Alice in her profile has explicitly allowed the dynamic content-based alteration of her preferences.

Alice, as shown in Figure 5, may see now a map which presents a route according to her preferences, created by MakeRoute class. 'Eiste edo' pin declares her current location, while 'Arxi' (Start) and 'Goneis' (Parents) pins are personal attractions that are close to this route.



Figure 5. Route map display. “Είστε Εδώ” stands for “You are here”, “Arxi” for “Start” and “Goneis” for “Parents”

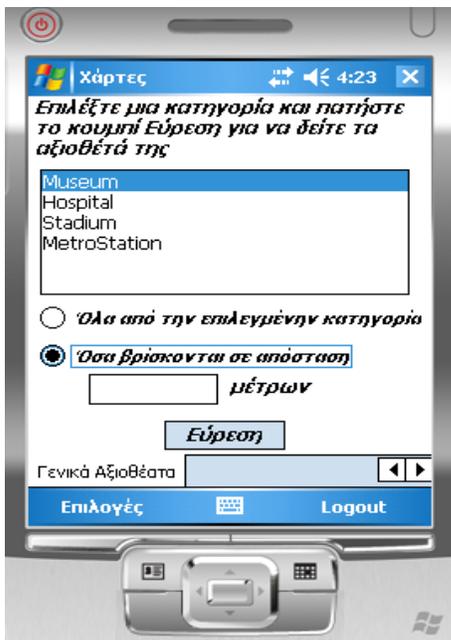


Figure 6. Find general attractions option.

As she moves, Alice decides to stop get informed about the closest personal attractions, or even to change the distance parameter that decides if a personal attraction is close to the route. She just has to select:

‘Route Options’ --> ‘Changing Attractions’

The map will reflect the changes regarding the presence of her personal attractions on the route. If Alice meets accidentally a friend with a car, willing to take her to the airport, she may select:

‘Route Options’ --> ‘Changing Route Type’

In this way, she receives a different route on the map, taking into account the potential one-way streets in this area.

She also may enrich her map, by selecting the *‘Find General Attractions’* submenu. In this way, she may further elaborate her preferences on her stored profile: she may choose which general attractions of her profile (based on their category or on their distance from the specific route) to be displayed on the map (see Figure 6). All these attractions are depicted as numbered pins on the map, but there is a proper menu option to get the explanations about these numbers on the map.



Figure 7. Presenting route instructions.

If Alice decides to get more detailed information on route directions (see Figure 7), she has to select:

‘Context-aware Options’ --> ‘Route Directions’

Also, Alice may ask to see the whole route on her mobile device, using the *‘Selected Route’* option of the *Context-aware Options* menu.



Figure 8. Presenting selected route. “Archeologiko Mousio” stands for “Archaeological museum” and “Eiste edo” stands for “You are here”

In Figure 8, the route to the ‘Archeologiko Mousio’ (Archaeological museum), an attraction of the museum category is presented. Note the difference between Figures 5 and 8: user’s current location (‘Eiste Edo’) is no longer at the center of the display to depict the whole selected route.

It must be noticed also that Alice had not declared this attraction or its category (museum) in her profile and also she had never asked information about it. However, other users with similar declared attributes in their profiles had chosen this attraction and thus her profile was dynamically updated by AdvPersonalization class. This was done because Alice in her profile has explicitly allowed the dynamic collaborative-filtering alteration of her preferences.

The application has similar functionality when users requesting route maps with destinations being personal attractions. It must be also clarified, that the ‘Default Route’ is making use of a specific destination location, namely ‘Telos’ (end), which has taken its value when user made the registration. In this way, every user has a quick way to ask a route map from his current location to a specific (frequently used) destination.

If Alice has no interest to be informed about route maps, she may select:

‘Map’--> ‘Present Current Location’

With this submenu she has access to the most simple location-based services, that is to display on the map her current location with or without nearby attractions (general and/or personal).



Figure 9. Managing personal attractions.

