SERVICE COMPUTATION 2010

The Second International Conferences on Advanced Service Computing

November 21-26, 2010 - Lisbon, Portugal

ComputationWorld 2010 Editors

Ali Beklen, IBM Turkey, Turkey
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Wolfgang Gentzsch, EU Project DEISA, Board of Directors of OGF, Germany
Teemu Kanstren, VTT, Finland
Arne Koschel, Fachhochschule Hannover, Germany
Yong Woo Lee, University of Seoul, Korea
Li Li, Avaya Labs Research - Basking Ridge, USA
Michal Zemlicka, Charles University - Prague, Czech Republic

SERVICE COMPUTATION 2010

Foreword

The Second International Conferences on Advanced Service Computing [SERVICE COMPUTATION 2010] was held between November 21 and 26 in Lisbon, Portugal and continued a series of events targeting service computation on different facets. It considered their ubiquity and pervasiveness, WEB services, and particular categories of day-to-day services, such as public, utility, entertainment and business. The ubiquity and pervasiveness of services, as well as their capability to be context-aware with (self-) adaptive capacities pose challenging tasks for services orchestration and integration. Some services might require energy optimization, some might requires special QoS guarantee in a Web-environment, while others require a certain level of trust. The advent of Web Services raised the issues of self-announcement, dynamic service composition, and third party recommenders. Society and business services rely more and more on a combination of ubiquitous and pervasive services under certain constraints and with particular environmental limitations that require dynamic computation of feasibility, deployment and exploitation.

We take here the opportunity to warmly thank all the members of the SERVICE COMPUTATION 2010 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to SERVICE COMPUTATION 2010. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the SERVICE COMPUTATION 2010 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that SERVICE COMPUTATION 2010 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the area of advanced service computing.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the beautiful surroundings of Lisbon, Portugal.

SERVICE COMPUTATION 2010 Chairs:

Ali Beklen, IBM Turkey, Turkey
Emmanuel Bertin, Orange-ftgroup, France
Hepu Deng, RMIT University - Melbourne, Australia
Paul Humphreys, Ulster Business School/University of Ulster, UK
Arne Koschel, Fachhochschule Hannover, Germanyh
Li Li, Avaya Labs Research - Basking Ridge, USA
Ying Li (李 影), IBM Research - China, China
Michele Ruta, SisInfLab / Politecnico di Bari, Italy
Toyotaro Suzumura, IBM Research / Tokyo Research Laboratory, Japan

SERVICE COMPUTATION 2010

Committee

SERVICE COMPUTATION Advisory Chairs

Academia
Hepu Deng, RMIT University - Melbourne, Australia
Paul Humphreys, Ulster Business School/University of Ulster, UK
Arne Koschel, Fachhochschule Hannover, Germany
Michele Ruta, SisInfLab / Politecnico di Bari, Italy

Industry
Ali Beklen, IBM Turkey, Turkey
Toyotaro Suzumura, IBM Research / Tokyo Research Laboratory, Japan
Emmanuel Bertin, Orange-ftgroup, France
Ying Li (李 影), IBM Research - China, China
Li Li, Avaya Labs Research - Basking Ridge, USA

SERVICE COMPUTATION 2010 Technical Program Committee

Isara Anantavrasilp, Technischen Universität München, Germany
Ismailcem Budak Arpinar, University of Georgia, USA
Irina Astrova, Tallinn University of Technology, Estonia
Ali Beklen, IBM Turkey, Turkey
Emmanuel Bertin, Orange-ftgroup, France
Noureddine Boudriga, University of Carthage, Tunisia
Sujit Kumar Chakrabarti, Philips Healthcare - Bangalore
Anis Charfi, TU Darmstadt, Germany
Dickson Chiu, Dickson Computer Systems, Hong Kong
Leandro Dias da Silva, Federal University of Alagoas, Brazil
Florian Daniel, University of Trento, Italy
Hepu Deng, RMIT University - Melbourne, Australia
Dwight Deugo, Carleton University, Canada
Tommaso Di Noia, Politecnico di Bari, Italy
Leandro Dias da Silva, Federal University of Alagoas - Maceió, Brazil
José Valente de Oliveira, Universidade do Algarve, Portugal
Erdogan Dogdu, TOBB University of Economics and Technology - Ankara, Turkey
Schahram Dustdar, Vienna University of Technology, Austria
Geoffrey Fox, Indiana University, USA
Vasilis Friderikos, King's College London, UK
Martin Gaedke, Chemnitz University of Technology, Germany
G.R. Gangadharan, Politecnico di Milano, Italy
Luisa Gargano, Università di Salerno, Italy
Paolo Giorgini, University of Trento, Italy

Luis Gomes, Universidade Nova de Lisboa / UNINOVA-CTS - Monte de Caparica, Portugal
Victor Govindaswamy, Texas A&M University - Texarkana, USA
Michael Hafner, University of Innsbruck, Austria
Jon Hall, Open University, UK
Paul Humphreys, Ulster Business School/University of Ulster, UK
Mirjana Ivanovic, University of Novi Sad, Serbia
Jinlei Jiang, Tsinghua University - Beijing China
Hai Jin, Huazhong University of Science and Technology - Wuhan, China
Paul Johannesson, Stockholm University, Sweden
Dimitris Karagiannis, University of Vienna, Austria
Arne Koschel, University of Applied Sciences and Arts - Hannover, Germany
Natalia Kryvinska, University of Vienna, Austria
Li Li, Avaya Labs Research - Basking Ridge, USA
Ying Li (李 影), IBM Research - China, China
Shih-Hsi Liu, California State University - Fresno, USA
Jan Lucenius, National Defence University - Helsinki, Finland
Kurt Maly, Old Dominion University, USA
Mihhail Matskin, KTH, Sweden
Susana Munoz Hernández, Universidad Politécnica de Madrid, Spain
Andreas Nearchou, University of Patras, Greece
Christos Nikolaou, University of Crete, Greece
Ingo Pansa, Karlsruhe Institute of Technology (KIT), Germany
Witold Pedrycz, University of Alberta, Canada
Juha Röning, University of Oulu, Finland
Michele Ruta, SisInfLab / Politecnico di Bari, Italy
Timothy K. Shih, Asia University - Wufeng, Taiwan
Eva Söderström, University of Skövde, Sweden
George Spanoudakis, City University - London, UK
Young-Joo Suh, POSTECH, Korea
Toyotaro Suzumura, IBM Research / Tokyo Research Laboratory, Japan
Vladimir Stantchev, Berlin Institute of Technology, Germany
Takeshi Tsuchiya Waseda University, Japan
Xia Wang, Jacobs University - Bremen, Germany
Zhengping Wu, University of Bridgeport, USA
Qi Yu, Rochester Institute of Technology, USA
Konstantinos Zachos, City University - London, UK
Arkady Zaslavsky, Lulea University of Technology, Sweden
Jelena Zdravkovic, SU/KTH - Stockholm, Sweden
Wenbing Zhao, Cleveland State University, USA

Copyright Information

Table of Contents

Generation of choreography skeletons from web service definitions

Annett Laube and Patrick Winkler
Bern University of Applied Science
Devision of Computer Science
Biel/Bienne, Switzerland
Email: annett.laube@bfh.ch, winkp1@bfh.ch

***Abstract*—Modern IT landscapes underlie constant evolution. Modeling activities - as a basis for continuous monitoring and maintenance - stay often behind. Service models describe the way how services interact. We propose reverse engineering techniques to generate choreography skeletons from web service definitions. We describe the necessary transformation to generate a detailed and consistent choreography than can be easily completed and merged with existing choreographies. We discuss the restrictions of the generated WS-CDL skeletons and how they can be overcome.**

***Index Terms*—web service, choreography, service model, reverse engineering**

I. Introduction

IT landscapes in industry or finance are often a result of a long evolution. Despite continuous efforts to keep models of systems and components up-to-date, rarely IT landscape models are consistent and complete.

The *landscape model* (sometime also called system model) consists often of two parts: the physical infrastructure and the service model. The physical infrastructure reflects in great detail components composed in the following groups: computer hardware (e.g. processing power, memory, etc.), computer software (e.g. OS, server applications), and network devices (e.g. links, traffic controllers - hubs/switches/gates/routes).

The *service model* is an abstraction of the set of all services. It defines the way how the services interact by exchanging messages and how more complex services are created by combining services. The terms *orchestration* and *choreography* describe two different aspects of creating business processes from composite web services.

Orchestration refers to an executable business process that can interact with both internal and external web services. Orchestration represents the composition from the viewpoint of the parties involved in this composition.

A *choreography* description concerns the composition of web services seen from a global viewpoint focusing on the common and complementary observable behavior. Choreography is particularly relevant in a setting where there is not a single coordinator. Choreography tracks the message sequences among multiple parties and sources - typically the public message exchanges that occur between web services - rather than a specific business process that a single party executes [1]. Typical examples are a travel agency that offers a broad range of services including air and train travel, bus tickets, hotels, car rental, excursions, insurance etc. or a company that wishes to purchase a fleet of cars from automobile suppliers, which in turn request quotes for specific bill or material items from their component manufacturers [2].

The main use of a choreography description is to precisely define the sequence of interactions between a set of cooperating web services in order to promote a common understanding between participants and to make it as easy as possible to automatically validate conformance, ensure interoperability and increase robustness [2].

We want to automate the generation of service models, more specific of web service choreographies. This reduces the modeling effort for existing service landscapes. Normally, IT service landscapes underlie constant evolution due to newly added or modified business functions. Quite often web services from partners or external service providers have to be integrated into the existing network. Examples are services from B2B partners or external data services, like Dun&Bradstreet to get business information and company assessments from business partners. In this case, our work helps to keep the service models up-to-date and to monitor changes in constantly changing service infrastructures.

We use *reverse engineering* techniques to extract the necessary information from the implemented web services. Main information source are the web service definitions, which provide the documentation for distributed systems and are available to all communication partners. As the information in the web service definitions is insufficient to build a complete service choreography, we concentrate on the generation of consistent skeletons, which then can be enhanced manually or enriched with business process information.

The paper continues with a description of related work. The basis features of WS-CDL and WSDL are addressed in Sections III and IV. In Section V, the transformation from web service descriptions to a choreography model is described. In Section VI, we discuss how the generated choreography skeletons can be completed. Our implementation of the transformation process is described in Section VII. In Section VIII, we conclude this paper and discuss further work.

II. Related Work

The most known languages to specify web services choreographies are Web Service Choreography Interface (WSCI,

[3]), Web Service Choreography Description Language (WS-CDL, [4]), and Ontology Web Language for Services (OWL-S, [5]). All are XML-based and support WSDL [6][7], the well-established standard to describe web services.

WSCI – sometimes considered as predecessor of WS-CDL (the last update was released in 2002) – describes the observable behavior of only one web service including temporal and logical dependencies in the message flow. WS-CDL and OWL-S are more powerful to express the collaboration of 2 parties. WS-CDL describes peer-to-peer collaborations of web services taking part in a choreography. It defines a set of agreements about ordering and constraint rules. The aim of OWL-S is to automate the discovery, invocation, composition, interoperation and monitoring of web services. A detailed comparison of the 3 languages can be found in [8].

More choreography languages, like Let's Dance and BPEL4Chor, are emerged in the last years. Let's Dance [9] is a language for modeling service interactions and their flow dependencies targeting business analysts. It is a language for high-level analysis and design. WS-CDL is a potential implementation language for Let's Dance models. BPEL4Chor [10] is an extension of Business Process Execution Language (BPEL, [11]). It adds participant behavior descriptions, i.e. control flow dependencies, the participant topology and their interconnection using message links and participant groundings, i.e. concrete configurations for data formats and port types to the standard BPEL.

To our knowledge, it is a novel approach to reengineer WSDL files to create a service model. But there are reverse approaches to generate WSDL descriptions (skeletons) from choreography models. In [12], the authors describe an approach to generate the orchestration behavior (BPEL stubs) and the necessary WSDL templates automatically from WS-CDL models. The same functionality is implemented in the visual modeling tool known as pi4SOA [13]. pi4SOA, an open-source implementation that plugs into Eclipse, is one of the few WS-CDL implementations available today.

There are many approaches, complementary to our approach, to recreate the process flow of interacting web services. *Business process mining*, or *process mining* for short, aims at the automatic construction of models explaining the behavior observed in the event log [14]. For example, based on event logs, it is possible to construct a *process model* expressed in terms of a Petri net or Event-driven Process Chain (EPC). Beside the process perspective, process mining can also focuses on the originator field (organizational perspective), to find out, which performers are involved and how are they related. The goal is either to structure the organization by classifying people in terms of roles and organizational units or to show relations between individual performers (e.g., build a social network [15]).

III. Choreography

The goal of specifying web service choreographies is composing peer-to-peer interactions between any kind of services, regardless of the programming language or the environment that hosts the service.

We have chosen to use WS-CDL for our service model, because its tight coupling to WSDL files and the recommendation of the W3C Web Services Choreography Working Group.

A WS-CDL model consists of 3 parts:

- **Collaborating parties:** describing the entities that exchange information, their roles and their relationships,
- **Collaborative behavior**: describing the physical order (message flow) of the information exchange and assigned constraints,
- **Exchanged information:** describing the type of information used in the information exchange.

A. Collaborating parties

Within a choreography, information is always exchanged between participants. A participant – described by a *participant type* – groups all the parts of the collaboration that must be implemented by the same entity. A *role type* enumerates potential behaviors of a participant within an interaction. A *channel type* is a point of collaboration between participants specifying where and how information is exchanged. Finally, a *relationship type* is used to identify the mutual obligations between participants that must be fulfilled to succeed.

B. Collaborative behavior

A *choreography* defines re-usable common rules that govern the ordering of exchanged messages. A choreography contains collections of *activities* that may be performed by one or more participants. There are three types of activities in WS-CDL, namely *control-flow* activities, *WorkUnit* activities and *basic* activities. In the first category, there are three types of activities: *Sequence*, *Parallel*, and *Choice*. These activities enclose a number of sub-activities. A *WorkUnit* activity describes the conditional and, possibly, repeated execution of an activity. The *basic* activities include *Interaction*, *NoAction*, *SilentAction*, *Assign*, and *Perform*. The most important element of WS-CDL is the *Interaction* activity that corresponds to an operation of a web service.

C. Exchanged information

InformationTypes describe the type of *variables*, *tokens* and *messages* used in the choreography. Their description at the package level makes them available to all enclosed activities. They normally refer either WSDL 1.1 message types, WSDL 2.0 schema elements or XML schema elements/types.

IV. WSDL

A web service is described by a web service description (WSDL file). Currently, there are 2 versions of the specification. WSDL 1.1 [6] is the widely accepted standard. Although WSDL 2.0 [7] is recommended by the W3C since June 2007 and promises an easier implementation, its adaption by SOAP servers, vendors and tools is still reluctant.

A WSDL 1.1 description containing six major elements (In this paper, we concentrate on WSDL 1.1, but the same information is also available in WSDL 2.0.):

- *types*, which provides data type definitions used to describe the exchanged messages.
- *message*, which represents an abstract definition of the data being transmitted.
- *portType*, which is a set of abstract *operations*, which refer to input and output messages.
- *binding*, which specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType.
- *port*, which specifies an address for a binding, thus defining a single communication endpoint.
- *service*, which is used to aggregate a set of related ports.

To generate the choreography skeletons, only the elements *message*, *portType* with *operations* and *service* are used. The name of *service* element is used to name the activities and collaborating parties in the choreography.

V. TRANSFORMATION

Our goal is the generation of a valid web service choreography (*.cdl file) out of one or several web service definitions (WSDL files). In the following, we describe the needed transformations to create the 3 essential parts of a WS choreography.

A. Collaborating parties

The *service* element of a web service description is used to generate the collaborating parties of the choreography. Each service element represents a relationship between service provider and service consumer. The name of the *service* element is used to generate the relationship type and the role types related to web service provider and consumer. In Figure 1, the graphical representation of such a relationship is shown.[1]

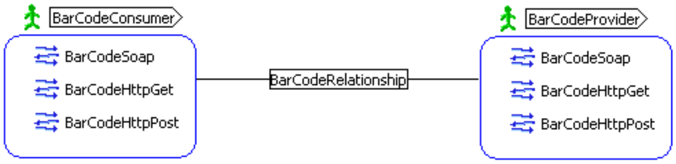


Fig. 1. Graphical Model of a WS-CDL Relationship

The role types enumerate potential observable behaviors a participant can exhibit in order to interact. This behavior corresponds to the *portTypes* (*interfaces* in WSDL 2.0) of the WSDL. In Figure 1, each role type has 3 behaviors assigned.

The optional behavior of the role types in the relationship is not filled during the transformation. In this case, all the behaviors belonging to this role type are identified as the commitment of a participant for this relationship. The right values have to be selected manually in a later stage.

The generated relationship type is assigned to the choreography (see in Figure 2) and bound to all interactions created from the WSDL file (see Section V-B).

```
<package>
  <roleType name="BarCodeProvider">
    <behavior interface="BarCodeSoap" name="BarCodeSoap"/>
    <behavior interface="BarCodeHttpGet"
       name="BarCodeHttpGet"/>
    <behavior interface="BarCodeHttpPost"
       name="BarCodeHttpPost"/>
  </roleType>
  <roleType name="BarCodeConsumer">
    <behavior interface="BarCodeSoap" name="BarCodeSoap"/>
    <behavior interface="BarCodeHttpGet"
       name="BarCodeHttpGet"/>
    <behavior interface="BarCodeHttpPost"
       name="BarCodeHttpPost"/>
  </roleType>
  <relationshipType name="BarCodeRelationship">
    <roleType typeRef="BarCodeProvider"/>
    <roleType typeRef="BarCodeConsumer"/>
  </relationshipType>
  <choreography name="BarCode" root="true">
    <relationship type="GlobalWeatherRelationship"/>
    ...
  </choreography>
</package>
```

Fig. 2. WS-CDL Relationship

Participant types are not generated. The participants are the logical entities or organizations implementing or using the web services. The necessary information is not available in the WSDLs of the web services and can be added later.

```
<informationType element="AnyType" name="BarCodeRef"/>
<token informationType="BarCodeRef" name="BarCodeRef"/>
...
<channelType action="request-respond"
     name="GenerateBarCodeChannel11"
     usage="distinct">
  <roleType typeRef="BarCodeProvider"/>
  <reference>
    <token name="BarCodeRef"/>
  </reference>
</channelType>
```

Fig. 3. WS-CDL ChannelType and related token

A channel type realizes a point of collaboration between participant types by specifying where and how information is exchanged. All our channel types have mostly the action type `request-respond`. In the rare cases, that a web service operation has no parameters or does not return anything (that means there is no input or output message assigned to the operation) the action type *respond* rsp. *request* is assigned. A channel type is named (the name is generated from the *wsdl:operation*) and then related to the role of the web service provider. In the case of several behaviors (corresponding to the *portTypes* in the WSDL), we generate equally channel types. The first of these is related to the interaction via a channel variable. A channel type gets a reference assigned to convey the information needed to contact the receiver of the message. This *reference token* is associated with an information type. As the reference information belongs to the business process, we can only generate the tokens and a dummy information type (see in Figure 3).

[1] In this and the following examples, we used a public web service for bar code generation available with its WSDL under http://www.webservicex.net/WCF/ServiceDetails.aspx?SID=40.

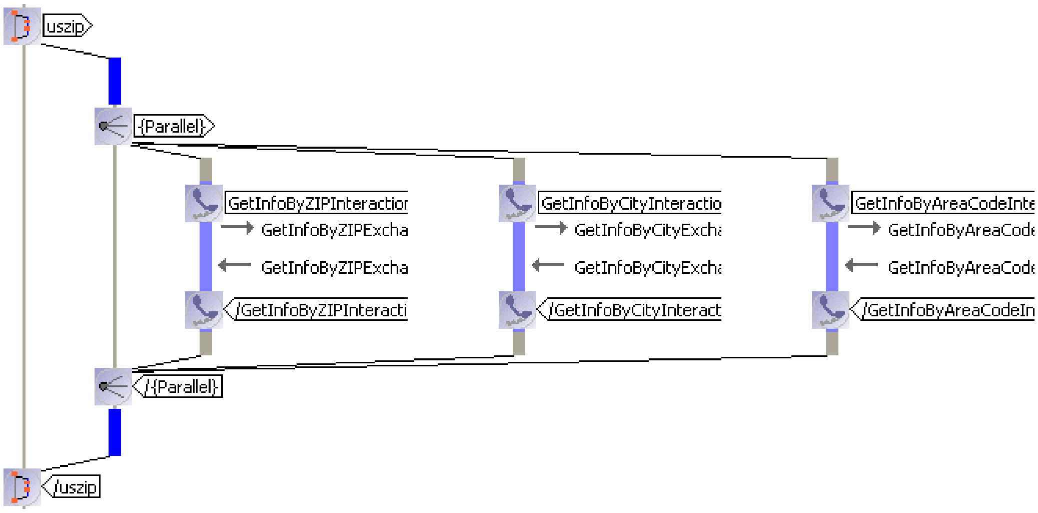


Fig. 4. Graphical Model of Parallel Activities

B. Collaborative behavior

The collaboration behavior of two partners is described by *activities* with an ordering structure. The web service definition contains no information about the order in which the different operations are called, therefore we assume an unrestricted *parallel* activity, in which one or more activities can be executed in any order or at the same time (see Figure 4[2]). By nature, all operations of a web service can be called at any time and in any order. Ordering restrictions are only given by the using business process. Our skeletons could be enriched later with the real ordering structure, either manually or automatically by integrating the information from, e.g., a business process model.

The *basic activities* generated from the WSDL files are *interaction* activities. Each web service operation corresponds to one *interaction* (see in Figure 5 a generated example interaction). The name of the web service operation is used to generate a name for the interaction and the `operation` attribute.

An interaction activity description has 3 main parts corresponding (i) to the participants involved, (ii) to the information being exchanged, and (iii) to the channel for exchanging the information.

The information about the involved participants is contained in the element *participate* and refers to the role types and relationship types described in Section V-A. The attribute `fromRoleTypeRef` refers the web service consumer role, the attribute `toRoleTypeRef` to the web service provider role, and the attribute `relationshipType` to the relationship between the two.

The exchanged information is described in the *message* part of a WSDL. The operation *input* and *output* elements connect the related messages to a WSDL operation. The information from the *message* element is transformed to the *exchange* element in the WS-CDL interaction. Depending on the operation type the `action` attribute of the exchange is generated differently: input messages → `action=request` and output message → `action=respond`.

```
<informationType type="GenerateBarCodeType"
name="GenerateBarCode"/>
<informationType type="GenerateBarCodeResponseType"'
name="GenerateBarCodeResponse"/>
...
<choreography name="BarCode" root="true">
...
  <parallel>
    <interaction
        channelVariable="GenerateBarCodeChannelVariable1"
        name="GenerateBarCodeInteraction"
        operation="ReceiveGenerateBarCode">
      <participate fromRoleTypeRef="BarCodeConsumer"
          relationshipType="BarCodeRelationship"
          toRoleTypeRef="BarCodeProvider"/>
      <exchange action="request"
          name="GenerateBarCodeExchange1"
          informationType="GenerateBarCode">
        <send/>
        <receive/>
      </exchange>
      <exchange action="respond"
          name="GenerateBarCodeExchange2"
          informationType="GenerateBarCodeResponse">
        <send/>
        <receive/>
      </exchange>
    </interaction>
    ...
  </parallel>
</choreography
```

Fig. 5. Generated WS-CDL Interaction

The attribute `informationType` of the *exchange* element refers to the information type used for the exchanged information. All information types are defined at the package level of the choreography. The elements *send* and *receive*, which contain application-dependent or state information are generated without attributes.

The channels used during the interactions are also derived

[2] To illustrate parallel activities, we have chosen a weather web service available under http://www.webservicex.net/WCF/ServiceDetails.aspx?SID=48

from the web service operations. The associated channel types are defined on package level and only referred via channel variable in the interaction (attribute `channelVariable`).

C. Exchanged information

Information types are mainly used in the *exchange* element of the interaction to describe the type of exchanged information. WS-CDL does not allow the construction of complex data types like possible in the *types* element of the WSDL. Therefore we have to generate new information types from the data type assigned to the WSDL 1.1 message parts.

The information types contain also the generated information type for the *reference tokens* (see Figure 3).

The complete mapping of web service definitions to a WS-CDL choreography is shown in Table I. Although the described transformation uses a single WSDL, the concept can equally applied to several files. Naming conflicts are anticipated by applying different namespaces for WSDL specific and generated elements.

VI. Skeleton completion

The following steps are necessary to complete the generated WS-CDL skeletons and to merge them into existing choreographies:

1) Create the participant types and assign the generated role types: Typically a participant type groups several roles. In a chain of interacting services, a participant can be the consumer of one service and the provider of another. Process mining techniques could be used to identify the roles that belong to one participant type.
2) Select the implemented behaviors in role types of the service consumer: The behaviors generated from the *portTypes* describe the different possibilities to communicate. But a specific service consumer could decide to use only a certain subset. This information belongs to the web service consumer's client application.
3) Verify the behaviors of the relationship types: Per default, all generated behaviors are committed from both sides. But for a specific combination of service consumer and service provider only a subset could be used (in accordance with the selection in 2).
4) Remove or flag unused channel types: A channel type describes the communication channel for each interaction in accordance with a selected behavior. If a web service consumer uses only a subset of the communication channels, some of the generated channel types become obsolete and can be removed.
5) Select the correct channel for each interaction: Each interaction has exactly one channel type assigned. Per default, we select always the first generated channel type. In accordance to the selected behaviors for the relationship type, the right channel type has to be selected. If the interaction can be executed on several channels the interaction has to be duplicated.
6) Fill additional information: Information from the business process and description of all components can be added.
7) Merge the choreography: The completed skeleton can now be merged with existing choreographies. This is a manual step that is not supported by the pi4SOA tool [13].
8) Establish the flow: The last step is to establish the correct ordering structure of the activities from the merged choreographies. The generated parallel activities have now to be brought in the right sequential order and to be integrated in the complex flow of activities. As information source serves mainly the knowledge about the business process. Existing workflow descriptions of the business process or process models constructed with process mining techniques could enrich the choreography.

VII. Implementation

Our transformation process consists of 3 automated steps (see Figure 6) complemented by manual activities to add additional information and to merge the skeletons with existing choreographies (see Section VI).

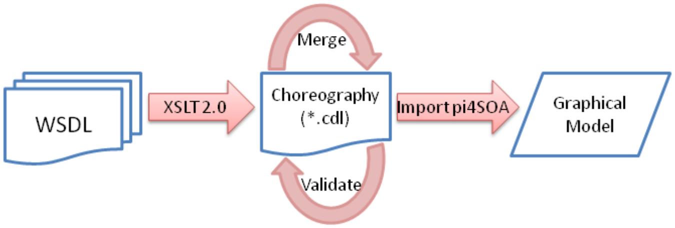


Fig. 6. Transformation flow

The transformation process starts with the selection of one or several WSDL files. A XSLT transformation transforms the input into a *.cdl file containing the choreography skeleton in WS-CDL. We use a XSLT 2.0 engine. The *.cdl file can now be manually completed and merged with existing choreographies. After this, a validation against the XML schema provided by [4] verifies the correctness of the manual editing. In the last automated step, the *.cdl file is imported into the Eclipse Plug-in pi4SOA [13]. During the import, the *.cdl file is semantically validated and transformed in a graphical model (stored as *.cdm file) that can be visually modified.

VIII. Conclusion and Future Work

We presented an approach to generate WS-CDL skeletons from web service definitions. The result is promising; we could generate complete and consistent choreographies that can be easily completed manually with the support of a graphical tool. Our approach of reengineering web service definitions facilitates the modeling process (the modeling time is reduced from several hours to a couple of minutes). It generates skeletons that describe the web service operations detailed as interactions including the cumbersome modeling and referencing of communication channels, behaviors, relationships, etc.

WSDL		WS-CDL	
Element	*Attribute*	*Element*	*Attribute*
message → part	element	informationType	name
(name=”parameters”)	element+”Type”	informationType	type
message → part	element	interaction → exchange	informationType
portType	name	roleType → behavior	name
	name	roleType → behavior	interface
portType → operation	name	channelType	name
	”request-respond”/”request”/”respond”	channelType	action
	”distinct”	channelType	usage
portType → operation	name+”Interaction”	interaction	name
	”Receive”+name	interaction	operation
	name+”ChannelVariable”	interaction	channelVariable
portType → operation → input	”request”	interaction → exchange	action
portType → operation	name+”Exchange[n]”		name
portType → operation → output	”respond”	interaction → exchange	action
portType → operation	name+”Exchange[n]”		name
portType → operation → fault	”respond”	interaction → exchange	action
	name		faultname
portType → operation	name+”Fault”		name
service	name+”Provider”	roleType	name
	name+”Consumer”	roleType	name
	name+”Relationship”	relationshipType	name
	name+”Provider”	relationshipType → roleType	typeRef
	name+”Consumer”	relationshipType → roleType	typeRef
service	name+”Relationship”	choreographie → relationship	type
service	name+”Ref”	informationType	name
	”AnyType”	informationType	element
	name+”Ref”	token	name
	name+”Ref”	token	informationType

TABLE I
WSDL TO WS-CDL MAPPING

We plan to further reduce manual modeling efforts by automated enrichments of the choreography model from existing workflow descriptions, like BPEL, or process models constructed by process mining. In [12], orchestration behavior was generated from a choreography. We will also try to reverse this process.

We consider our work as a first step in the direction of automated service model generation as a basis for constant monitoring of steadily evolving service landscapes. So far, we concentrated on the functional feature of web services. In the future, we want also consider non-functional aspects, like security and dependability. This will require extensions to the choreography languages, like WS-CDL.

ACKNOWLEDGMENT

This work was partially funded by the European Commission under the Seventh Framework Project ”PoSecCo” (IST 257129).

REFERENCES

[1] C. Peltz, “Web services orchestration and choreography,” *Computer*, vol. 36, no. 10, pp. 46–52, 2003.

[2] D. Austin, A. Barbir, E. Peters, and S. Ross-Talbot, “Web services choreography requirements 1.0,” World Wide Web Consortium, March 2004. [Online]. Available: http://www.w3.org/TR/2004/WD-ws-chor-reqs-20040311

[3] A. Arkin et al., “Web service choreography interface 1.0,” 2003. [Online]. Available: http://www.w3.org/TR/wsci/

[4] N. Kavantzas et al., “Web Services Choreography Description Language 1.0,” November 2005. [Online]. Available: http://www.w3.org/TR/ws-cdl-10/

[5] D. Martin et al., “OWL-S: Semantic Markup for Web Services,” 2004. [Online]. Available: http://www.w3.org/Submission/OWL-S/

[6] E. Christensen et al., “Web Services Description Language (WSDL) 1.1,” March 2001. [Online]. Available: http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[7] R. Chinnici et al., “Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language,” Juni 2007. [Online]. Available: http://www.w3.org/TR/wsdl20/

[8] M.-E. Cambronero, G. Daz, E. Martinez, and V. Valero, “A Comparative Study between WSCI, WS-CDL, and OWL-S.” in *ICEBE*. IEEE Computer Society, 2009, pp. 377–382.

[9] J. M. Zaha, A. P. Barros, M. Dumas, and A. H. M. ter Hofstede, “Let’s Dance: A Language for Service Behavior Modeling,” in *OTM Conferences (1)*, ser. Lecture Notes in Computer Science, vol. 4275. Springer, 2006, pp. 145–162.

[10] G. Decker, O. Kopp, F. Leymann, and M. Weske, “BPEL4Chor: Extending BPEL for Modeling Choreographies,” in *Proceedings of the IEEE International Conference on Web Services (ICWS)*.

[11] IBM, BEA Systems, Microsoft, SAP AG, and Siebel Systems, “Business process execution language for web services version 1.1,” 2007. [Online]. Available: http://www.ibm.com/developerworks/library/specification/ws-bpel/

[12] F. Rosenberg, C. Enzi, A. Michlmayr, C. Platzer, and S. Dustdar, “Integrating quality of service aspects in top-down business process development using WS-CDL and WS-BPEL,” in *EDOC*. IEEE Computer Society, 2007, pp. 15–26.

[13] pi4 Technologies Foundation, “Pi4soa,” 2007. [Online]. Available: http://pi4soa.sourgeforge.net/

[14] “Business process mining: An industrial application,” *Information Systems*, vol. 32, no. 5.

[15] W. M. van der Aalst and M. Song, “Mining social networks: Uncovering interaction patterns in business processes,” *Business Process Management*, pp. 244–260, 2004.

[16] A. Barros, M. Dumas, and P. Oaks, “A Critical Overview of the Web Services Choreography Description Language (WS-CDL),” *BPTrends*, March 2005.

R-Event: A RESTful Web Service Framework for Building Event-Driven Web

Li Li
Avaya Labs Research
Avaya Inc.
Basking Ridge, New Jersey, USA
lli5@avaya.com

Wu Chou
Avaya Labs Research
Avaya Inc.
Basking Ridge, New Jersey, USA
wuchou@avaya.com

Abstract— **As the Web is becoming a communication and collaboration platform, there is an acute need for an infrastructure to disseminate real-time events over the Web. However, such infrastructure is still seriously lacking as conventional distributed event-based systems are not designed for the Web. To address this issue, we develop a RESTful web service framework, R-Event. It represents and encapsulates the structural elements of Event-Driven Architecture (EDA) into the infrastructure of REST (Representational State Transfer), the architectural style that underlies the Web. Our approach leads to an event-driven web consisting of 4 layers of RESTful web services. The R-Event framework implements the layers that are pivotal to the event-driven web. The core component of this framework is federated topic hubs that provide services for notification publication, subscription, delivery, tracking, and linking. The advantages and applications of this approach are presented and discussed, including the important features of addressability, connectedness, dynamic topology, robustness, scalability, and efficient notifications. A prototype system for presence driven collaboration is developed and the preliminary performance tests show that the proposed approach is feasible and advantageous.**

Keywords - Web service; REST; Topic Hubs, Event-driven; EDA.

I. INTRODUCTION

The Web has undergone a rapid evolution from an informational space of static documents to a space of dynamic communication and collaboration. However, to some large extent, the Web is still a reactive informational space and information sharing is still mostly pull based. Consequently, there could be significant latency between the availability of a piece of information and the use of that information. This model of information sharing has worked well for the Web, but is becoming increasingly insufficient for new emerging applications.

In the early days of Web, changes to web content were infrequent and a user could rely on web portals, private bookmarks, or search engines to find information. However, in the era of Web 2.0, dynamic and user generated contents become increasingly popular, such as blogs, wikis, mashups, folksonomies, social networks, etc. People are demanding timely and almost instant availability of these dynamic contents, and interactive use of this information, without being overwhelmed by the information overload. This drives the Web from an informational space towards a communication and collaboration oriented environment that affects both consumer and enterprise application spaces. These new trends demand an event-driven web in which information sharing is driven by events to support the dynamic and near real-time information exchange.

Despite many existing event notification systems developed over the years, infrastructures and technologies for such an event-driven web are still seriously lacking. As the architectures, protocols, and programming languages of the existing event notification systems are developed outside of the web, there is an acute need for a unifying framework that can provide a seamless integration of these notification systems with the infrastructure of web and web based services.

For such a unifying framework, we lay our foundation on Event-Driven Architecture (EDA) [12], in which information is modeled as asynchronous events that are pushed to the interested parties as they occur. By synchronizing the states of the communicating parties through events, EDA makes real-time communication and collaboration possible. Moreover, EDA is a natural fit for the Web as both do not assume any centralized control logic. However, the current web protocols are based on client-server architecture which does not readily support EDA. Even though some recent standards and industrial efforts, such as Atom [4][5], Server-Sent Events [9], Web Sockets [10] and HTML 5 [8], introduce the notion of feed and event, they are aimed at the web browsers and human users. As far as we know, there is no research work to combine EDA and REST to enable and support federated event-driven web services.

Because EDA is an abstract architecture whereas REST has concrete protocol (HTTP), we need to first resolve how to project the elements of EDA to those entities of REST [1][2] in a consistent framework. In our approach, we found that many important features and problems in conventional event notification systems can be established and resolved efficiently in our REST based framework. For instance, the uniform interface, connectedness, and addressability of REST can apply and facilitate the discovery of notification web services. The idempotent operations and statelessness of REST can add robustness and scalability to notification web services. Furthermore, projecting EDA to REST can facilitate transformation of conventional notification systems into RESTful web services, because EDA can be viewed as a generalization of the architectural elements in those notification systems.

In our approach, the key concepts of EDA are projected into 4 layers of an event-driven web. Each layer consists of interconnected resources that collectively provide RESTful web services for applications. This projection leads to our RESTful web service architecture, R-Event that defines the notification web services for such event-driven web. To maximize the reuse and interoperability, these layers are weaved and combined through RESTful web services composition and linking. A prototype event-driven web consisting of topic hubs and topic webs is implemented to demonstrate the feasibility and advantages of this approach.

The rest of the paper is organized as follows. Section II introduces the background and related work. Section III describes the model of event-driven web. Section IV introduces the R-Event framework and its components, e.g. topic hub and topic web. Section V summarizes the advantages of this approach. Section VI is dedicated to a prototype implementation and experimental study results. Findings of this paper are summarized in Section VII.

II. RELATED WORK

REST stands for REpresentational State Transfer, the architecture style underlying the Web as described in [1] [2] [3]. The fundamental concept of REST is a resource. REST promotes the following architectural choices: 1) Addressability: each resource can be addressed by URI. 2) Connectedness: resources are linked to provide navigations. 3) Uniform Interface: all resources support a subset of the uniform interface, namely GET, PUT, DELETE and POST. GET is safe and idempotent, while PUT and DELETE are idempotent. Idempotent operations can be resubmitted if failed without corrupting resource states. 4) Statelessness: all requests to a resource contain all of information necessary to process the requests, and the servers do not need to keep any context about the requests. Stateless servers are robust and easy to scale. 5) Layering: intermediate proxies between clients and servers can be used to cache data for efficiency.

RSS [6] and Atom [4] are two data formats that describe the published resources (feeds), including news, blogs, wikis, whose contents are updated by the content providers. The content providers syndicate the feeds on their web pages for the feed readers which fetch the updates by periodically polling the feeds. However, such polling is very inefficient in general, because the timing of the updates is unpredictable. Polling too frequently may waste a lot of network bandwidth, when there is no update. On the other hand, polling too infrequently may miss some important updates and incur delay on information processing.

To address the inefficiency of poll style feed delivery, Google developed a topic based subscription protocol called PubSubHubbub [22]. In this protocol, a hub web server acts as a broker between feed publishers and subscribers. A feed publisher indicates in the feed document (Atom or RSS) its hub URL, to which a subscriber (a web server) can registers the callback URL. Whenever there is an update, a feed publisher notifies its hub which then fetches the feed and multicasts (push) it to the registered subscribers. While this protocol enables more efficient push style feed updates, it does not describe how hubs can be federated to provide a global feed update service across different web sites. The protocol defines the unsubscribe operation by overloading POST which should have been DELETE. Also the subscriptions are not modeled as addressable resources.

Many techniques have been developed over the years to address the asynchronous event delivery to the web browsers, such as Ajax, Pushlet [7], and most recently Server-Sent Events [9] and Web Sockets [10]. However, these techniques are not applicable to federated notification services where server to server relations and communication protocols are needed.

In software engineering, Publisher-Subscriber [15] or Observer [11] is a well-known software design pattern for keeping the states of cooperative components or systems synchronized by event propagation. It is widely used in event-driven programming for GUI applications. This pattern has also been standardized in several industrial efforts for distributed computing, including Java Message Service (JMS) [24], CORBA Event Service [25], CORBA Notification Service [26], which are not based on web services.

Recently, two event notification web services standards, WS-Eventing [18] and WS-Notification [19][20] are developed. However, these standards are not based on REST. Instead they are based on WSDL [27] and SOAP [28], which are messaging protocols alternative to REST [1]. WS-Topic [21] is an industrial standard to define a topic-based formalism for organizing events. However, these topics are not REST resources but are XML elements in some documents.

Recently, much attention has been given to Event-Driven Architecture (EDA) [12][16] and its interaction with Service-Oriented Architecture (SOA) [17] to enable agile and responsive business processes within enterprises. The fundamental ingredients of EDA are the following actors: event publishers that generate events, event listeners that receive events, event processors that analyze events and event reactors that respond to events. The responses may cause more events to occur, such that these actors form a closed loop.

A comprehensive review on the issues, formal properties and algorithms for the state-of-the-art event notification systems is provided in [13]. The system model of the notification services is based on an overlay network of event brokers, including those based on DHT [14]. There are two types of brokers: the inner brokers that route messages and the border brokers that interact with the event producers and listeners. A border broker provides an interface for clients to subscribe, unsubscribe, advertise and publish events. An event listener is responsible to implement a notify interface in order to receive notifications. However, none of the existing notification systems mentioned in [13] is based on RESTful web services.