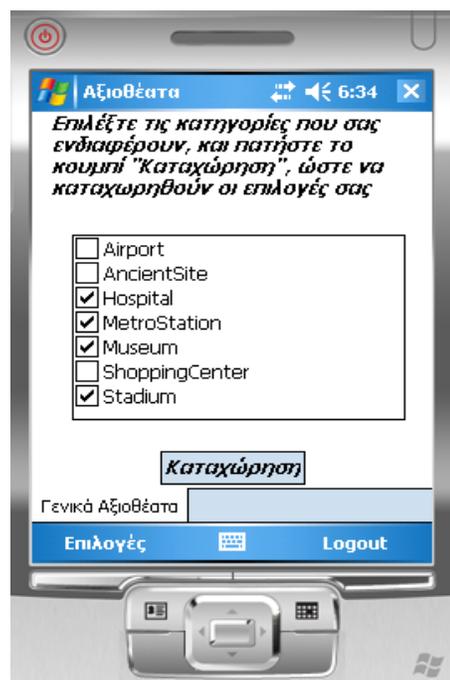


Figure 10. Managing general attractions.

Finally, it must be noticed that ‘Manage Attractions’ option provides powerful administrative functions to Alice, to add, edit, or delete personal attractions (see Figure 9), and/or categories of general attractions (see Figure 10). In this way, Alice may update her profile and enjoy

personalized location-based services. An additional administrative function is the allowance (or not) of the advanced personalization mechanisms, and consequently the dynamic alteration of her profile.

VIII. CONCLUSION AND FUTURE WORK

In this paper we discussed the mobile setting issues, m-commerce adoption concerns, location-based services, context-awareness, and its significance to personalize mobile applications. The design of an indicative end user application is discussed, which makes use of context parameters (such as user's location and preferences), and demonstrates how location-aware context is a powerful enabling factor for any m-commerce application.

In order to demonstrate certain personalized and user-friendly features of our approach, a typical LBS scenario is analyzed and presented. Future research efforts will be focused on investigating a more detailed categorization of users' requirements for LBS, as well as on formulating user's behavior in a flexible context model. Through the use of this context model and the development of more applications, we hope to further increase our understanding of personalization and context-awareness.

ACKNOWLEDGMENT

The author would like to thank Stavros Gitsioudis and Apostolos Provatidis for their contribution in the development of the application case study in this paper.