III. EVENT-DRIVEN WEB

To project EDA to REST, we model the EDA concepts notification, subscription, publisher, and reactor as interconnected resources that support the uniform interface of REST. As the result, an event notification system becomes an event-driven web: a web of resources that responds to events as envisioned by EDA. There is no longer any boundary between different event notification systems as the event-driven webs are interconnected into the Web and interoperable under REST. Because an event-driven web is built on layered resources, we divide it into 4 layers as in Figure 1.

Layer 1 is a web of event publishers. They could be any resource that generates, advertises and publishes its events.

Layer 2 is a web of subscription resources that depends on Layer 1. Subscription resources define how notifications flow from the publishers to the reactors. They provide services for subscribers to manage the subscription links, such as change the filter, as well as to deliver and track the notifications.

Layer 3 is a web of notifications that depends on Layer 2. Notifications are treated both as resources and messages. This approach allows us to link notifications with relevant subscriptions and topics to facilitate information sharing and discovery. It also allows us to link notifications according to message exchange patterns and participants to capture the social interactions in communications and collaborations.

Layer 4 is a web of reactors that depends on Layer 3. The resources in this layer receive, process and react to the notifications. A reactor can be both a listener and publisher.

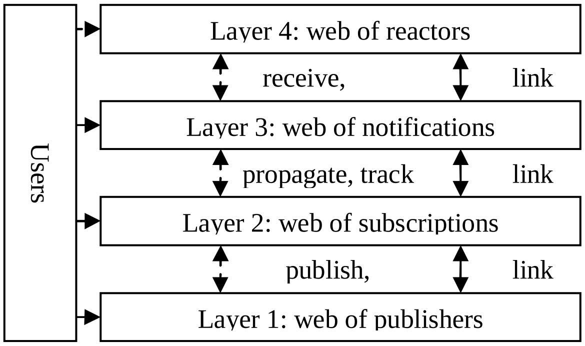

Users
Layer 4: web of reactors
receive,
link
Layer 3: web of notifications
propagate, track
link
Layer 2: web of subscriptions
publish,
link
Layer 1: web of publishers

Figure 1: Mapping EDA to layers of web

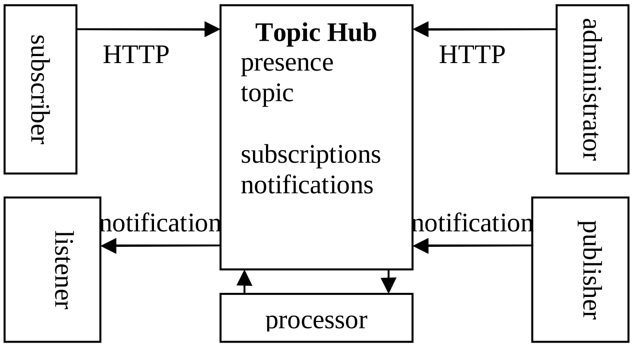


Figure 2: Topic hub resources and interactions

It should be pointed out that the resources in these layers are interconnected, such that a user can enter an event-driven web from any layer and navigate to other layers. Because layers 2 and 3 constitute services shared by publishers and reactors, they are pivotal to the event-driven web. We propose R-Event, a RESTful web service architecture to implement these two layers.

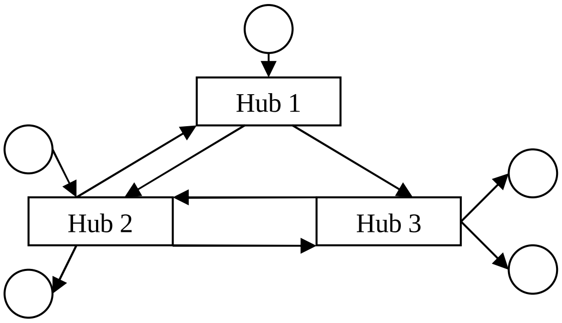


Figure 3: A topic web

IV. R-EVENT FRAMEWORK

The basic component of the R-Event framework is a topic hub that provides RESTful web services for notification publication, subscription, delivery, tracking and linking. A topic hub hosts three types of resources: topic, subscription and notification. Each hub also hosts a presence resource through which an administrator can start or shut down the services. A hub can be owned and operated by a single user or shared by a group of users. A topic hub can also invoke distributed event processors to process notifications. The high level interactions between a topic hub and its clients and servers are illustrated in Figure 2.

The topic hub is a light weight component and it can be run on any devices, including mobile phones that support HTTP protocol. It can be a servlet on a HTTP server, a standalone HTTP server, or embedded in another application. The interactions between the topic hub and its clients and servers are all based on RESTful web services.

The topic hub can also be used as a gateway that translates conventional event infrastructures into REST web services. This approach can significantly reduce the cost of web service development while ensuring the quality of services.

Because a topic hub is based on REST design, it is stateless. Consequently, a topic hub can shut down and restart safely without losing any of its services, provided that the resource states are persisted. This is especially useful when the hubs are hosted on mobile devices, which can be turned on and off. Because a topic hub is stateless, it is also scalable. We can add more topic hubs to support more clients without worrying about client session replica or affinity.

Topic hubs can be interconnected by subscriptions to provide routing services to notifications. An example topic web is illustrated in Figure 3, where topic hubs are represented as rectangles and publishers/listeners are represented by circles. The arrows indicate the subscription links on which notifications flow.

The following section describes the elements in R-Event framework in a more formal setting. In these descriptions, the left-side symbol of an equation represents a resource and the right-side tuple represents the key properties of the resource defined by this framework. Implementations can add more properties to these resources as needed.

A. *Topic Tree*

A topic tree is a set of topics organized as a tree. A topic is a resource to which events can be published and subscribed. More formally, a topic *t* has a set of events *E*, a set of children topics *C*:

$t = (E, C), C=\{ t_j \mid t_j \textit{ is a child topic of } t\}.$

B. *Subscription*

Conceptually, a subscription is a directed link from a publisher (*p*) to a listener (*l*). We extend subscription to have a set of alternative listeners (*L*), filter (*f*), expiration (*d*), and status (*u*), such as active or paused. More formally, we have:

$s = (p, L, a, f, d, u), L = \{l \mid l \textit{ is a listener}\}$

A notification *n* can propagate to one of the listeners in *L* if and only if the filter is evaluated to true, i.e. $f(n)=true$. Which listener is selected is determined by an algorithm *a*, defined by the subscriber. A simple algorithm is to try listeners according to the order they are created until one succeeds.

Subscriptions can be used to link two topics by treating them as either publisher or listener. A subscription link from a publishing topic to a listening topic is represented by two subscription resources, each as a subordinate resource of the involved topics. On the publishing topic, it is called outbound subscription (*os*), as notifications flow out of it. On the listening topic, it is called inbound subscription (*is*), as notifications flow into it. The two matching subscriptions are double linked to keep their correspondence. More formally, we have:

$os = (L, a, f, d, u), L=\{l \mid l=(t_j, is, g(is))\}, a(L)=l$

$is = (l, g(l)), l \in L$

Here each listener resource *l* consists of: 1) listening topic t_j; 2) inbound subscription *is*, and 3) the presence of *is*: $g(is)$. An inbound subscription consists of: 1) the listener *l*; and 2) the presence of *l*: $g(l)$.

C. *Topic Web*

Given a set of topic hubs $H=\{h_i\}$ where each hub hosts a set of topic trees $T(h_i)=\{t \mid t \textit{ is a topic on } h_i\}$, these topic trees form a web of topics linked by subscriptions. More formally, a topic web $W(H)$ on top of a set of hubs *H* is defined as:

$$W(H) = \bigcup_{h_i \in H} T(h_i)$$

D. *Notificatiion*

A notification is also modeled as an addressable resource that can be updated. More formally, we have:

$n_i=(o, r, b, R),$

$r=\{(t,m) \mid t \textit{ is a topic, } m \textit{ is timestamp}\},$

$R=\{n_j \mid n_j \textit{ is a response to } n_i\}$

where:

- *origin (o)*: the URI of the original notification as it was posted. Any propagated copy of the original notification inherits this property to track its origin.
- *route (r)*: the ordered set of topics (*t*) and timestamp (*m*) visited by this notification during delivery. This is used to detect loop and to expose topics to listeners.
- *about (b)*: the URI of the notification that this message responds to.
- *Responses (R)*: the set of notifications responded to this notification

The topic web may contain cycles of subscriptions. To facilitate loop detection, each notification message has a special property route, which contains a list of topics visited by the notification during propagation. Each hub checks if the current topic is in the list. If so, a loop is found and the notification will not be propagated. Otherwise, the hub appends the topic to the list and propagates the notification.

E. *Resource Design and Hub Protocols*

The key properties, interfaces and relations of the resources are depicted in the UML class diagram in Figure 4.

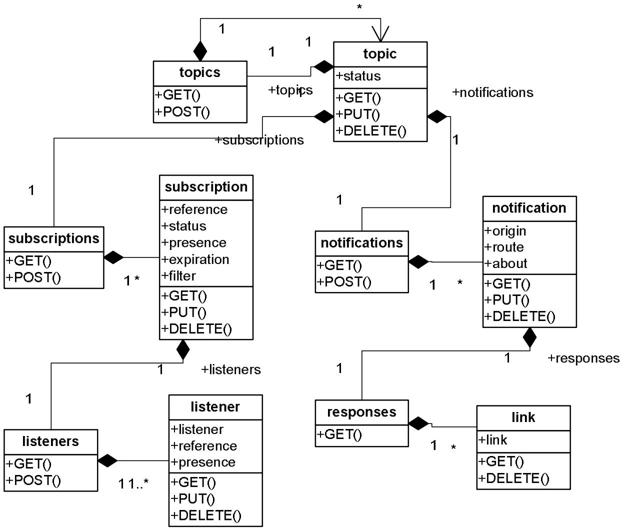


Figure 4: Main resources on topic hub

To facilitate client access, each resource on a hub is addressed by a predefined URI template that reflects the subordinate relations defined above:

- Topic *t*: */topics/{t}*;
- Child topic t_j of topic *t*: */topics/{t}/topics/{*t_j*}*;
- Subscription *s* of topic *t*: */topics/{t}/subscriptions/{s}*;
- Listener *l*: */topics/{t}/subscriptions/{s}/listeners/{l}*;
- Notification with UUID *{n}* on topic *t*: */topics/{t}/notifications/{n}*.

A subscription link from topic *ta* on hub A to topic *tb* on hub B is established by a user using a web browser as follows:

1. The user requests a subscription resource under *ta* with POST;
2. Before returning to the user, hub A creates the outbound subscription under *ta* and requests the corresponding inbound subscription under *tb* with PUT (nested inside the POST);
3. Both requests succeed and the response is returned to the user;

A notification is propagated between hubs by a user as follows:

1. The user posts a notification to a topic on hub A using POST that returns when the resource is created;
2. The notification is delivered by a scheduler to all listening topics with PUT that maintains the original UUID assigned to the notification by hub A; as the result, all the propagated notifications on different hubs can be identified by the same UUID;

This framework does not define the representations of its resources, which is left to the implementations. Different representations (media types) of the same resource are supported through HTTP content negotiation. The communications between web browsers and the topic hubs are also outside the scope of this framework, as we expected they can be addressed by the upcoming W3C standards [9].

F. Security

The communication between the topic hubs are secured using HTTPS with PKI certificates based mutual authentication. For this to work, each topic hub maintains a X.509 certificate issued by a CA (Certificate Authority) that is trusted by other hubs. It is possible or even preferable, to obtain two certificates for each topic hub: one for its client role and one for its server role, such that these two roles can be managed separately.

The communications between the topic hubs and web browsers (users) are also secured by HTTPS. In this case, the browser authenticates the topic hub certificate against its trusted CA. In return, the users authenticate themselves to the hub using registered credentials (login/password or certificate). Once a user is authenticated to a topic hub A, it employs role-based authorization model to authorize a user for his actions.

If the user wants to create a subscription link from hub A to hub B, B has to authorize A for the inbound subscription. To satisfy this condition, the user first obtains an authenticated authorization token from hub B. The user then sends this token with the subscription message to hub A. Hub A uses this token to authorize itself to hub B for the inbound subscription creation. Once hub B creates the resource, it returns an access token to hub A to authorize it for future notifications to that topic.

An alternative to the above scheme is to use the OAuth 1.0 Protocol [31] that allows a user to authorize a third-party access to his resources on a server. In this case, hub A becomes the third-party that needs to access the topic resources on hub B owned by the user. Here is how it works at a very high level: 1) the user visits hub A to create a subscription to hub B; 2) hub A obtains a request token from hub B and redirects the user to hub B to authorize it; 3) the user provides his credentials to hub B to authorize the request token and hub B redirects the user back to hub A; 4) hub A uses the authorized request token to obtain an access token from hub B and creates the inbound subscription on B.

In both approaches, the user does not have to share his credentials on hub B with hub A.

V. ADVANTAGES OF EVENT-DRIVEN WEB

On surface, the event-driven web built on top of the R-Event framework, as described in the previous section, appears similar to the broker overlay network in the conventional notification architecture [13]. However, it has the following advantages due to a REST based design.

A. Addressability and Connectedness

Unlike conventional broker overlay networks that are a closed system whose usability is prescribed by the API, all resources in a topic web are addressable and connected. Unlike in conventional broker overlay network that distinguishes between inner, border, or special rendezvous brokers, a topic web consists of homogeneous topic hubs with the same type of web services. The users can navigate and search the topic web to find the interested information using regular web browsers or crawlers. The addressability and connectedness increase the “surface area” of the web services such that the information and services in a topic web can be integrated in many useful ways beyond what is anticipated by the original design.

B. Dynamic and Flexible Topology

Unlike in conventional broker network where brokers have fixed routing tables, a topic web can be dynamically assembled and disassembled by users for different needs. Its topology can be changed on the fly as subscriptions are created and deleted and hubs join and leave the topic web. For example, a workflow system can be created where work items are propagated as notifications between users. In an emergence situation, a group of people can create an ad-hoc notification network to share alerts and keep informed. In an enterprise, a topic web about a product can be created on-demand such that alerts from field technicians can propagate to proper sales and supporting engineers in charge of the product to better serve the customers. In any case, once the task is finished, the topic web can be disassembled or removed completely. In this sense, a topic web is similar to an ad-hoc peer-to-peer network. However, a topic web is based on REST web services whereas each type of P2P network depends on its own protocols.

In conventional notification services, a broker routes all messages using one routing table. Therefore, it cannot participate in more than one routing topology. In our framework, a hub can host many topics, each having its own routing table (subscription links). As a result, a hub can simultaneously participate in many different routing networks. This gives the users the ability to simultaneously engage in different collaboration tasks using the same topic web.

C. Robustness and Scalability

Topic hubs are robust because its resource states can be persisted and restored to support temporary server shutdown or failover.

The safe and idempotent operations, as defined by HTTP 1.1 [29] also contribute to the robustness. Our framework uses nested HTTP operations where one operation calls other operations. We ensure that such a chain of operations is safe

ISBN: 978-1-61208-105-2

and idempotent by limiting how operations can be nested inside each other as follows:

nested(GET)={GET}
nested(POST)={GET,POST,PUT,DELETE}
nested(PUT)={GET,PUT,DELETE}
nested(DELETE)={GET,PUT,DELETE}

The robustness and scalability also come from the statelessness of REST design. The statelessness means that a topic hub can process any request in isolation without any previous context. By removing the need for such context, we eliminate a lot of failure conditions. In case we need to handle more client requests, we can simply add more servers and have the load balancer distribute the requests at random to the servers who share the resources. If the resource access becomes a bottleneck, we can consider duplication or partition of resources. This robustness and scalability is crucial when a topic hub serves as the gateway to large-scale notification systems.

D. *Efficient Notifications*

In conventional notification systems, notification is a message that can only be transmitted and stored. In our framework, notifications are also modeled as REST resources that provide services. Such model addresses the following issues in notification services:

Inline update: Because notifications are treated as addressable resources, a publisher can update a posted notification (using PUT) without having to create a new one. The updates will propagate over the subscription links in the topic web. This kind of inline update is more difficult to achieve in conventional notification services that treat notifications as messages.

Duplicate notification: In the topic web, a topic may receive different copies of a notification from multiple routes or multiple inline updates of the same notification, leading to potential duplicated notifications. Because our framework uses PUT to deliver notifications, the duplicate notifications to a hub become multiple updates to a resource. Therefore, we can use HTTP `ETag` and `If-None-Match` headers to efficiently detect duplicate notifications and avoid spurious alerts to the users. Compared to the solution proposed in [13], this approach solves the difficult problem without constraining the topology of the topic web.

VI. IMPLEMENTATION AND EXPERIMENTS

A prototype event notification system has been developed based on the described R-Event framework. The notification system allows users within a group to publish and subscribe presence information. Users can respond to received presence information to enable real-time collaboration. For example, when an expert becomes available through his presence notification, a manager may respond to the notification and propose a new task force be formed with the expert as the team leader. This response is propagated to the group so that interested members can set up a new workflow using the proposed topic web.

The prototype was written in Java using Restlet 1.1.4 [23]. The implementation followed the Model-View-Controller (MVC) design pattern. The Model contains the persistent data stored on disk. The Controller contains the resources and the View contains the view objects that generate XHTML pages from the XHTML templates. The topic hub stack was implemented by four Java packages, as illustrated below.

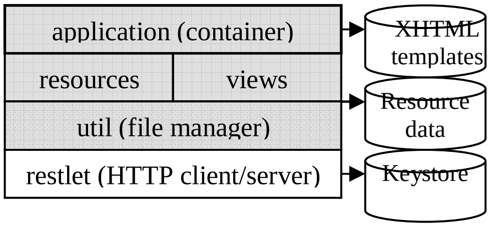


Figure 5: Topic hub stack

For this prototype, we used OpenSSL package [30] as the CA to generate certificates for the topic hubs, and Java keytool to manage the keystores for the hubs. Resources states are managed by a file manager that synchronizes the access to them. A hub used a separate thread to dispatch notifications from a queue shared by all resources. Because HTML form only supports POST and GET, we used JavaScript (XMLHttpRequest) to implement the PUT and DELETE operations for pages that update or delete resources.

Users interact with the services using web browsers (Firefox in our case). For demo purpose, the notifications were delivered to the browsers using automatic page refreshing. This is a temporary solution as our focus is on communications between hubs, instead of between browser and server. However, the R-Event framework should work with any client side technologies, such as Ajax or Server-Sent Event technologies.

We measured the performance of the prototype system in a LAN environment. The hubs were running on a Windows 2003 Server with 3GHz dual core and 2GB RAM. The performances of several key services were measured, where S means subscription, L means listener, and N means notification. The time durations for each method are recorded in the following table. The time duration includes processing the request, saving data to the disk, and assembling the resource representation.

TABLE 1: PERFORMANCE MEASURED IN MILLISECONDS

task/time	POST S	POST L	PUT S	POST N	PUT N
avg	14.1	38.9	6.2	9.5	0
std	13.7	16.8	8.0	8.1	0

The table shows that adding a listener (POST L) takes the longest time and this is expected because it is a nested operation, where

t(POST L)=processing time + network latency + t(PUT S).

The time to update a notification (PUT N) is ignorable (0 ms) and this is good news, since we use PUT to propagate notifications.

VII. CONCLUSIONS

The contributions of this paper are summarized as follows:

1. We presented an approach and a framework in which the elements in EDA can be projected and represented by REST resources, protocols and services;
2. We developed a RESTful web service framework, R-Event, based on this projection. The REST resources, protocols, services and securities are defined formally as well as described informally;
3. We illustrated that an event-driven web can be built using this framework, and discussed the advantages, including addressability, dynamic topology, robustness and scalability, etc. of this approach over conventional notification systems.
4. We developed a prototype using secure HTTP. The preliminary performance tests showed that the proposed approach is feasible and advantageous.

Our plan is to test the framework in a large scale network environment and analyze its behaviors and performance in those deployments.

REFERENCES

[1] Richardson, L. and Ruby, S., *RESTful Web Services*, O'Reilly Media, Inc. 2007.

[2] Fielding, R., *Architectural Styles and the Design of Network-based Software Architectures*, Ph.D. Dissertation, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. Last Accessed: August 27, 2010.

[3] Jacobs, I. and Walsh, N., (eds), *Architecture of the World Wide Web, Volume One*, W3C Recommendation 15 December 2004. http://www.w3.org/TR/webarch/, Last Accessed: August 27, 2010.

[4] The Atom Syndication Format, 2005, http://www.ietf.org/rfc/rfc4287.txt, Last Accessed: August 27, 2010.

[5] The Atom Publishing Protocol, 2007, http://www.ietf.org/rfc/rfc5023.txt, August 27, 2010.

[6] RSS 2.0 Specification, 2006, http://www.rssboard.org/rss-specification, Last Accessed: August 27, 2010.

[7] Pushlets, http://www.pushlets.com/, Last Accessed: August 27, 2010.

[8] HTML Working Group, 2009, http://www.w3.org/html/wg/, Last Accessed: August 27, 2010.

[9] Hickson, I. (ed), Server-Sent Events, W3C Working Draft 29 October 2009, http://www.w3.org/TR/eventsource/, Last Accessed: August 27, 2010.

[10] Hickson, I. (ed), The Web Sockets API, W3C Working Draft 29 October 2009, http://www.w3.org/TR/websockets/, Last Accessed: August 27, 2010.

[11] Gamma, E., Helm, R., Johnson, R. and Vlissides, J., *Design Patterns*, Addison-Wesley, 1995

[12] Taylor, H., Yochem, A., Phillips, L. and Martinez, F., *Event-Driven Architecture, How SOA Enables the Real-Time Enterprise*, Addison-Wesley, 2009.

[13] Mühl, G., Fiege, L. and Pietzuch, P.R., *Distributed Event-Based Systems*, Springer, 2006.

[14] Rowstron, A., Kermarrec, A.M., Castro, M. and Druschel, P., SCRIBE: The design of a large-scale event notification infrastructure, Proc. of 3rd International Workshop on Networked Group Communication, November 2001, pp 30-43.

[15] Buschmann, F. et al. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*. West Sussex, England: John Wiley & Sons Ltd., 1996.

[16] Chandy, K. M. (2006). Event-Driven Applications: Costs, Benefits and Design Approaches, Gartner Application Integration and Web Services Summit 2006, http://www.infospheres.caltech.edu/node/38, Last Accessed August 27, 2010.

[17] Michelson, B. M. (2006). Event-Driven Architecture Overview, http://soa.omg.org/Uploaded%20Docs/EDA/bda2-2-06cc.pdf, Last Accessed August 27, 2010.

[18] Davis, D., Malhotra, A., Warr, K. and Chou, W., (eds), Web Services Eventing (WS-Eventing), W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-eventing/, Last Accessed August 27, 2010.

[19] Graham, S., Hull, D., Murray, B., (eds), Web Services Base Notification 1.3 (WS-BaseNotification), OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf, Last Accessed August 27, 2010.

[20] Chappell, D. and Liu, L., (eds), Web Services Brokered Notification 1.3 (WS-BrokeredNotification), OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf, Last Accessed August 27, 2010.

[21] Vambenepe, W., Graham, S. and Biblett, P., (eds), Web Services Topics 1.3 (WS-Topics), OASIS Standard, 1 October 2006. http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf, Last Accessed August 27, 2010.

[22] Fitzpatrick, B., Slatkin, B. and Atkins, M., PubSubHubbub Core 0.2, Working Draft, 1 September 2009, http://code.google.com/p/pubsubhubbub/, Last Accessed August 27, 2010.

[23] Restlet, RESTful Web framework for Java, http://www.restlet.org/, Last Accessed August 27, 2010.

[24] JMS (2002). Java Message Service, version 1.1, 2002, http://www.oracle.com/technetwork/java/index-jsp-142945.html, Last Accessed August 27, 2010.

[25] Event Service Specification, Version 1.2, October 2004, 2004.

[26] Notification Service Specification, Version 1.1, October 2004.

[27] Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S., Web Services Description Language (WSDL 1.1), W3C Note, 15 March 2001. http://www.w3.org/TR/wsdl, Last Accessed August 27, 2010.

[28] Gudgin, M., et al, SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, 27 April 2007. http://www.w3.org/TR/soap12-part1/, Last Accessed August 27, 2010.

[29] Fielding, R., et al. Hypertext Transfer Protocol – HTTP/1.1. http://www.w3.org/Protocols/rfc2616/rfc2616.html, Last Accessed August 27, 2010.

[30] OpenSSL: http://www.openssl.org/, Last Accessed August 27, 2010.

[31] The OAuth 1.0 Protocol: http://tools.ietf.org/html/rfc5849, Last Accessed August 27, 2010.

Code Contracts for Windows Communication Foundation (WCF)

Bernhard Hollunder
Department of Computer Science
Furtwangen University of Applied Sciences
Robert-Gerwig-Platz 1, D-78120 Furtwangen, Germany
Email: hollunder@hs-furtwangen.de

***Abstract*—Code contracts allow the specification of preconditions, postconditions and invariants for .NET interfaces and classes. Code contracts not only perform constraint checking at runtime, but also provide tools for static code analysis and documentation generation. WCF is another .NET technology supporting the creation and deployment of distributed services such as Web services. Currently, WCF services cannot be equipped with code contracts. Though a combination of both technologies would bring additional expressive power to WCF and Web services, there does not exist a solution yet. In this paper, we present a novel approach that brings code contracts to WCF. Our solution combines standard technologies such as WSDL and WS-Policy. The feasibility of the approach has been demonstrated by a proof of concept implementation.**

***Keywords*-Code Contracts; Windows Communication Foundations; WCF; Web Services; WS-Policy**

I. Introduction

Code contracts [1] are a specific realization of the *design by contract* concept proposed by Bertrand Meyer. With code contracts, i) methods of .NET types can be enhanced by preconditions and postconditions, and ii) .NET types can be equipped with invariant expressions that each instance of the type has to fulfill. While the application developer specifies code contracts for interfaces and classes, it is the responsibility of the runtime environment for checking the constraints and signaling violations. Furthermore, following tools are available for code contracts:

- Static code analysis;
- Documentation generation;
- Integration into VisualStudio IDE.

From a theoretical point of view, static code checking has its limitations and cannot detect all possible contract violations. Nevertheless, it is a sophisticated instrument to help identifying common programming errors during compile time thus improving code quality at an early stage.

With the Windows Communication Foundation (WCF), service-oriented, distributed .NET applications can be developed and deployed on Windows. WCF provides a runtime environment for hosting services and enables the exposition of .NET types, i.e., Common Language Runtime (CLR) types, as distributed services. WCF employs well-known standards and specifications such as XML [2], WSDL [3], SOAP [4], and WS-Policy [5]. The Web Services Interoperability Technology (WSIT) project [6] demonstrates how to create Web services clients and implementations that interoperate between the Java platform and WCF.

When developing a WCF service one starts with the definition of an interface (e.g., in C#) that is annotated with a `ServiceContract` attribute. To implement the service, a class is created that implements the interface. During service deployment, WCF will automatically generate an interface representation in the Web Services Description Language (WSDL). WSDL is programming language independent and makes it possible to create client applications written in other programming languages (e.g., Java) and running on different platforms. With the help of tools such as `svcutil.exe` and `wsdl2java` so-called proxy classes for specific programming languages can be generated. A proxy object takes a local service invocation and forwards the request to the real service implementation on server side by exchanging so-called SOAP documents.

In order to bring code contracts to WCF, one may proceed as follows: The methods in a WCF service implementation class are extended with code contracts expressions, i.e., preconditions, postconditions, and object invariants. In fact, the compiler will not produce any errors and will create executable intermediate code. However, the code contracts constraints are completely ignored when WCF generates the WSDL description for the service. As a consequence, a WCF client application cannot profit from the code contracts attached to the service implementation. This behavior has already been observed elsewhere [7]; however, a generic solution has not been elaborated yet.

This paper presents a novel approach that combines WCF with code contracts. The strategy is as follows. When deploying a WCF service, the code contracts contained in the service implementation class are extracted. Next, code contracts constraints are represented in a programming language independent manner with WS-Policy [5]. The WS-Policy description will be attached to the service's WSDL. On service consumer side, the generation of the proxy classes is enhanced by including the code contracts expressions, which are extracted from the WSDL/WS-Policy file.

The approach has the following features:

- It combines standard technologies such as WSDL and WS-Policy to bring code contracts to WCF.
- The approach is transparent from a WCF service de-

velopment point of view. There are no special activities required.

- Code contracts are already checked on client side, including static code analysis. This may save resources during runtime because invalid service requests will not be transmitted to server side.
- The feasibility of the approach has been demonstrated by a proof of concept implementation.

The paper is structured as follows. The next section will shortly introduce the underlying technologies. Section III will recapitulate the problem description; the solution proposed will be presented in Section IV. Section V will show how to represent code contracts with WS-Policy and how to attach a WS-Policy description to a WSDL file. Then, in Section VI, the client side proxy generation will be addressed. An implementation strategy (proof of concept) will be given in Section VII. The paper will conclude with a summary and directions for future work.

II. Foundations

This section will give a brief overview on the required technologies. We start with introducing code contracts, followed by WCF and WS-Policy.

A. *Code Contracts*

With code contracts [1] additional expressivity is brought to .NET interfaces and classes by means of preconditions, postconditions, and object invariants. A method can be equipped with preconditions and postconditions. A precondition is a contract on the state of the system when a method is invoked and typically imposes constraints on parameter values. Only if the precondition is satisfied, the method is really executed; otherwise an exception is thrown. In contrast, a postcondition is evaluated when the method terminates, prior to exiting the method.

Code contracts provide a `Contract` class in the namespace `System.Diagnostics`. Static methods of `Contract` are used to express preconditions and postconditions. To give an example, consider a method `squareRoot` that should not accept negative numbers. This could be encoded as follows:

```
using System.Diagnostics.Contract;

class MyService {
  double squareRoot(double d) {
    Contract.Requires(d >= 0);
    return Math.Sqrt(d);
  }
}
```

Definition of a precondition for `squareRoot`.

The `Contract.Requires` statement defines a precondition. There is an analogous method `Contract.Ensures` that can be used to specify postconditions.

Object invariants of code contracts are conditions that should hold on each instance of a class whenever that object is visible to a client. During runtime checking, invariants are checked at the end of each public method. In order to specify an invariant for a class, an extra method is introduced that is annotated with the attribute `ContractInvariantMethod`. Within this method, the conditions are defined with the method `Contract.Invariant`.

The above sample shows how preconditions can be expressed for *classes*. As a method in an *interface* is described only by its signature and cannot have a body, code contracts foresee a simple trick to encode constraints for interface methods. The required constraints are specified in another class, which is associated with the interface.

Suppose a class `AContract` should implement code contracts for an interface `IA`. Then `IA` is annotated with the attribute `[ContractClass(typeof(AContract))]`, and `AContract` is equipped with `[ContractClassFor(typeof(IA))]`. Now the code contracts of `AContract` apply to the interface `IA`.

Note that most methods of the `Contract` class are conditionally compiled. It can be configured via symbols to which degree code contracts should be applied during compilation. Code contracts can be completely turned on (full checking) and off (all `Contract` methods are ignored); it is also possible to check only selected code contracts constraints such as preconditions.

B. *Windows Communication Foundation*

According to [8], “WCF is a software development kit for developing and deploying services on Windows.” Services are autonomous, distributed and have well-defined interfaces. An important feature of a WCF service is its location transparency: a consumer always uses a local proxy object – regardless of the location (local vs. remote) of the service implementation. The proxy object has the same interface as the service and forwards a call to the service implementation by exchanging SOAP documents. As the messages are independent of transport protocols, WCF services may communicate over different protocols such as HTTP, TCP, IPC and Web services.

The following listing shows the `squareRoot` functionality from above as a WCF service.

```
using System.ServiceModel;

[ServiceContract]
public interface IService {
  [OperationContract]
  double squareRoot(double d);
}

public class IServiceImpl : IService {
  public double squareRoot(double d) {
    return Math.Sqrt(d);
  }
}
```

`squareRoot` as a WCF service.

In order to successfully deploy a WCF service, the WCF runtime environment requires the definition of at least one endpoint. An endpoint consists of

- an *address*,
- a *binding* defining a particular communication pattern,
- a *contract* that defines the exposed services.

Endpoints are typically defined in an XML configuration file (external to the service implementation), but can also be created programmatically.

During deployment, WCF generates a WSDL interface description for the service. A WSDL description has an interchangeable, XML-based format and comprises different parts, each addressing a specific topic such as the abstract interface, the mapping onto a specific communication protocol such as HTTP, and the location of a specific WCF service implementation.

There are tools that transform WSDL descriptions into a programming language specific representation. Such a representation comprises classes for the proxy objects used by client applications. WCF delivers the tool `svcutil.exe`, which generates proxy classes for, e.g., C# together with a configuration file containing endpoint definitions. Basically, a proxy object constructs a SOAP message, which is sent to server side. A SOAP message consists of a body, containing the payload of the message (including the current parameter values of the request), and an optional header, containing additional information such as addressing or security data.

C. WS-Policy

When taking a closer look to a WSDL file one will find a couple of *policy* entries. These entries add further information to the service such as security requirements.

With the help of the WS-Policy specification [5], policies can be expressed in an interoperable manner. In general, WS-Policy is a framework for defining policies, which comprise so-called (WS-Policy) assertions. A single assertion may represent a domain-specific capability, constraint or requirement.

The following XML fragment shows how to associate a WS-Policy description to a service definition.

```
<definitions name="Service">
  <Policy wsu:Id="SamplePolicy">
    <ExactlyOne>
      <All>
        <EncryptedParts> <Body/> </EncryptedParts>
      </All>
    </ExactlyOne>
  </Policy>
  ...
  <binding name="IService" type="IService">
    <wsp:PolicyReference URI="#SamplePolicy"/>
    <operation name="squareRoot"> ... </operation>
  </binding>
  ...
</definitions>
```

WS-Policy attachment.

In the example, a WS-Policy description is attached to the `squareRoot` service via the `PolicyReference` element. The policy states that the body of the SOAP request must be encrypted. Note that the policy is part of the WSDL interface of the service. Hence, if a client does not encrypt the message body, the server would reject the request.

III. Problem Description

Suppose we want to create a WCF service with code contracts. A straightforward approach to combine both technologies would be as follows:

```
using System.ServiceModel;
using System.Diagnostics.Contract;

[ServiceContract]
public interface IService {
  [OperationContract]
  double squareRoot(double d);
}

public class IServiceImpl : IService {
  public double squareRoot(double d) {
    Contract.Requires(d >= 0);
    return Math.Sqrt(d);
  }
}
```

WCF service with code contracts.

We define a WCF service interface as usual. The code contracts for the service are encoded in the implementation class of the service.

This WCF service implementation can be successfully compiled and deployed. However, the generated WSDL description does not include any information about code contracts. In other words, code contracts are completely ignored and are not part of the WSDL interface. There are two important consequences to stress here:

1) Code contracts imposed on the service implementation are not considered when generating the proxy classes.
2) Clients of the WCF service are not aware of any code contracts. Hence, code contracts support such as static analysis and runtime checking is not available on client side.

Next we will elaborate a concept that resolves these deficits.

IV. Code Contracts and WCF: the Concept

We observe that a WCF service implementation class can use the methods of the `Contract` class according to the code contracts programming model (see Section II-A). When deploying the service, the following additional activities will be performed:

- The code contracts expressions are extracted from the WCF service implementation class and are translated into corresponding WS-Policy assertions (so-called code contracts assertions).
- The resulting WS-Policy description is included into the WSDL interface of the WCF service.

In order to exploit code contracts contained in WSDL on client side, we will enhance the generated proxy classes. This is achieved by two activities:

- Extraction of the code contracts expressions contained in the WSDL description.
- Creation of corresponding `Contract` method calls and integration into the proxy classes.

Before we will discuss each of these steps in more detail, we give some remarks. From a service development point of view, the approach is transparent. One can apply the standard programming models both for WCF and code contracts. The enhanced deployment infrastructure has the responsibility to realize the above mentioned activities. Secondly, code contracts imposed on WCF services are also available for client technologies other than .NET. Finally, due to enhanced proxy generation, code contracts tool support is available for .NET clients. Again, this enhancement is transparent for the (client) developer.

V. Code Contracts Assertions for WS-Policy

To formally represent code contracts expressions with WS-Policy, we introduce a WS-Policy assertion type, which is called `CodeContractsAssertion`.

The XML schema is defined as follows. (Note that we omit, for sake of simplicity, some attributes such as `targetNamespace`.)

```
<xsd:schema ...>
  <xsd:element name = "CodeContractsAssertion"/>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "requires"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
      <xsd:element name = "ensures"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
      <xsd:element name = "invariant"
                   type = "xsd:string"
                   maxOccurs = "unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "name"
                   type = "xs:anyURI"/>
    <xsd:attribute name = "context"
                   type = "xs:anyURI"
                   use  = "required"/>
  </xsd:complexType>
</xsd:schema>
```

XML schema for `CodeContractsAssertion`.

A `CodeContractsAssertion` has two attributes: `name` and `context`. The `context` attribute specifies the service to which the constraint applies. To be precise, the value of the context attribute is the (uniquely defined) name of the service as specified in the `binding` section of the WSDL.

The body of `CodeContractsAssertion` consists of a set of `requires`, `ensures`, and `invariant` elements. The values of these elements have the type `xsd:string` and should be valid code contracts expressions. The expressions contained in the `requires` and `ensures` elements typically refer to parameter names of the service, which are also part of the WSDL. An `invariant` expression applies to instances of data types used as service parameters. Such an expression may impose restrictions on the (public) members of the type.

Observe that code contracts expressions should only be imposed on parameters that are visible at WCF service interface level, and hence are meaningful to the client developer.

The created `CodeContractAssertions` are packaged into a WS-Policy description, which is attached via a `PolicyReference` to the service definition. The following WS-Policy description is produced for the WCF service `squareRoot` from the previous section.

```
<definitions name="Service1">
  <Policy wsu:Id="CCPolicy">
    <ExactlyOne>
      <All>
        <CodeContractsAssertion
          name="squareRootAssertion"
          context=
            "IService.squareRoot(System.Double)">
          <requires>d >= 0</requires>
        </CodeContractsAssertion>
      </All>
    </ExactlyOne>
  </Policy>
  ...
  <binding name="IService" type="IService">
    <wsp:PolicyReference URI="#CCPolicy"/>
    <operation name="squareRoot"> ... </operation>
  </binding>
</definitions>
```

Code contracts policy.

Before we will describe in Section VII how to create and attach policies for code contracts during the deployment process, we first take a look at the service consumer side.

VI. Code Contracts on Client Side

On client side, a WSDL description is compiled into proxy classes of a concrete programming language. The tool `svcutil.exe`, provided by WCF, takes a URL of a WSDL description and creates C# proxy classes. To be precise, a C# interface is generated that defines the available services, and a C# class that implements the interface. This class is instantiated by the client application to invoke a WCF service.

The standard version of `svcutil.exe` does not take into account custom WS-Policy descriptions such as code contracts policies. Hence, the generated proxy classes do not contain any code contracts expressions.

In order to include code contracts into proxy classes, one can proceed as follows. One can either modify the generated client proxy classes by incorporating the required `Contract` methods calls. For object invariants new methods will be added. Alternatively, an additional class can be created that contains only the code contracts expressions. This class will be linked via the `ContractClassFor` attribute to the proxy interface.

From a client developer point of view, the enhanced proxy classes bring the following advantages. First, a static analysis of the code contracts can be performed, which helps detecting invalid invocations of the WCF service during compile time. Second, during runtime a validation of the constraints will already be performed on client side. As a consequence, invalid service calls are not transmitted to the service implementation thus saving resources such as bandwidth and server consumption.

VII. Proof of Concept

A. *Code Contracts Extraction*

Given a WCF service implementation, we need some mechanism to obtain its preconditions, postconditions and invariants. Recently, API functions have been published to access code contracts expressions. These functions are part of the *Common Compiler Infrastructure* project [9]. We adapted the proposed visitor pattern to obtain the methods' code contracts expressions and created a function `getCodeContractsForAssembly` that computes for a given assembly a code contracts dictionary; the *key* is the full qualified name of the method and the *value* is a list of strings each representing a code contracts expression. Each expression starts either with `pre:`, `post:`, or `inv:` to indicate its type.

The function makes use of types defined directly or indirectly in the namespace `Microsoft.Cci`.

B. *Creation of WS-Policy Code Contracts Assertions*

In this step, we create an XML representation for the code contracts expressions according to WS-Policy. The XML schema for `CodeContractsAssertion` has been described in Section V.

This transformation is realized as follows: It takes the code contracts dictionary from the previous step and iterates over the keys (i.e., methods with code contracts). For each key, a corresponding `CodeContractsAssertion` is created. A single `CodeContractsAssertion` may contain several expressions. As each expression string starts with `pre:`, `post:`, or `inv:`, it is clear which of the elements `requires`, `ensures` and `invariant` are to be created in the assertion.