REFERENCES

- [1] C.K. Georgiadis, "Mobile Commerce Application Development: Implementing Location-aware Information Services", in Proc. of the 5th Advanced International Conference on Telecommunications (AICT 2009), Venice/Mestre, Italy, IEEE CS, pp. 333-338, 2009.
- [2] S.Y. Ho and S.H. Kwok, "The Attraction of Personalized Service for Users in Mobile Commerce: An Empirical Study", ACM SIG eCOM, http://www.sigecom.org/exchanges/volume_3/3_4-Ho.pdf, 22.6.2009, 2003.
- [3] S. Koukia, M. Rigou, and S. Sirmakessis, "The Role of Context in m-Commerce and the Personalization Dimension", in Proc. of the 2006 IEEE/WIC/ACM Intern. Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE CS, Washington, DC, pp. 267-276, 2006.
- [4] G. Elliot and N. Phillips, *Mobile Commerce and Wireless Computing Systems*, Addison Wesley – Pearson Education. Harlow, England, 2004.
- [5] M. Chae and J. Kim, "What's so different about the mobile Internet?", *Communications of the ACM*, 46(12), pp. 240-247, 2003.
- [6] C. Ververidis, G.C. Polyzos, and K.-P. Mehdi, "Location-Based Services in the Mobile Communications Industry", *Encyclopedia of E-Commerce, E-Government and Mobile Commerce*, Idea Group Reference, Hershey, USA, 2006
- [7] M. Kakihara and C. Sorensen, "Expanding the "mobility" concept", *ACM SIGGROUP Bulletin*, 22(3), pp. 33-37, 2001.
- [8] M. Kakihara and C. Sorensen, "Mobility: An Extended Perspective", in Proc. of the 35th Hawaii International Conference on System Sciences, 2002.
- [9] Y.E. Lee and I. Benbasat, "A framework for the study of customer interface design for mobile commerce", *International Journal of Electronic Commerce* (1086-4415/2004), 8(3), pp. 79-102, 2004.
- [10] C.K. Georgiadis, "Adaptation and Personalization of User Interface and Content", Chapter in "HANDBOOK OF RESEARCH ON MOBILE MULTIMEDIA", I.K. Ibrahim (Ed.), Information Science Reference Inc. (IGI Group-Idea), ISBN 1-59140-866-0, pp. 266-277, May 2006.
- [11] M. Bina, D. Karaiskos, and G.M. Giaglis, "Motives and Barriers Affecting the Use of Mobile Data Services", in Proc. of the IEEE International Conference on Mobile Business (ICMB 2007), 2007.
- [12] J.H. Wu and S.C. Wang, "What drives mobile commerce? An empirical evaluation of the revised technology acceptance model", *Information & Management*, 42(5), pp. 719-729, 2005
- [13] F.D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technologies", *MIS Quarterly* 13(3), pp. 319-340, 1989.
- [14] V. Venkatesh and F.D. Davis, "A theoretical extension of the technology acceptance model: four longitudinal field studies", *Management Science* 46(2), pp. 186-204, 2000.
- [15] ISO/IEC 9126-1, "Software product evaluation –quality characteristics and guidelines for the user", Geneva: International Organization for Standardization, 2001.
- [16] J. Garofalakis, A. Stefani, V. Stefanis, and M. Xenos, "Quality attributes of consumer-based m-commerce systems", *IEEE Int. Conf. on E-Business and Telecommunication Networks, ICETE, ICE-B, Barcelona, 2007*, pp. 130-136.
- [17] C.K. Georgiadis, I. Mavridis, and A. Manitsaris, "Context-based Humanized and Authorized Personalization in Mobile Commerce Applications", *IJCIS*, Vol. 3, No. 2, pp. 1-9, 2005.
- [18] K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. Veijalainen, A. Katanosov, and H. Tirri, "Developing GIS-Supported Location-Based Services, in Proc. of WGIS'2001 – First International Workshop on Web Geographical Information Systems, Kyoto, Japan, pp. 66-75, 2001.
- [19] S. Steiniger, M. Neun, and A. Edwardes, "Foundations of Location Based Services", *Lecture Notes on LBS*, Department of Geography, University of Zürich, http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf, 25.6.2009, 2006.
- [20] K. t. Hagen, M. Modsching and R. Kramer, "A location-aware mobile tourist guide selecting and interpreting sights and services by context matching", in Proc. of the 2nd Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2005), San Diego, California, 2005.
- [21] S. Boll, J. Krosche, and A. Scherp, "Personalized Mobile Multimedia meets Location-Based Services", in Dadam, P., Reichert, M., eds.: *Workshop "Multimedia - Information systems"* at the 34th Annual meeting of the German Computer Society (INFORMATIK 2004), Vol. 51 of LNI., Ulm, Germany, GI, pp. 64-69, 2004.
- [22] C.K. Georgiadis and S.H. Stergiopoulou, "Mobile Commerce Application Development: Implementing Personalized Services", in Proc. of the International Conference on Mobile Business 2008 (ICMB 2008), Barcelona, Spain, July 2008, IEEE CS, pp.201-210, 2008.
- [23] MapPoint Web Service, Microsoft Corp., <http://www.microsoft.com/MapPoint/en-us/default.aspx>, 20.7.2009, 2008.



www.iariajournals.org

International Journal On Advances in Intelligent Systems

✦ ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

✦ issn: 1942-2679

International Journal On Advances in Internet Technology

✦ ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING

✦ issn: 1942-2652

International Journal On Advances in Life Sciences

✦ eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO

✦ issn: 1942-2660

International Journal On Advances in Networks and Services

✦ ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION

✦ issn: 1942-2644

International Journal On Advances in Security

✦ ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

✦ issn: 1942-2636

International Journal On Advances in Software

✦ ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS

✦ issn: 1942-2628

International Journal On Advances in Systems and Measurements

✦ ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL

✦ issn: 1942-261x

International Journal On Advances in Telecommunications

✦ AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA

✦ issn: 1942-2601