How to embed a set of `CodeContractsAssertions` as a WS-Policy description into a WSDL file is described next.

C. *WS-Policy Creation and Attachment*

In WCF, additional policies can be attached to a WSDL file via custom bindings. We define a custom binding that uses the `PolicyExporter` mechanism also provided by WCF. To achieve this, we implement two classes:

- `ExporterBindingElementConfigurationSection`
- `CCPolicyExporter`.

The former class is derived from the abstract WCF class `BindingElementExtensionElement`. The inherited method `CreateBindingElement` is implemented in such a way that an instance of `CCPolicyExporter` is created. `CCPolicyExporter` has `BindingElement` as super class and implements the `ExportPolicy` method, which contains the specific logic for creating code contracts policies.

The following figure visualizes the class layout.

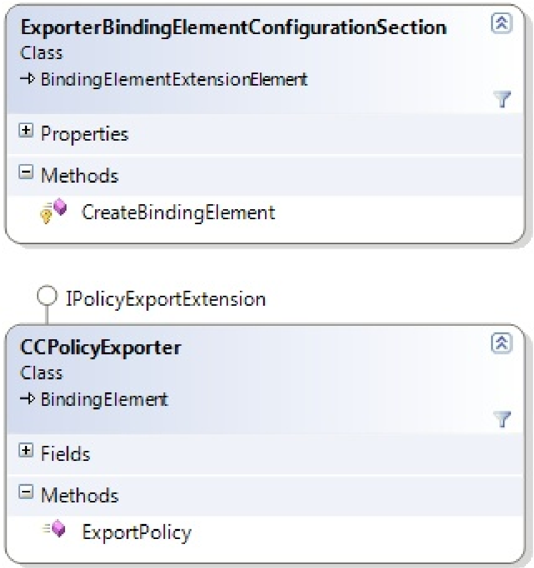


Figure 1. Class diagram for WS-Policy creation.

In our case, the `ExportPolicy` method creates the `CodeContractsAssertions` as described in the previous step. The result of this activity is an enriched WSDL description as shown in Section V.

To use the custom binding, the configuration file of the WCF service must be adapted as follows:

1) In the definition of the service endpoint, the attribute `binding` is changed to `customBinding` and the attribute `bindingConfiguration` is set to `exporterBinding`.
2) In the `bindings` section, the element `customBinding` declares `exporterBinding`.
3) The element `bindingElementExtensions` is introduced in the `extensions` section. Its `add` element specifies the assembly in which the `ExporterBindingElementConfigurationSection` class is implemented.

During deployment of the service, WCF now uses the custom binding. As a result, the generated WSDL file will include the code contracts policy.

D. Importing Code Contracts Policies

In order to invoke a service, a WCF client application requires a definition of a service endpoint. Typically, this is declared in a configuration file, similar to the one used on server side. In our case, we extend the endpoint definition by a `policyImporters` element that refers to the class `CCPolicyImporter`.

We have realized this class in the following way. It implements the WCF interface `IPolicyImporterExtension`, which declares the `ImportPolicy` method. `CCPolicyImporter` implements this method in such a way that code contracts policies referenced in the WSDL are imported. During the import, a code contracts dictionary (similar to the one on server side as described in Section VII-A) is constructed. This dictionary will be used to enhance the proxy classes, which is shown next.

E. Enhanced Proxy Generation

The tool `svcutil.exe` does not process custom policies. Hence, the standard proxy classes generated do not contain any code contracts constraints.

In our proof of concept we have realized the following approach. First, we apply `svcutil.exe` to create the standard proxy classes. In a second step, the following activities are performed:

1) Create an additional source file that contains a contract class for the proxy interface;
2) Link the generated contract class to the proxy interface.

The contract class will contain all constraints that are found in the code contracts policy. In the proof of concept, we construct the contract class as follows. Via the reflection interface we iterate on the methods of the proxy interface. For each method contained in the code contracts dictionary we create a method body with the corresponding `Contract.Requires`, `Contract.Ensures`, and `Contract.Invariant` statements. Otherwise, if the code contracts policy does not contain any constraints for the method at hand, an empty method body is generated, which means that no additional constraint is imposed to the method.

Next, we link the generated contract class to the proxy interface. This is achieved by equipping the contract class with the `ContractClassFor(typeof(...))` attribute. Finally, the proxy interface generated by `svcutil.exe` will be extended by an analogous `ContractClass(typeof(...))` attribute. This completes the generation and linkage of the code contract class with the proxy interface.

We have developed a simple tool `ccsvcutil.exe` that wraps `svcutil.exe` as described. Thus, a client developer uses `ccsvcutil.exe` to generate the client proxy infrastructure. It should be noted that the code contracts processing is transparent for the client developer – with the exception that the code contracts runtime environment and tools are now available on client side.

F. Object Invariants

In WCF, so-called *data contracts* are types that can be passed to and from the service. In addition to built-in types such as `int` and `string` user defined data contracts can be introduced be annotating a class with the `DataContract` attribute. WCF will serialize all fields that are marked with `DataMember`. To impose object invariants on data contracts one may introduce a method annotated with `ContractInvariantMethod` that contains `Contract.Invariant` statements (cf. Section II-A).

As an example consider a data contract `AddressData` with members such as street, zip and city and an object invariant method that, for example, controls the zip format. Suppose a WCF service `ChangeAddress` takes an instance of `AddressData` together with a customer id as parameters. Because `AddressData` is part of the service's signature, it has a representation as `complexType` in the WSDL. Therefore, `svcutil.exe` will generate a corresponding C# class `AddressData`, which is used by the service consumer to construct address instances. We note that this class contains only a default constructor to create "empty" instances; their members can be accessed via public getters and setters.

In order to invoke the `ChangeAddress` service, a client may proceed as follows: i) create an empty instance of `AddressData`, ii) set the specific values of the members with the public setters, and iii) pass the instance together with the customer id to the service. Unfortunately, the code contracts infrastructure on client side will report an error after the first step. This is due to the fact that the empty zip member contains an invalid value, which is recognized by the object invariant.

To overcome this problem, one needs on client side a public constructor that takes all relevant address data and constructs a properly initialized instance (which conforms to the object invariant). However, such a constructor is not generated by the standard `svcutil.exe` tool. Thus, we propose that the code contracts aware version `ccsvcutil.exe` should generate for each user defined data contract a corresponding public constructor.

On WCF service provider side this is not an issue, though. When introducing a data contract, specific constructors can be implemented by the creator of the WCF service. These constructors are available for general usage on WCF provider side.

G. Exception Handling

There are two separated code contracts runtime environments: one on WCF service consumer side and one on WCF service provider side.

As described in Section 7 of [1], code contracts support several runtime behavior alternatives. By default, a contract violation yields an "assert on contract failure". Thereafter, a user interaction is required to continue or abort program execution. While this behavior may be acceptable on client side during the development and testing phase, an analogous behavior would not be helpful on WCF provider side. Each time a violation occurs, the WCF service process requires a user interaction, which means that the server process must be observed the whole time. In general, this is not acceptable, not even during development and testing.

To remedy this problem, we disable "assert on contract failure" in the WCF service project. As a consequence, a contract violation now leads to the creation of an exception, which will be handled by the WCF runtime environment. By default, WCF returns a `FaultException` to the client indicating that something went wrong without giving detailed information. In order to embed the real reason into the exception (e.g., a "Precondition failed: d >= 0" message) the `IncludeExceptionDetailInFaults` parameter of the `ServiceBehavior` attribute in the WCF service implementation class is set to true.

On client side, standard exception handling can be applied to inspect the exception's reason.

H. Service Provider Side Development Model

To sum up, the development model that brings code contracts to WCF services is as follows:

1) Creation of a WCF service and an assembly with VisualStudio as usual, e.g. as *WCF Service Library project*.
2) Definition of a service endpoint that includes a custom binding as described in Section VII-C.
3) Deployment of the WCF service by launching the project.
4) Creation of a WCF client project as usual.
5) Invocation of `ccsvcutil.exe` to generate the enhanced proxy classes.
6) Usage of the code contracts infrastructure on client side.

VIII. Summary and Future Work

In this paper we have elaborated a concept that combines WCF with code contracts. As a consequence, WCF application developers – both on server and client side – can now profit from the additional expressive power of code contracts including runtime and tool support. It has been stressed elsewhere that there does not exist a generic solution yet.

Our novel approach exploits well-known standards such as WSDL and WS-Policy. We have described how to transform code contracts expressions contained in the WCF service into a programming language independent representation. This representation will be used to generate an enhanced client proxy infrastructure, thus allowing to evaluate the WCF service's code contracts already on client side.

We see several areas for future work. One direction is concerned with a precise definition of "WCF code contracts expressions." When defining code contracts for WCF services, only those variables should be referred that are visible to the service consumer. While service parameters are public and hence meaningful for a service consumer, it is not useful for the client when members of the service implementation class are included into the created code contracts assertions. Therefore, rules should be defined that i) characterize valid expressions (similar to the ones presented in Section 5 on contract extraction in [10]) and ii) translate the code contracts statements into corresponding WS-Policy assertions embedded into the service's WSDL description.

Additional tool support for WCF code contracts is another topic. We have shown how a custom binding can be defined such that code contracts expressions are exported to (resp. imported from) the WSDL. For a WCF developer, it would be helpful to have a specific "WCF code contracts" project type for VisualStudio that automatically introduces the required elements in the WCF configuration files.

This work is concerned with making code contracts available for a WCF client environment. Another interesting question is how a WCF service consumer developed with an alternative technology such as Java (see e.g., [11]) can process the code contracts expressions.

Acknowledgments

I would like to thank the anonymous reviewers for giving helpful comments. This work has been partly supported by the German Ministry of Education and Research (BMBF) under research contract 17N0709.

References

[1] Microsoft Corporation, "Code contracts user manual," 2009.

[2] Extensible Markup Language (XML) 1.1. http://www.w3.org/TR/xml11/.

[3] Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl/.

[4] SOAP Version 1.2. http://www.w3.org/TR/soap/.

[5] Web Services Policy 1.5 - Framework. http://www.w3.org/TR/ws-policy/.

[6] Web Services Interoperability Technology (WSIT). https://wsit.dev.java.net.

[7] Writing rock solid code with Code Contracts. http://blog.hexadecimal.se/2009/3/9, last access on 08/24/2010.

[8] J. Löwy, *Programming WCF Services*. O'Reilly, 2007.

[9] Common Compiler Infrastructure: Code Model and AST API. http://cciast.codeplex.com/, last access on 08/24/2010.

[10] M. Barnett, M. Fahndrich, and F. Logozzo, "Embedded contract languages," in *ACM SAC - OOPS*. Association for Computing Machinery, 2010.

[11] E. Hewitt, *Java SOA Cookbook*. O'Reilly, 2009.

Automated Service Evolution

Dynamic Version Coordination Between Client and Server

Virginia Smith
Business Technology Optimization
HP Software
Roseville, CA, USA
virginia.smith@hp.com

Bryan Murray
Business Technology Optimization
HP Software
Bellevue, WA, USA
bryan.murray@hp.com

***Abstract*— While client/server integrations may be loosely coupled so that the evolution of the service endpoints occurs with minimal impact on backward compatibility, installing and configuring application upgrades to take advantage of new application functionality is still painful for customers and involves manual work by administrators. Coordinating changes in version between client and server has traditionally been done using either a central registry or through manual configuration, both of which can be error prone. The authors propose that clients and servers be aware of the versions they consume and provide and that they coordinate between themselves to adapt dynamically to new versions.**

Keywords - automation, versioning, web service, REST client/server, evolution

I. Introduction

Upgrading deployed software has been an ongoing problem in the software industry. Much of the research has focused on this problem in several main areas. One area of focus is adaptive software where a software system can adapt itself in response to specific internal or external conditions as detailed in [10]. This research focuses on the software system itself (the service) and does not address the problems that occur in the client/server communication when a service is upgraded. Another area of research focuses on maintaining backwards compatibility to eliminate client problems after the service upgrade, usually through the addition of a new component. Some examples are [5] which uses adapters and [4] which uses an interface monitoring component.

This version evolution problem is even more acute today as more and more functionality is deployed as web services where the client and server are independently controlled. In addition, in many enterprise deployments, multiple versions of client and services must coexist due to business requirements or software supplier constraints. Services must evolve to handle new customer requirements and clients want to know when a service is upgraded so they can take advantage of new functionality immediately without waiting for a manual configuration. To handle this dynamic environment, clients must be able to deal with multiple service versions and services must be able to deal with multiple client versions. The authors propose a method of dynamic negotiation between client and server that enables them to adapt to this kind of deployment environment.

We showcase our proposed solution using the Representation State Transfer (REST) [2] client/server architectural style as defined by Roy Fielding in his doctoral dissertation. One of the key benefits of the REST architectural style is that the client and server become much more loosely coupled than was possible using the operation-oriented approach. A RESTful architecture is being adopted by many applications to enable easy and consistent integration development. While RESTful application integrations may be loosely coupled and, therefore, the evolution of the service endpoints occurs with minimal impact on backward compatibility, installing and configuring application upgrades to take advantage of enhanced application functionality is still painful for customers and involves manual work by administrators. The authors demonstrate dynamic version coordination between client and server using a method that enables RESTful integration participants to seamlessly configure themselves to use a new endpoint version as the client is updated and/or a new service version becomes available.

All services, even those written using the REST architectural style, will need to modify their data models at some point. With care, a client and service can continue to work even with many data model changes, as long as those changes are backwards compatible. The W3C TAG draft document [8] on versioning languages addresses the issue of maintaining compatibility between versions of a language and provides insight into a number of design patterns for constructing extensible languages and defining a language versioning strategy. These strategies help to ensure compatibility between versions of a language and thus between a service and its clients.

However, even when there is language compatibility between a service and its clients, there are reasons that may prompt a server to move to a new version. Bug fixes are one scenario. Another scenario is when there is a functionality change in the content of a client request to the server. For example, a server might support new query parameters. The client can then add new logic to communicate with the server using the new functionality. A third scenario occurs when there is an expectation of some new action related to the resources controlled by the service. For example, a resource has a state attribute of 'on' or 'off' but a new state is introduced such as 'standby'. The new language version may be compatible with the old version but there is new functionality represented by this change. The service supports the new state attribute with some specific actions. In fact, the service might require that clients make use of the new state in a new version of the service. In these scenarios, the client is not satisfied with simply maintaining language compatibility with the server. The client is interested in using the latest version of a service to take advantage of new functionality or new language elements. Therefore, even with a well-defined versioning strategy, there is a need to address the ease of migration of a service and its clients to newer versions when that migration is desired.

The remainder of this paper is organized as follows. Section II defines the terms used throughout the paper and presents an example that is used to demonstrate the concepts. Section III describes the problem that occurs when individual applications are combined to deliver an enhanced solution to the customer. Sections IV and V present an approach to solving this problem using common REST technologies. Section VI offers suggestions for implementing version evolution in non-RESTful environments. Finally, the paper concludes with thoughts on the general applicability of the approach.

II. TERMS

An **integration** is a point of communication between two applications for the purpose of sharing resources. For example, the operations management application can open a ticket in the help desk application when an alarm is raised. The business impact analysis application can add additional relevant information to the help desk ticket to help the operator triage the problem.

There are two parties to every integration point, a **client** and a **server**. In REST architecture, these are defined as the two main connector types. “The essential difference between the two is that a client initiates communication by making a request, whereas a server listens for connections and responds to requests in order to supply access to its services. A component may include both client and server connectors.” [2]

An **endpoint** is the implementation of a service interface. In a RESTful web service, it is defined by a set of related URLs and the HTTP methods that are valid for those URLs. The endpoint implementation acts as the server in an integration. The term **service** is also used here to mean the service endpoint.

III. ASYNCHRONOUS MANUAL CONFIGURATION

While loosely coupled integrations allow for the client and server to evolve independently, upgrading to a new version can cause integration configuration problems. With multiple versions of the client and multiple versions of the server available in the field, it is necessary to configure which version of the server a client connects to. This is often a manual process that is performed by administrators and is error prone. Making the matter worse is that the applications participating in integrations rarely follow the same upgrade timeline. When and how should an administrator configure a new version of an integration (e.g., reconfigure the endpoint URLs) and what happens if there are multiple application versions that exist in a customer’s environment?

Consider the example of an IT management solution. This solution is enabled through integrations between four related applications as shown in Figure 1. This product suite solution is now being upgraded to enable additional new collaboration between the applications. Each application must implement its part of this new collaboration functionality. The products have the following schedules for the release of the version that will support the enhanced solution:

- Product A: version 5 is already released.
- Product B: version 6 will release in 2 months.
- Product C: version 7 will release in 3 months.
- Product D: version 2 will release in 6 months.

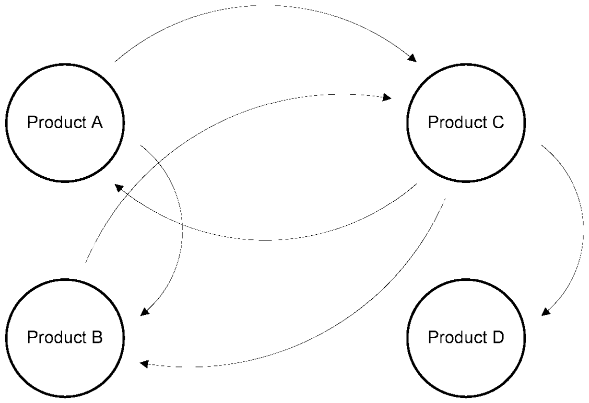


Figure 1. A 4-product solution showing the integration points.

While it is difficult to synchronize the release timelines of any two products developed independently, it is even more difficult to synchronize the upgrade of different applications in a customers’ environment where there may be sets of constraints by users of those applications on availability, risk of change or introducing incompatibilities, etc. The administrator must not only install the new version of the application, but also reconfigure new versions of all of the integrations between that application and other applications. Some applications may or may not be ready for a new version of an integration, making the upgrade process error prone. This results in customer frustration and increased support calls. As a result, customers are sometimes very slow to upgrade their applications. This can have a detrimental impact on the ability to bring end-to-end solution improvements to customers.

The authors propose an approach that enables dynamic version coordination between client and server. The approach defines how a client can automatically discover when a server is upgraded and how the client can reconfigure itself to use the new version of the server without requiring either a central registry or manual intervention by an administrator.

ISBN: 978-1-61208-105-2

IV. AUTOMATED EVOLUTION

Through careful orchestration of the messages exchanged and the incorporation of version information in the messages, integrated applications can maintain their relationship automatically, always using the latest version shared by the client and server. This significantly improves the decoupling of individual product releases for integrated applications and makes the deployment of enhanced integrations and solutions a simpler, more automated process.

The authors' proposal is composed of two behaviors: Discovery and Notification. Discovery is used to assure that when a client starts, it is using the latest version of the service that it supports. Notification is used to inform the client of an available new service version when the client has been running and the service was asynchronously updated.

The Discovery behavior defines how a service advertises its capabilities, and how a client approaches using the service. The key elements of Discovery are:

- Each application that provides services to integrating partners makes available to the client information about what versions it currently supports and how to access each version.
- Clients are expected to access this information when they begin using the service and select the appropriate version of the service to access the resources of interest.

The Notification behavior defines a process for independently updating a client or service without updating, manually reconfiguring, or restarting other applications. For example, it allows for installing a new version of a service (in parallel with an existing version of the service) without requiring a manual reconfiguration of an application that is a client of that service. The Notification behavior defines how a service informs a client that a newer version of the service is available, and how a client behaves upon receiving such a notification. The key elements of Notification are:

- When a server receives a message from a client that is not the latest version, the server includes a notification that a new version is available as part of the response to the client (along with the location of the new version). The location of the version information document is also included in the response.
- When a client receives a notification indicating a newer version, it may follow the link to the latest version information document and discover the server versions there or it can follow the link to the requested resource using the latest version of the service. If the client does not support a newer version, it ignores the new version notification.

There are significant benefits to this approach. No endpoint registry needs to be maintained, no periodic checking for new application versions needs to occur, and no manual configuration of the upgraded client or server application is necessary. The version information document is always up to date and the binding of the server and client occurs at the last possible time. Clients can seamlessly configure themselves to use a new server version as the client is updated and the server version becomes available.

V. IMPLEMENTING DISCOVERY AND NOTIFICATION BEHAVIORS

This section will map the proposed dynamic version coordination approach to an implementation suitable for use in RESTful services.

A. Discovery

In order to address the Discovery behavior described in Section IV, the authors propose using the Atom Publishing Protocol (APP) Service Document to advertise the versions that a service currently supports. The APP specification [3] defines a Service Document to be a set of Workspaces, each containing references to a set of Collections. The APP specification does not attach any particular meaning to a Workspace.

The authors define a new element, `version`, to be a child of the APP `workspace` element. The `workspace` element already groups collection references into a cohesive set. The addition of the `version` element adds the concept of version to a workspace. Multiple workspaces may have the same value for the `title` element, as long as the value of `version` is different for the two workspaces. An example of a Service Document using the new `version` element is shown below. The example shows how the workspace grouping is used to advertise two versions of a service. Note the difference in the URLs for the collections.

```
<?xml version="1.0" encoding="utf-8"?>
<service xmlns="http://www.w3.org/2007/app"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns:v="urn:x-auto-version:version">
  <workspace>
    <atom:title>Help Desk Svc</atom:title>
    <v:version>1</v:version>
    <collection
        href="http://example.org/incidents">
      <atom:title>Incidents</atom:title>
      ...
    </collection>
  </workspace>
  <workspace>
    <atom:title>Help Desk Svc</atom:title>
    <v:version>2</v:version>
    <collection
     href="http://example.org/v2/incidents">
      <atom:title>Incidents</atom:title>
      ...
    </collection>
    <collection
     href="http://example.org/v2/operators">
      <atom:title>Operators</atom:title>
      ...
    </collection>
    ...
  </workspace>
  ...
</service>
```

The `version` element allows a service to advertise multiple versions of its endpoint(s) with links to resource collections as a cohesive set for a given version. A service supporting multiple workspaces before adding a second version can still support multiple versioned workspaces. A client can easily determine whether the service supports multiple versions by searching for the `version` elements and selecting the workspace(s) to use based on the available versions and the versions supported by the client.

B. Notification

In order to address the Notification behavior described in Section IV, the HTTP Link header [7] is used in response messages. Use of the HTTP Link header allows the Notification behavior to work with any media type. The Link header includes a URI reference and an indication of how the resource indicated by the URI reference is related to the resource in the response body. Two relation types are used in the Notification behavior. First, the `service` relation defined in the Web Linking specification [7] is used to identify the location of the Service Document. Second, a new relation is defined to indicate the location of a resource using the latest version of the service: `urn:x-auto-version:new-service-version`.

A link with the `service` relation can be included in any response message from a service. The link must be included when a newer version of the service is available. The URI reference in a service link identifies the location of the APP Service Document used for the Discovery behavior.

A link with the `new-service-version` relation indicates that the service provides a newer version than the client was accessing in the request message. The resource referenced by the URI is the resource the client requested, but in a newer version of the service. The `version` link parameter is also defined for the `new-service-version` relation. This parameter indicates the new version of the service and must contain the latest version supported by the service. The client may also access the Service Document for information on how to access other versions of the service if appropriate. The response containing the `new-service-version` link will use the same version that the client used in the request. A `new-service-version` link must not be included in a response message unless the service supports a newer version.

An example of how the links are used in a response sent from a service is shown below. The example shows how the service link is used to indicate the location of the service document, and how the new-service-version link indicates the location of the requested resource using a newer version of the service.

```
Link: <http://example.org>; rel="service"
Link: <http://example.org/v2/incidents>;
  rel=
   "urn:x-auto-version:new-service-version";
  version="2"
```

The `new-service-version` and `service` link relation types allow a service to notify a client that a newer version of the service is available. A service indicates the location of both a newer version of the referenced resource and the service's Service Document. A client can use the referenced Service Document to find the available versions and determine which version is appropriate. The client also has access to the newest version of the resource it was accessing. This document does not define any meaning for the `new-service-version` and `service` link relations in requests sent from the client to the service.

C. Service Actions

When a service deployment is updated to support a new version, it is important for the service to continue supporting one or more older versions to allow for clients that cannot be upgraded at the same time and preserve loose coupling between a service and its clients. The service provides an updated Service Document advertising the new version of the service and one or more supported older versions. In the case where a service receives a request sent to an older version, it notifies the client of the availability of the newer version.

It is not difficult for the service to support a newer version of the Service Document. All requests, regardless of version, will return the same Service Document listing all of the available versions. The Service Document for the service should always be at the same location. In any case, the `service` link relation will always indicate the location of the Service Document.

Support for the notification to clients when a newer version is available requires that older versions of a service are aware that a newer version is available. This awareness only needs to extend to the ability to add the HTTP Link header to the response where the request used an older version.

There are four use cases that occur in a multi-version environment. The following discussion will use version 1 to mean an older version of the application (client or service), and will use version 2 to mean a newer version of the application. With respect to the client, the version is intended to indicate which version(s) of the service the client understands. That is, a version 1 client understands only version 1 of the service and a version 2 client understands version 2 of the service and also supports version 1 of the service.

a) Version 1 service receives version 1 client request
b) Version 1 service receives version 2 client request
c) Version 2 service receives version 1 client request
d) Version 2 service receives version 2 client request

The cases where a service receives a message from a client matching its version (use cases a and d above) are not interesting and will not be discussed here. In addition, a well-behaved client will only send messages to a service that the service will understand because the client has performed a discovery of supported versions of the service. Thus, use case b will not occur in a well-behaved environment.

The main concern is with an 'evolving state' where the client and server are out of sync with respect to their versions (use case c). When a version 2 service receives a message from a version 1 client, it means the service has been upgraded to a newer version while the client remains at an older version. In this situation, the service will generate a response using the same version that the client used, but will add the HTTP Link header indicating that a newer version is available and where to find it.

D. Client Actions

A well-behaved client will initially start from the Service Document for a service in order to find the resources in which it is interested. A client is given the address of the Service Document through configuration performed when the client is first deployed. The client will choose the appropriate version of the service from the Service Document.

Version selection is done by reviewing all of the `workspace` elements within the Service Document, noting their respective versions based on the value of the `version` element. The client will choose to use the endpoints in the workspace(s) with the highest version that is less than or equal to the highest version the client understands. Once the workspace(s) have been selected, the client can proceed with the discovery of resources.

If the installed client is version 1, the client will choose the version 1 workspace(s). The client may receive newer version notifications from version 2 of the service but will continue to use version 1 since that is the latest version it understands. In the case where the installed client is version 2 but the service is version 1, the only workspace available to the client will be version 1. (Normally, a client will continue to support several versions of the service for some period of time in order to handle this use case.) Later, after the service is upgraded to version 2, the next time the client accesses the service, it will receive a notification in the response that indicates a newer version of the service is available.

A client that is using an older version than the highest it can understand should check every response from the service to see if it includes the HTTP Link header indicating a newer version. Just because a new version of the service was deployed, does not mean that the clients of that server must be restarted. When a client receives the notification of a newer version it should either start the discovery process over, or update its cache for the resource location and continue on using the newer version for the resource.

As described for services, there are four use cases that will be examined from the client's point of view. As mentioned previously, the following discussion will use version 1 to mean an older version of the application, and will use version 2 to mean a newer version of the application. With respect to the client, the version is intended to indicate which version(s) of the service the client understands. That is, a version 1 client understands only version 1 of the service and a version 2 client understands version 2 of the service and also supports version 1 of the service.

e) Version 1 client receives version 1 server response
f) Version 1 client receives version 2 server response
g) Version 2 client receives version 1 server response
h) Version 2 client receives version 2 server response

As with services, the case where a client receives a response from a service with the matching version (use cases a and d above) are not interesting and will not be discussed here.

When a version 2 service sends a response to a version 1 client, it must add the HTTP Link header (defined in Section V.B) as the notification to the client that a newer version is available. In this case the client is not capable of understanding the newer version and will ignore the notification. It is possible that the client will not even be checking for newer versions if it is already using the highest version it understands.

If a client receives a response message containing the HTTP Link header indicating a newer version is available and it supports a later version of the server than it is currently using, it should use the links to begin using the newer version.

E. Example Scenario

The following scenario demonstrates the Discovery and Notification behaviors. This scenario occurs when the client is upgraded to a newer version before the service. The sequence of steps involved is shown in Figure 2. The opposite scenario where the service is upgraded first is very similar and involves the same actions although in a different order.

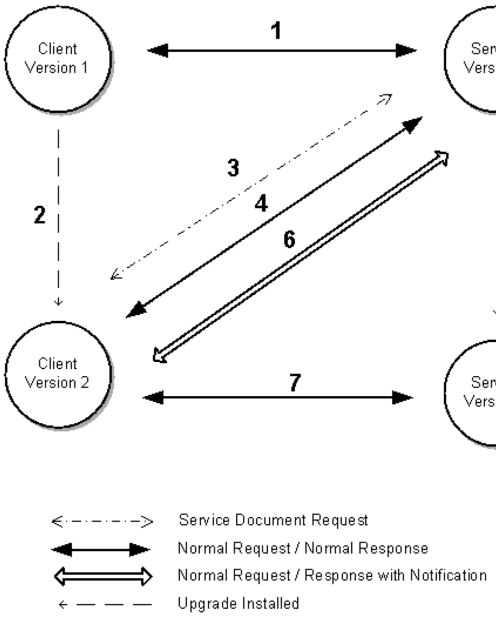


Figure 2. Steps to evolve client and service to new version

The actions that occur at each step are the following:

Both the service and its client are at version 1 and continuously execute the normal request/response cycle.

The client is upgraded to version 2 although it still supports version 1 for ease of migration. (The service is unaware of this upgrade.)

As part of its normal startup, the client requests the service's Service Document. The client selects version 1 of the service. In this scenario, version 1 is the only version currently supported by the service and therefore is the only version available in the Service Document.

The client and the service continue to execute the normal request/response cycle as if both were at version 1.

The service is upgraded to version 2 although it still supports version 1. (The client is unaware of this upgrade.)

The next time the client sends a request to this service (still using version 1), the service sends back the normal response but this time it includes a notification to the client that there is a later version of the service available.

Upon receiving a response that includes a newer version notification, the client automatically begins to use version 2 of the service from this point forward.

F. *Performance Impact*

A demonstration of the described research has been coded as APP-based client and service, and a pilot project within HP has been initiated. The impact of the Discovery behavior on message size and processing time is minimal since Discovery is used only when a client connects to a service for the first time or after a service notifies a client of a new service version. These are infrequent events.

The impact of the Notification behavior on message size and processing time is more important since it can affect most messages between the client and service. The demonstration service uses different URLs for different versions, always sends the link to the service document, and conditionally sends the link to the new resource version. The message size is increased by the size of these two HTTP headers. The processing time is increased by the time to write the headers.

The processing time impact for every message is the check for the presence of the notification. If present the Discovery behavior is initiated. The demonstration client checks for notifications only when it is not operating with its most recent version, otherwise the client can ignore them thus incurring no extra processing time.

The minimal change in processing time and message size is deemed a good trade-off for the significantly reduced manual configuration normally done for version changes of services and their clients.

VI. EXTENDING THIS APPROACH TO OTHER TYPES OF SERVICES

The previous section describes an implementation of the proposed approach to automated service evolution that can be easily applied to RESTful services and clients. There are other alternatives that could be used in this same context. For example, the HTTP Link header is explicitly defined as semantically equivalent to an HTML `LINK` element [8] or `atom:link` elements in an Atom feed [4]. The advantage of choosing the HTTP Link header is that it can be used to provide version notifications independent of the media type used for the data in the response body.

There are some types of services, for example SOAP-based services, where it is less obvious how to apply the proposed approach to enable independent version evolution of applications. It is still necessary to provide both Discovery and Notification behaviors.

Using a non-RESTful architecture (such as SOAP), applications can still perform the Discovery behavior by using the APP Service Document as described above. However, other approaches may be more natural for the environment. For instance, versions of services could be advertised in a registry. The basic actions that the client goes through for discovery are similar to the approach described above, just using a different source for the version information.

The HTTP Link header will continue to work for the Notification behavior in a non-RESTful architecture as long as HTTP is used as a transport. When HTTP is not used as a transport, it will be necessary to find a way to convey the availability of a newer version from the service to the client either as part of the transport or in the body of the message itself. For instance, XML-based messages could include an optional element or attribute in the message body to provide the notification.

VII. CONCLUSION

The original motivation for this solution was the integration of HP enterprise management products in order to bring more comprehensive, end-to-end, and synergistic IT management solutions to our customers. However, the authors feel that this approach provides value in general situations where the client and server are under the control of different organizations as is the case for many web services. This approach enables a seamless, automated evolution of web services and their clients.

REFERENCES

[1] T. Berners-Lee, R. T. Fielding, and H. F. Nielsen, “Hypertext transfer protocol—HTTP/1.0”, IETF RFC 2616, May 1996.

[2] R. T. Fielding, “Architectural styles and the design of network-based software architectures”, PhD Dissertation. Dept. of Information and Computer Science, University of California, Irvine, 2000. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (last access October 27, 2010).

[3] J. Gregorio and B. de hOra, “Atom Publishing Protocol”, IETF RFC 5023, October 2007.

[4] B. Kalali , P. Alencar , D. Cowan, “A service-oriented monitoring registry”, Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research, October, 2003.

[5] P. Kaminski , H. Müller , M. Litoiu, “A design for adaptive web service evolution”, Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems, May, 2006.

[6] M. Nottingham and R. Sayre, “Atom Syndication Format”, IETF RFC 4287, December 2005.

[7] M. Nottingham, “ Web Linking”, IETF Draft, January 2010.

[8] D. Orchard, ed., Extending and Versioning Languages: Compatibility Strategies, World Wide Web Consortium, September 2008.

[9] D. Raggett, A. Le Hors, I. Jacobs, eds., HTML 4.01 Specification, W3C Recommendation 24, December 1999.

[10] M. Salehie and L. Tahvildari. Self-adaptive software: Landscape and research challenges”, ACM Transactions on Autonomous and Adaptive Systems, May 2009.

An Architecture to Measure QoS Compliance in SOA Infrastructures

Alexander Wahl   Ahmed Al-Moayed   Bernhard Hollunder
Department of Computer Science
Hochschule Furtwangen University
Furtwangen, Germany
alexander.wahl@hs-furtwangen.de   *ahmed.almoayed@hs-furtwangen.de*   *bernhard.hollunder@hs-furtwangen.de*

***Abstract*—In the last couple of years Service Oriented Architecture (SOA) has gained in importance and became widely used. With increased acceptance the demand of non-functional requirements, so-called Quality of Service (QoS) attributes, arose. QoS attributes were applied to SOA environments, resulting in QoS-aware SOAs. Within the QoS-aware SOAs, compliance to the desired QoS in general is not easy to measure. In this work, we offer a solution architecture to measure actual data that relate to QoS attributes. Further, these data are compared to their target state. The aim is i) to evaluate compliance of the entire QoS-aware SOA to the desired QoS attributes and ii) to start suitable activities. In a proof of concept a solution architecture, based on the technique of Complex Event Processing, is implemented. Within this proof of concept selected QoS attributes are applied and compliance to the SOA is measured.**

***Keywords*-Service Oriented Architecture; Quality of Service; QoS Attributes; Complex Event Processing;**

I. Introduction

Service Oriented Architectures (SOA) are a design paradigm to compose and structure loosely coupled components to form distributed applications. SOA offers a way to map business processes from the business domain to the technical domain of computer systems. After a business process is analyzed and its single activities are identified, the individual activities are mapped to the technical domain by implementing corresponding services. To execute a business process, the services are called in corresponding sequences.

Web Services (WS) are the predominant technology to realize the services of a SOA. They are used to implement the functional aspects of business processes, which in brief define the input/output behavior of a component. Additional, in many business domains it is crucial to fulfill non-functional requirements. A non-functional requirement, or Quality of Service (QoS) attribute, specifies *how* a component is supposed to behave. Examples for QoS attributes are robustness, security, performance, scalability and accounting. More detailed descriptions of QoS attributes in SOA can be found in [1] and in [2]. Within a SOA equipped with QoS attributes, which we call QoS-aware SOA, desired QoS attributes are described in a formal manner. Therefore a policy language is used typically. These so-called service policies define the target state of the desired QoS attributes.

The crucial need to fulfill non-functional requirements is reflected by QoS attributes applied to SOA. For example, consider security aspects, like integrity and confidentiality, which are applied to Web Services using WS-Security [4]. When implementing a SOA from scratch, QoS attributes can be designed from the beginning. But many SOAs are grown, which means that they expanded over time. Such SOAs often integrate existing legacy applications and enhance them by QoS attributes they were not equipped with before. Also QoS attributes may have changed several times. So how can compliance to QoS attributes be measured? For example, assume a SOA of high complexity that has grown over time. For this SOA, a roles and rights model is specified. During runtime violations to the roles and rights specification are observed. In consequence, the entities that caused the violations are to be fixed. Further, the SOA is to be analyzed on compliance of all entities to the given roles and rights policy. But how can such an analysis be performed efficiently?

An efficient solution is to monitor and analyze the QoS attributes of a SOA at dedicated measurement points. Monitoring approaches were already elaborated in several publications. Berbner et al. [5] selected Web Services (WS) based on QoS properties guaranteed by Service Level Agreement (SLA). They ensured compliance to a given SLA using a monitoring component, which was not described in detail. Zeng at al. [6] introduced a high-performance QoS monitoring system. In their work they focus on service monitoring architecture and QoS metric computation. Artaiam and Senivongse [7] described a JMX-based monitoring extension of application servers for selected QoS attributes. Michlmayr et al. [8] integrated existing client-side and server-side monitoring approaches using Complex Event Processing (CEP). Finally, Oriol et al. [9] described the monitoring of adaptable SOA. We will go into more detail on these approaches and the differences to our work later in Section VI.

In this work, we propose a solution architecture that evaluates compliance of a QoS-aware SOA to the desired QoS attributes. The solution architecture monitors the SOA at dedicated measurement points. The thereby collected actual data are filtered according to a filter policy and compared to target states specified in the service policies of the SOA.

A SOA application landscape consists of several compo-

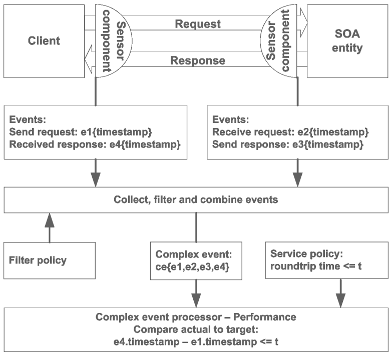

Client
Sensor component
Request
Response
Sensor component
SOA entity
Events:
Send request: e1{timestamp}
Received response: e4{timestamp}
Events:
Receive request: e2{timestamp}
Send response: e3{timestamp}
Collect, filter and combine events
Filter policy
Complex event:
ce{e1,e2,e3,e4}
Service policy:
roundtrip time <= t
Complex event processor – Performance
Compare actual to target:
e4.timestamp – e1.timestamp <= t

Figure 1. Mechanism for the exemplary QoS attribute performance

nents like services, processes, application servers, hardware platforms, etc. we refer to as SOA entities (SE). Dedicated measurement points at the SOA entities are equipped with sensor components (SC). The characteristics of the sensor components differ depending on the measurement point. A sensor component may be some source code attached to a services source code, a JMX client component, or even a GUI element, like buttons, sliders, etc. In common, these sensor components collect actual data from the SOA entities.

A sensor component emits events that include the information needed for further analysis. The included information as well as the necessary number of events strongly depends on the desired QoS attribute. The latter depends, among others, on the number of measurement points. If several events are needed they are combined, which generates an abstract event, also called a complex event [10].

Figure 1 visualizes the mechanism for the QoS attribute performance:

1) In the sensor components events including timestamps are emitted.
2) The events are collected and filtered based on filter policies. The filter policies thereby describe *what* events emitted by *which* sensor components are combined to a complex event.
3) A complex event is generated including the collected events needed.
4) The complex event is processed by a complex event processor. Incoming complex events are analyzed on compliance to QoS attributes defined in the service policy of the SOA.

In our example target roundtrip time is compared to the calculated actual roundtrip time. Therefore two of four measurement points (send request, receive request, send response, receive response) are used.

In summary: With a grown SOA it is desirable to evaluate compliance to specified QoS attributes. The combination of SOA and CEP results in a highly flexible approach to detect compliance to or violation of QoS attributes constraints by target-actual comparison. This work offers a solution architecture that is able to perform such a target-actual comparison. The comparison is not limited to information extracted from single services only, but also from whole business processes, the application server and/or the system platform. By filter policies analysis can be controlled to single QoS attributes or SOA entities. The solution architecture is able to analyze QoS attributes from technical domain as well as from business domain. Exemplary QoS attributes are performance, roles and rights, reliability, schedule and cost. The solution architecture also is able to react in various ways, reaching from display on a dashboard towards automatic anatagonization using dedicated escalation applications.

The paper is organized as follows: The next section gives a brief description of the requirements this architecture has to address. Afterwards our solution architecture is described in detail, including a statement on coverage of the given requirements. A realization of the solution architecture and exemplary implemented QoS attributes are described in the proof of concept section. We then provide a discussion on related work. Finally, a description of potential future work and our conclusion is given.

II. Architectural requirements

In a SOA application environment, there are several situations where it is desirable to support QoS attributes. Remember the QoS attributes performance, schedule and cost, which relate to an ordering process with a due time for shipment. But how can compliance of a SOA to its QoS attributes be shown? A flexible and powerful solution architecture is required to measure compliance of the SOA to its QoS attributes.

The aim is to evaluate conformance to desired QoS attributes. Thereby, QoS attributes may relate to technical domain, like performance, or business domain, like cost and schedule. Ability to handle QoS attributes of both domains is a requirement.

The solution architecture

1) needs to be able to determine the actual state of a SOA concerning its desired QoS attributes and
2) to compare this actual situation to the defined QoS attributes.

To sum up, the architecture performs a target-actual comparison on QoS attributes described in the service policies.

The actual situation concerning QoS attributes is determined based on relevant data captured from SOA entities. The solution architecture therefore needs to provide an appropriate capturing mechanism. Relevant data are to be captured by sensor components at dedicated measurement points located at specific SOA entities. The SOA entities thereby may be of different type (service, process, application server, etc.), possibly distributed and under diverse governance. The need of source code change at the SOA entity to apply the sensor component is to be kept to a minimum to increase acceptance, applicability and interoperability. The aim is to minimize necessary modifications to the SOA.

Next, the solution architecture needs to provide a configurable filter component to filter the data according to defined filter policies. The filter policies define i) on *which* QoS attributes target-actual comparison is to be performed ii) *what* data are to be captured and iii) *where* the data are to be captured. For a filter policy a declarative language ,e.g., WS-Policy [3], is to be used to enable modification the filter behavior without the need of recompilation.

The relevant data captured at the SOA entities can be seen as a kind of events that contain the appropriate data for further analysis. The solution architecture needs to be able to combine desired events to a more abstract event, as described before in the performance example.

Another requirement is to offer a flexible mechanism that allows to react on specified conditions. Appropriate activities thereby include execution of applications for active antagonization (e.g., cancellation of request execution) as well as the compilation of statistics (e.g., display of statistics on QoS attribute conditions).

Finally, the solution architecture should be based on standards and well-known frameworks. By using standard frameworks and products the applicability to and the interoperability of different environments is increased.

III. Solution Architecture

A. Description of solution architecture

In this section, an architecture that meets the requirements in the previous section will be presented. Figure 2 gives a basic overview of the solution architecture, which consists of four components: i) the QoS-aware SOA, ii) the filter component, iii) the analysis and statistics component and iv) the escalation component. The QoS-aware SOA is already existing. It is to be enhanced and monitored by the other three components to enable a target-actual analysis on desired QoS attributes.

In a QoS-aware SOA, the desired QoS attributes are defined by service policies, which describe the target states. To determine the actual state of QoS attributes, sensor components are applied to SOA entities at specific measurement points. The sensor components are responsible for emitting events with collected actual data. The data was read and composed from the SOA entities and sent to the monitor and filter component.

The monitor and filter component has two tasks: i) to observe the SOA environment and to collect the emitted data from the sensor components; ii) to filter the received data according to filter policies.

The term policy is used in both, QoS-aware SOA and monitor and filter component. It is to be interpreted depending on the context: Within a SOA, the term policy refers to service policies, for example W3C standard WS-Policy [3], that specify the target non-functional behavior of a certain Web Service. Within the monitor and filter component, the term policy defines the behavior of data filters.

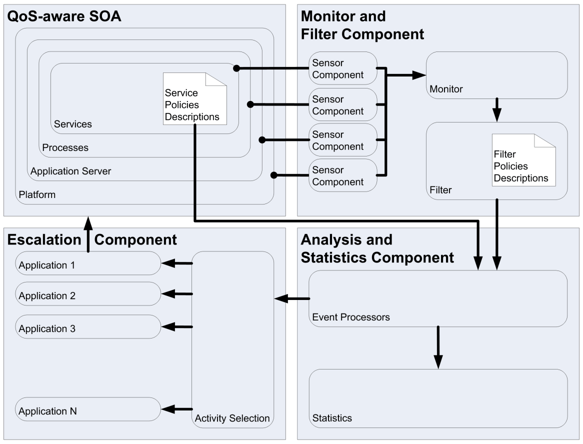


Figure 2. Basic overview on the solution architecture

The analysis and statistics component compares actual data to service policies of the SOA environment. To do that, the filtered data are compared with the service policy. This component also creates statistics on QoS attribute conformance or violation. If an escalation is desired, this component will trigger the next component to perform the appropriate escalation.

Finally, the escalation component provides desired activities, like solutions to solve service policy violations on the SOA environment. The activity strongly depends on the objectives, as will be shown later by example in Section V.

B. Why does the architecture meet the requirements?

First, the solution architecture is able to collect data from the SOA entities. For example, with performance it is able to collect timestamps data whenever a SOAP message is initiated or received (see Figure 1). The architecture attaches sensor components to SOA entities like e.g., Web Services. These components collect data and send them as an event to the filter component. Once an event is received, the filter component checks its filter policy in order to decide what to do with such an event.

Second, our architecture keeps code modifications to SOA entities to a minimum. Ideally, none of their source code will be changed. As mentioned before, the monitor and filter component must be able to collect data from the SOA environment. Therefore, data from the SOA entities needs to be sent to the monitor and filter component. There are different ways to put this into effect. Either new code fragments must be added to the SOA entities, or realized by sensor components, like SOAP message handlers, that work as proxies for the incoming and outgoing SOAP messages. The sensor component could be based on different technologies, such as JMX or even ESB events components. Also, a network sniffer could be used as a sensor component to analyze network traffic for a certain request or response. In summary: Several options do exist to implement the handler approach. The decision, which one to use, strongly depends on different factors; for example, to which extend the SOA entities are allowed to be modified.

All the sensor components have in common that they are attached to measurement points within the SOA. The measured data then need to be transferred to the monitor and filter component. Therefore, events are created and emitted at the sensor components. All incoming events at the monitor and filter component are filtered according to the filter policy. The events that the policy allows are forwarded to the analysis and statistic component. Other events will be ignored. Optional, all incoming events are saved permanently.

At the analysis and statistics component the events that passed the filter are further processed. If needed, the events are combined to complex event. Analysis is performed on the complex events. The solution architecture described here is able to handle both, single events as well as complex events.

Compliance to desired QoS attributes are detected either based directly on the events or on complex events. This method enables the system to measure compliance of a SOA to a single quality attribute. Moreover, several complex events again can be combined with events or complex events. With this mechanism, the solution architecture is also able to measure compliance to combinations of several QoS attributes. For example, if several QoS attributes in combination have impact on other QoS attributes.

The architecture is based on standards, frameworks and products. However, some components, like sensor components, need to be implemented from scratch.

The escalation component is an important part in this architecture. It provides a way to initiate appropriate escalation measures as well as a way to start a certain activity in case of a service policy violations. The functionality of this component strongly depends on the kind of violation and predefined objectives of the escalation component.

In a nutshell: The provided solution architecture fulfills all the requirements specified in Section II. It is able to monitor diverse entities of a SOA application landscape. Changing the source code within the SOA landscape is kept to a minimum by using sensor components to the monitored component. Events emitted by the sensor components are collected by a monitor and filtered by an adjustable filter. The filtered events are further analyzed to measure compliance of the SOA to QoS attributes, which are described in a services' policy. Also based on these events statistics are generated. This architecture is based on standards and well-known frameworks. Finally, the architecture offers an escalation component, which is used to trigger desired activity in case of compliance to or violation to a services' policy.

IV. Solution Architecture Details

For each of the four systems of the solution architecture we will in detail describe the input and output data, the performed tasks and the entities within the systems.

A. *SOA Application Landscape*

The SOA application landscape consists of several entities that we named SOA entities. But what are these entities? Obviously, there are the different kinds of services, like component service, composite service, workflow service, etc. In addition to that there are processes, realized by appropriate combinations of services. An entity of the SOA application landscape is also the application server itself. Finally, the platforms (operating system, hardware components, etc.) are such entities, too. Typically, these SOA application landscapes are huge and grown distributed systems. In consequence, these systems are highly complex, and so is the communication structure within.

B. *Monitor and Filter Component*

All the SOA entities are to be interlinked with a monitor and filter system to determine the actual states of the QoS attributes of the entire SOA entities. The applied sensor components collect actual data, encapsulate these data in events and finally emit these events. The sensor components are situated within the SOA application landscape, but they are part of the monitor and filter component. The sensor components detect and indicate changes in state of the SOA entities. As a simple example: At the time a service receives a request, for example a SOAP message, its state changes. This change in state is detected by the sensor component, which is situated prepending to the service. With rights and roles the sensor component will then determine the user principal of the SOAP message. Afterwards an event that contains (besides other information) the principal is generated and emitted. At a more global view each SOA entity equipped with such a sensor component emits a corresponding event once a SOAP message is received. Received events are optionally stored before the further processing, like the filter mechanism. Because of the storage the system is enabled to perform retrospective analysis.

The received events are filtered according to the desired filter behavior. The filter behavior is described by the filter policies in a declarative manner, for example using XML. For example: Suggest a SOA equipped with sensor components for performance and for rights and roles. In the filter policy, the desired QoS attribute (rights and roles) and its corresponding sensor component IDs are described. For each received event the sensor component ID is compared to the ones specified within the filter description. If matching, the event is forwarded to the corresponding subsequent processing unit, as described later. So in our example events related to rights and roles pass the filter, events related to performance do not pass. The output of the monitor and filter component are events.

C. Analysis and Statistics Component

The output events of the monitoring system is the input for the analysis and statistics system. Within this component the input event vectors are further processed. For each event vector, respectively the corresponding QoS attribute, an event processor is provided. With the example stressed before two event processors are provided - one for roles and rights and one for performance.

Within the event processors the events are combined to complex events and analyzed according to the QoS attributes requirements. In a first step, the service policies (located in the SOA) of the SOA entities are read by the event processors. As described above, the service policy contains the description of the target state for a QoS attribute. Next, from the event vector the events that correspond to the SOA entity are extracted and combined for further analysis. Based on these complex events the actual state concerning the QoS attribute is determined. In the performance example stressed before, the actual message transfer times and the processing time of a SOA entity are determined by four events. The filtered events of the same SOA entity ID and message ID, which corresponds to receiving a request and sending the response, are combined to a complex event. The processing time can now be determined from the complex event by subtraction of the timestamps. Finally, the result (actual state) is compared to the target state. The result indicates compliance to or violation of target state. In either case, statistics can be generated, like performance violation per time unit or a list of principals for each SOA entity.

In a nutshell, the analysis and statistics system is a collection of event processors and generated relations. For each quality attribute a dedicated event processor is needed, since combination of used events and attached additional information is individual for each quality attribute. From the results of the event processors desired relations are generated. The outputs of these components are QoS attributes compliance or violation vectors and the generated relations.

D. Escalation Component

The final component of our solution architecture is the escalation component. The component in essence is a collection of individual application that establishes certain activities based on the output of the analysis and statistic component. These activities are highly individual. For example, on performance violation an application that issues a ticket to a ticketing system might be started. Or a kind of management application that upscales resources for the SOA application landscape. Another option is an information cockpit application. On compliance of actual states to target states the cockpit indicates green condition, on violation red condition. Additional information, like performance status of the last hour, may also be displayed by gaining access the corresponding statistics.

Relationship among event and escalation activity can be 1-by-1 or 1-by-n. This means that for an individual result of the analysis and escalation system, like a QoS attribute violation, several escalation activities may be issued. For example, on performance violation a ticket is issued and the resources available to the SOA application landscape are upscaled. By these examples it also becomes obvious that the escalation system does not necessarily influence the SOA application landscape. Issuing a ticket does not directly influence the SOA, but resource upscaling does.

V. PROOF OF CONCEPT

In the following, we will describe our proof of concept implementation of the prior described solution architecture.

A. System Overview

The solution architecture in general is realized based on several established frameworks and standards. For the SOA application landscape we used the Enterprise SOA (eSOA) showcase by q-ImPrESS [11]. For the monitor and filter component as well as for the analysis and statistics component GlassFishESB with IEP runtime component [12], [13] is used. IEP includes an implementation of CEP. The event processors are implemented using NetBeans and IEP design time component. At runtime the event processors are hosted at the GlassFishESB application server. To store events the default setting of the IEP component, Apache Derby, is used.

The eSOA showcase is a set of exemplary applications from the domain of order and supply chain management forming a non-trivial service oriented system. It implements simulators for customer-relationship-management (CRM), product data management (PDM), pricing, inventory, order and shipment. The showcase is based on Web service technology and Java. For communication SOAP messages are used.

Sensor components are applied to the Web services of the eSOA showcase. In detail, we implemented SOAP message handlers for some exemplary QoS attributes, as we will describe later. On server-side the sensor components are positioned before the individual Web services, as can be seen in Figure 1. Before, in this case, means that the SOAP message handler is positioned between the client and the Web service. In consequence, a SOAP request first passes the SOAP handler before it is received by the Web service. And a response of the Web service first passes the SOAP handler before it is received by the client. In addition, subsequent means that a request message first passes the SOAP handler and is then sent to the Web service. A response message first passes the SOAP message handler and is afterwards forwarded to the client. On client-side the situation is vice-versa, if a client-side sensor component is needed.

All sensor components have in common that they emit events. The characteristics of the events depend on the QoS attributes. The emitted events are collected by the IEP component of GlassFishESB. Further, these events are filtered using corresponding event processors. For the filter policy description XML is used. The filter policy thereby consists of the QoS attribute to pass the filter.

After passing the filter the events are processed by the analysis and statistics component. The analysis is performed by event processors. An event processor basically consists of an input stream and some processing entities that end in an output stream. For each QoS attribute a corresponding event processor is implemented. The individual event processors are designed and executed using the IEP design time and runtime components.

B. *Exemplary quality attribute implementations*

In our proof of concept, we implement selected QoS attributes. In a first step, we implemented the QoS attribute of performance. In more detail: Based on SOAP message handlers (client-side and server-side) we determined request and response transmission time, service calculation time and roundtrip time.

Also from the technical domain, roles and rights are implemented. In brief: The individual Web Services of eSOA are called by clients. Each client has an individual identification. The right and roles model defines which Web Service may be called by a certain client. The motivation for this scenario is the analysis of a grown SOA on conformance to given right and roles model. The task is to generate a statistic on principals of service requests.

Different sensor components are implemented for this task. A first kind of sensor component is a SOAP message handler at server-side prepending to a Web service. Within this SOAP message handler the principal of the incoming request is determined from the SOAP message context. Then, an event including this information on the principal is emitted to the corresponding event processor. In brief, an IEP component is a JBI module, that is added to and deployed with a composite application. The event processor appears as a Web Service that, in our case, uses SOAP for communication. At the sensor component, respectively our SOAP message handler, the created event, in essence, is a SOAP message. The structure of the SOAP message is defined in the JBI modules WSDL. Emitting the event means, that this SOAP message is sent to the Web Service exposing the event processor.

With regards to righs and roles, one concrete example would be the SOAP request from the client is processed by the SOAP handlers handleRequest() method. From the MessageContext the principal of the SOAP request is determined using method getUserPrincipal(). Next, a SOAP message including the principal is generated and sent to the event processors Web Service. At the event processor the principal is compared to a database that contains the roles and rights model, and any violation is indicated.

An alternative kind of sensor component uses the JMX interface of the application server to extract the information on principal of request on hosted services. Both kinds of sensor components do solve the task, and either can be used depending on existing restrictions. The advantage of the second is that it does not touch services at all, but accesses to application server management console is needed.

Another exemplary QoS attribute is from the business domain: schedule. With this a due date for shipment is agreed. Motivation for this scenario is, for example: Shipment of a placed order is guaranteed within 24 hours, otherwise a certain discount is allowed. The task is to ensure this in due time. If the due time is exceeded, statistics on time overruns are to be generated and discount is to be given out. Again, the Web services are equipped with SOAP message handler as sensor components. From the SOAP messages the information on order ID, actual time and due time is extracted. With that information time overruns are detected and statistics are generated at the analysis and statistics component. On violation an application within the escalation component automatically allows a discount.

VI. Related work

In 2005, an architecture based on Web services including comprehensive QoS support was described by Berbner et al. [5]. Within it particular Web services are composed to workflows. Thereby, the selection of Web service is based on their QoS properties that they guarantee in Service Level Agreement (SLA). To ensure compliance to given SLAs a monitoring component is mentioned, but not described in detail.

The design and implementation of a high-performance QoS monitoring system was presented by Zeng et al. [6]. Their two main issues on the monitoring system are the service monitoring architecture and the QoS metric computation. Within their work a *QoS observation metamodel* with three types of monitoring context, one on processes and two on services, was developed. So the measurement points for QoS monitoring are limited to the services and processes. Our work is a more general approach, since we do not limit ourselves to services and processes, but support any SOA entity as described before.

A JMX-based monitoring extension of Java system application server for the QoS attributes availability, accessibility, performance, reliability, security and regulatory was described by Artaiam et al. [7]. They also give a detailed description of QoS attributes metrics. However, they are limited to service-side monitoring, which means QoS monitoring of services within the application server (GlassFish). Client-side QoS monitoring is not included. Our approach explicitly enables both, server-side and client-side monitoring.

The integration of an existing client-side monitoring approach and a server-side monitoring using CEP to monitor SLA was elaborated by Michlmayr et al. [8]. For CEP the open source implementation ESPER is used. For monitoring at client side so-called QoS monitoring schedules are used that specify that certain services are monitoring in certain time intervals. On server side a .NET technology is used, which i) is a limitation to certain server infrastructure and ii) also limits the service implementation to .NET technology, as is mentioned by the authors. The described solution architecture of the authors also includes a notification mechanism to subscribers on detected SLA violations. Their approach is similar to our approach. However, they use ESPER, focus on .NET technology and monitor at dedicated points in time. In contrast, our work uses the IEP component. Although our proof of concept uses Java, it is also applicable to other technologies, like .NET. Next, we use continuous monitoring rather than dedicated points in time. And finally, we provide a mechanism to trigger activities.

Monitoring of adaptable SOA was described in Oriol et al. [9]. The focus is on dynamic adoption of a QoS-aware SOA. Within the QoS-aware SOA QoS attributes are stated using SLA. The current QoS values are monitored by a monitor component and compared to the stated SLA at an analyzer component. SLA violations are shown to a decision maker component, which is able to perform adjustments within the SOA. In contrast to our work, this approach focuses on adjustments within the SOA. Our approach is more general. In our solution, architecture adjustments are just one aspect of possible escalation activities.

VII. Conclusion

By the combination of SOA system, monitor and filter component, analysis and statistics component and escalation system a versatile and powerful tool is available to analyze SOAs on compliance to the defined QoS attributes. Using this solution architecture compliance analysis is not limited to services and processes, but also includes other SOA entities, like application server and platform. This enables for several QoS attributes yet not supported, especially QoS attributes, like in our schedule example. With the escalation component a variety of activities can be carried out. These activities may be of more passive nature, like to issue a ticket. Or of active nature, like enabling for a self-scaling SOA. The boundaries of the given approach have not been yet explored.

Our perspective is to enrich the existing systems with additional QoS attributes that are not yet supported. Therefore, it is necessary to determine which QoS attributes are requested from both, the technical and the business domain. And which of these QoS attributes can be formalized and further supported. A related question is the use of alternative description languages for QoS attributes.

We will also implement additional tools to support developers with an interest for our approach. The generated tools are to be added to different IDEs. A tool chain to define additional QoS attributes, to equip Web Services with these, and to deploy such Web services is implemented. Also additional components to ensure compliance to these QoS attributes will be provided.

Acknowledgments

We would like to thank the anonymous reviewers for giving us helpful comments. This work has been partly supported by the German Ministry of Education and Research (BMBF) under research contract 17N0709.

References

[1] “Web services quality factors v1.0,” http://www.oasis-open.org/committees/download.php/38611/WS-Quality_Factors_v1.0_cd02.zip, accessed at 27. Aug 2010

[2] L. O’Brien Lero, P. Merson, and L. Bass, “Quality attributes for service-oriented architectures,” in *Systems Development in SOA Environments, 2007 (SDSOA’07)*

[3] “Web Services Policy 1.5,” http://www.w3.org/TR/ws-policy

[4] “Web Services Security v1.1,” http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[5] R. Berbner, O. Heckmann, and R. Steinmetz, “An Architecture for a QoS driven composition of Web Service based Workflows,” in *Networking and Electronic Commerce Research Conf. (NAEC’05)*

[6] L. Zeng, H. Lei, and H. Chang, “Monitoring the qos for web services,” in *Proceedings of the 5th Int. Conf. on Service-Oriented Computing (ICSOC’07)*

[7] N. Artaiam and T. Senivongse, “Enhancing service-side qos monitoring for web services,” *ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD’08)*

[8] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar, “Comprehensive qos monitoring of web services and event-based sla violation detection,” in *Proceedings of the 4th Int. Workshop on Middleware for Service Oriented Computing (MWSOC’09)*

[9] M. Oriol, J. Marco, X. Franch, and D. Ameller, “Monitoring Adaptable SOA-Systems using SALMon,” in *Workshop on Service Monitoring, Adaptation and Beyond*

[10] D. Luckham, *The power of events*, 5th ed. Boston, Mass. [u.a.]: Addison-Wesley, 2007.

[11] Q-ImPrESS, “Enterprise SOA Showcase.” http://www.q-impress.eu/wordpress/software/, accessed at 27. Aug 2010

[12] S. Microsystems, “Glassfish application server.” https://glassfish.dev.java.net/, accessed at 27. Aug 2010

[13] Oracle, “Intelligent event processing (iep).” https://open-esb.dev.java.net/IEPSE.html, last accessed at 27. Aug 2010

Nontechnical SPAM Detection Paradigm in Unified Communications Systems

Moritz Giesecke
School of Engineering, Pforzheim University of Applied Sciences
D-75175 Pforzheim, Germany
moritz.giesecke@hs-pforzheim.de

***Abstract*—The recognition and filtering out of unwanted messages in technical communications media presents an ever more difficult challenge. The best-known of these problems is with ubiquitous e-mail. Most e-mail sent are unwanted spams. In order to protect the recipient the most diverse applications must be used. Longer observations have shown that spam is continually adapted and is able to overcome the most up-to-date recognition programs. In the future the most widely different communication methods are growing together such as e-mail, telephony and others, so that soon we will be able to speak of unified communication. There is a danger that these other communications media will increasingly become the target of new types of spam. On the other hand this logical union opens up new possibilities for spam recognition. In this paper, a behaviour-based evaluation paradigm is introduced which works on a uniform basis for all communications media. It uses an evaluation of the three parameters of abstracted times of usage, distance of communication partners and costs. All communication events between media using actors create a social network whereby the actors are clustered according to their social proximity. The evaluation of spam is a result of the actors and cluster specific communication behaviour up to a point. In this way a new non-technical level of analysis is created, which spammers can only overcome with difficulty. Likewise the problem of limited focus in network centred filtering programs is dealt with. The presented filtering paradigm can be used unitary in all technical communications media and works with the same three nontechnical parameters at a behavior-based level.**

***Index Terms*—spam, spit, unified communications and social networks.**

I. Introduction

In modern communications media the proportion of unwanted messages is continually growing. A classical example is e-mail spam, which has appeared since the widespread use of e-mail services. Normally, these are differentiated between unsolicited commercial mail (UCE) and unsolicited bulk e-mail (UBE). Both forms are normally characterized as spam [1]. The particular societal and economic meaning of this amount, around 120 billion spam e-mails per day, or calculated at up to 20 spams per day per person is fatal [2]. Typical return rates are under 1 per thousand, depending upon the quality of the spam [3]. By processing these spams economic damages are incurred in the form of lost working time, server and energy costs as well as the irritation of the users of the e-mail service. By today, classical spam technologies are no longer used only for advertising purposes; they are used for fraud, typically called phishing [2].

Until now various classes of processes were used, based upon the individual e-mail infrastructure or the users' mail boxes to protect against incoming spam. These differ according to granularity, effectiveness as well as complexity of the filtering. In a typical mail server, these filters are arranged in a cascade, see Fig. 1. Starting with a Firewall, all incoming connections from IP addresses recognized as known spam senders are blocked. After this first IP address based list process, different black and white lists with known spam servers and e-mail addresses are queried. These come from specialized companies, which put a lot of effort into the finding of the most up to date and correct data. Typically, the highest level of effectiveness is achieved through recognition of spam e-mails and through the avoidance of false positives. For this, three different lists are used and then evaluated with a two out of three decision. Afterwards, information in the e-mail header is checked to see if the address name and server details are correct for the domain to send e-mails by searching DNS records, which contains appropriate mail exchange information [4]. After this, a specification of the SMTP protocol is exploited using the grey listing process [5]. In doing so, a temporary problem in receiving the e-mail is simulated in the users own server. Real mail servers wait a certain amount of time and try a new delivery; typical spam senders on the other hand drop the repeated delivery of the spam e-mail. A further recognition method, which consists of comparing as many as possible of received e-mails is the well known process called *Distributed Checksum Clearinghouse* (DCC) with a distributed checksum filter [6]. This consists of cross server boundary checking for the existence of the same or similar e-mails. Following this, e-mails are evaluated according to content based on the signalling process method. Typically, Bayes-filters are used, which must firstly be trained with the typical appearance of spam and reasonable e-mails [7]. Afterwards, on the basis of this training, the e-mails were scanned for these patterns of learned words and then weighed against each other. This results in a decission about the current received e-mail, whether it contains spam or not. In addition to these generally easily accessible procedures, as they are reproducible from open source software, there are different commercial service providers with proprietary procedures offering the evaluation of incoming e-mails. An example named here is Cisco SenderBase, which works off of a central database containing reputation values of individual e-mail addresses and organisations [8]. The evaluation process uses more than a hundred different parameters for the evaluation [9]. An important critical point is the central capture of the e-mail traffic occurs nearly in real time and the long term storage of the same. Parallel to these technical filtering processes, the introduction of a global legal barrier for allowed e-mail marketing

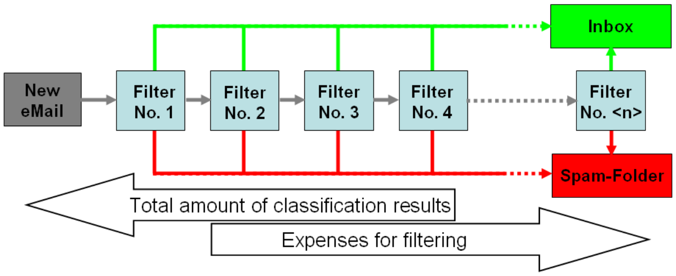


Figure 1. Spam filter cascade with cost-benefit relation

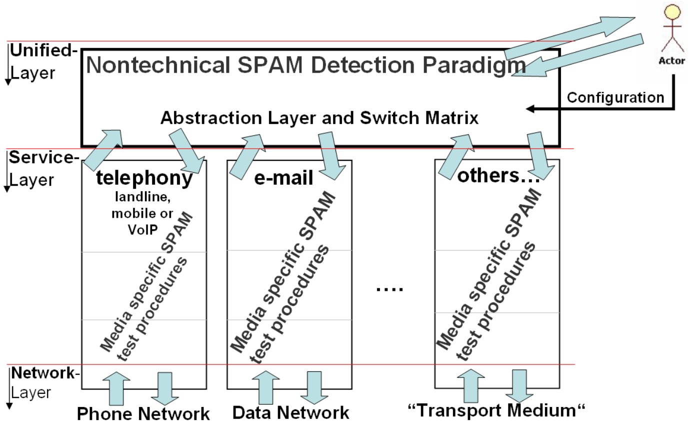

Unified-Layer
Nontechnical SPAM Detection Paradigm
Actor
Configuration
Abstraction Layer and Switch Matrix
Service-Layer
telephony
landline, mobile or VoIP
e-mail
others...
Media specific SPAM test procedures
Media specific SPAM test procedures
....
Media specific SPAM test procedures
Network-Layer
Phone Network
Data Network
"Transport Medium"

Figure 2. Unified Communications System in an abstract view

since at least the end of the 1990s took place, i.e., in the USA and Germany USA and Germany [10] [11]. In spite of a few, but deterring judgements there has been no visible reduction in spam. The leading suspects are known in part by name and with a photograph [12].

A. Characteristics and transformation of spam

In order to be interesting for the spam-senders, a few characteristics must be present in the communication media. Mainly, low cost, so that an individual spam process can not incur any costs. Because of the low rate of reply gigantic amounts of individual spams are sent. Equally important is the possibility of sending a *variable content*. In order to provoke a response from the recipient of the spam, the spam appearance must be varied. *Limited traceability* meaning that the spammer tries to conceal their true identity to avoid trouble with the recipient of spam. There can be possible civil suits for damages and financial compensation and severe criminal consequences. *Simple completion*; meaning that the recipient of the spam should easily be able to respond to the spam. Typically, spam has a feedback link, which logically lies as close as possible to the communication media of the spam. For example, a successful spam e-mail pulls the user who received it directly to a web page, which may instantly be opened with one mouse click.

In spite of the laws against spam and other legal instruments available there is little help on the way towards a more tightly ordered e-mail framework. That is why new technological evolution must be continuously developed and implemented in order to act against the continuous flood of spam. Communications media are the target of spam as soon as the above mentioned characteristics are fulfilled. In addition to the old e-mail spam problem, the spam over internet telephony (SPIT) is growing recently, but is still not as intensive as traditional e-mail spam. With a telephone call, the recipient is provoked into giving a reaction which allows the spammer to make a profit. An example hit German customers that used a SIP-based VoIP connection in the first mass spamming in September 2008 [13] [14].

B. The idea of Unified Communication

The users of modern communication technologies are taken in more and more by the complexity of the technology and the operating effort for the user of different communications media such as e-mail, telephony (land lines, mobile or cell phones and voice mail services) as well as multiple specialized services (Instant Messaging, Pager, Groupware solutions). In addition to various user variations and configuration options on the part of the recipient's end it is also difficult for the communications initiator to reach the desired communications partner with the correct communications medium. The origins lie in the asynchronous communications media such as e-mail, for which here the term Unified Messaging is used. The idea of Unified Communications (UC), is the managing of different types of communications media bundled in one location with a supporting function to relieve the user. It is not totally clear in the general language usage as of when Unified Communication can be spoken of or rather, which criteria must be completely fulfilled. For the consideration of this paper it is assumed that in addition to the other aspects of UC technology a central instance exists that captures all communication procedures, manages all of the user's preferences and can appropriately influence the communication. As an example calls can be rerouted to a voice mailbox and the message can be sent by or as an e-mail. It is equally possible to think of calling up e-mails by telephone and having them read aloud using a text2speech system. The required settings would be in put by the user themselves, and thereby be part of the user's preferences. Such an Unified Communication System (UCS) scheme is shown in Fig. 2. All communications media in the UCS, are connected to the rest of the world by their different transport media in the *network layer*. Typically, this is an IP-based network for e-mail and VoIP or traditional phone networks often referred to as PSTN. Any *other* communication media are of course also conceivable. The different communications media resides, in the so-called *service layer*, where they are considered as separately existent. The usual *media specific Spam test procedures* (e.g. signalling and content evaluation), are applied here. The *unified layer* is aware of all communication events sent or received by the actor, indicated by the light-blue arrows. Furthermore, this layer makes available also all the benefit features described with the idea of UC, once it gets configured by the actor. This is the instance is the place, where the subsequent described nontechnical Spam detection is carried out.

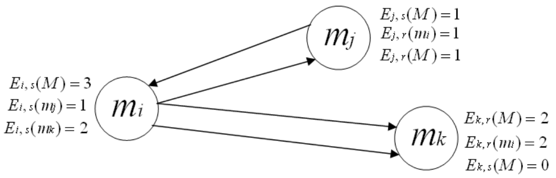


Figure 3. Social network as a graph with actor properties

C. Information gain with social networks

With the idea of the social network connections between people can be formally modelled, whereby the interactions of people can be graphed. Fig. 3 depicts the connections E, shown as edges, between the individual human actors $M = \{m_1, m_2, \dots\}$, shown as nodes. The edges arise from the performed communication events within the group of observed actors. This concept uses a communication process, for example a delivered e-mail or a finished telephone conversation or any other discrete event using any other possible communications method. In this way any interaction between people through the use of communications media can be represented. At first invisible information content is made up of exposing the relationship, the organizational structures, work processes and the influences of events. The sociological and mathematical formulated questions of the *social network analysis* (SNA) have been researched enough and have found practical applications in sociology, economics and criminalistics. Further applicable methods such as data mining can be derived from general sources [15] [16].

In large UC systems of telecom companies are I different human actors present, who have access to K different communications media. Between the system members and the system non-members from outside user communications processes take place, which can be transferred into a social network. At the observation starting point there are already a number of communications processes available so that an adequate connection density amongst the actors of the social network exists. Because of the freedom of the modern communications media, boundless interactions between them are possible and the evaluation of the social proximity network follows through the intensity of the incidences of communications events. Therefore the choice of the communications medium K is irrelevant. Important is the individual communications event E only. Thus a first approximation of the social proximity N_{soc} between two actors is given through the amount of events between them. Here is a summary of reciprocally received events by recipients (**r***eceived*) where the syntax $E_{<acteur>,r}(< partneracteur >)$ is used:

$$N_{soc}(m_i, m_j) = E_{i,r}(m_j) + E_{j,r}(m_i)$$

A proposition with this first assumption does not take into account the relativity of the amount of communications events that the actors or partner actors sent (indicated by *s*) to the other actors within the observed social network. In order to be able to capture these relatively important reciprocal events, both actors are introduced with an additional proportionality factor $\frac{E_{<Acteur>,s<PartnerActor>}}{E_{<Actor>,s}(M)}$. In the sum of all E_s of an actor to a partner actor and the sum of all E_s from this actor to all other actors M, the attractiveness of the respective partner actor is determined from the ratio between the count E_s to the respective partner actor and the count of E_s to all others actors in M. Because in a UCS the human actors continuously communicate, the form of social network is seen as variable and therefore also those with the equivalent (1), determined value for N_{soc} between two actors.

$$N_{soc}(m_i, m_j) = E_{i,r}(m_j) * \frac{E_{i,s}(m_j)}{E_{i,s}(M)} + E_{j,r}(m_i) * \frac{E_{j,s}(m_i)}{E_{j,s}(M)} \quad (1)$$

That is why it is recommended to define an observation interval Δt, within which a calculation of N_{soc} (approximation) is seen as valid and must be newly recalculated. These order of magnitudes of the observation intervals result in the emergence of new E, the system performance capability towards the eradication of filtering cycles (see Section II) as well as the volatility of the current spam in comparison to the recognition capability of the filter systems.

For N_{soc} the valid conditions are that the value of the result is non-dimensional, $N_{soc}(m_i, m_j) = N_{soc}(m_j, m_i)$ and $N_{soc} \geq 0$. The clamping of such a social network can first take place after a initial observation time, meanwhile the actors have produced a certain amount of communications events E. For the new evaluation process of this paper the quality of the social network depends upon as many as possible of the intended communications processes in the network derived by means of the intended prototypical performance. Only a few spams which were able to overcome the previous filters can be tolerated as they will be detected as inappropriate.

D. Related works

Most spam recognition processes are based on the technical signalling information of the various communications media, e.g. within the transmission of e-mail typically on the level of Internet Protocol and SMTP. A further class of processes work on the basis of content, in the e-mail service as an example Bayes and Markov filters or DCC as well as VoIP methods for the differentiation of humans and machines. Furthermore there are ideas for the use of processes out of the field of social network analysis (SNA) for the recognition of spam [17] [18].

Typically these approaches are used on ordinary e-Mail traffic and use the results of various metrics to scan for the characteristics of spam. Here, primarily two general classes of procedures are widely used. On the one hand it is attempted to assign each communication participant a reputation value based on experience over a long period of time. On the other hand, the behaviour of communication participants vis-à-vis other participants can conclude the likelihood of spam.

Nevertheless, in the technical reality there exists the problem of a limited focus. Every filter instance of any such SNA based process can only work with the information that runs through the communications system used. This leads to spammers from outside the SNA based filter system, in certain instances, exhibiting no complete characteristics of spam if they, for example, only send a small amount of spam in the focus of the SNA filter systems [19]. Thereby the spammers wouldn't, in certain circumstances, be recognized as such. From the main countries of origin spam is distributed globally and it can be assumed that these will not be completely hit by the individual SNA based filter systems.

The filter processes presented in this paper primarily observe only individual system communication participants using all available communications media. It evaluates, using the method of normal user behaviour as though these successful incoming communications processes are desired. In the following, a non-technical level of analysis that circumvents the problem of limited SNA focus and thereby presents a realistic scenario suitable for the communications infrastructure of the telecom companies (development of design technology for telecommunications service provider), is introduced.

II. Filter Methods

The UCS passes all data of the users to the filtering processes. These parameters are: communication partners and times, the communication media, resulting costs as well as possible profile information. If interaction occurs with a foreign actor from outside of the system limits of the UCS, which involves a previously unknown actor, this communication will be saved to the database as well. Doing so, all available data of the communication events is captured.

In the literature there are countless methods describing how to search available data bases according to problem oriented parameters. Most of these procedures have their origins in the optimization of business process in commercial fields. Others come from purely scientific queries, for example the researching of questions in sociology. In the UCS there exist the communication relationships of individual actors of the UCS amongst themselves and beyond the UCS. This data is spanned as a social network. In this way there is no differentiation made about the type of communication media that was used for a communications event that has taken place. In the social network there are clusters in which the partial totals of the actors are significantly more densely bound together compared to communications participants from other areas of the social network. These clusters are detected using a *k-Means-procedure* and the actors are appropriately assigned to them [17]. This class of procedure is a simple and fast method of cluster identification and is widely used in SNA applications. The number of clusters C_{count} is predetermined because the k-Means-procedure is hard partitioned. As a distance function of the k-Means-procedure the social proximity N_{soc} between

TABLE I
EXAMPLE PARAMETERS OF DIFFERENT COMMUNICATION MEDIA TO FEATURE EXTRACTION

communication media	user identification	location parameter	starting time	approx. expenses
eMail	MAIL FROM	initial MTA	delivery start time	amount of data
PSTN	phone number	prefix number	call start time	estimated charges
Cellular radio	phone number	home network	call start time	estimated charges
VoIP	SIP indentity	IP-Subnet	call start time	estimated charges

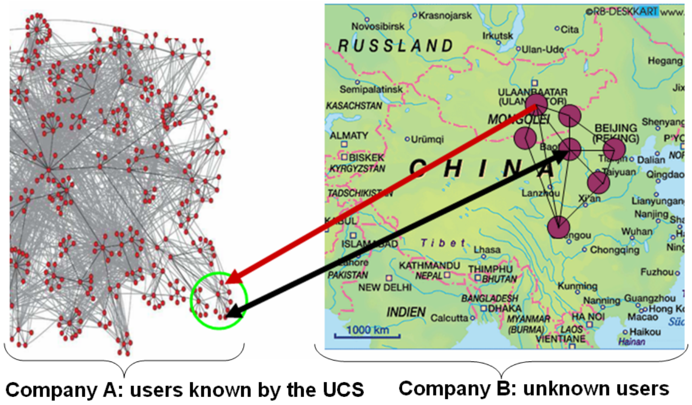


Figure 4. Example of a usage scenario with an incoming event

individual actors is used from (1).

At this point, the theory of operation will be explained with of a pithy example. In a fictive situation company A is collaborating with a company B during a product developement process. The two companies are far away from each other, one of them for example in China, see Fig. 4. The communication of the users in company A are protected with the new filters residing in their UCS. The heads of both development departments have a lively exchange over different communication media with each other, these events are indicated by the two black lines. Then an external developer hired by company B gets a problem definition to solve, for which he has to communicate with a developer of company A. So, both developers had not been in contact before and with the fact of the external hiring, traditional filters like (personal) listing procedures can not use common criterias of company B, e.g. domains in e-mail adresses or transmitted phone numbers. At the moment of receiving a communication event in company A's UCS, sent by the external developer, the filters know about the social proximity between company A's head of developement and company A's developer (actuall receiving). By the former communication behavior between the two developement heads in A and B, the filter system is aware of the heads A parameters in times of usage, distance of communication partners and costs of communication. When deciding about this communication event solely by tradional spam filters, the result could be unsure. And when deciding with former personal behavior of the developer in company A, the result could be unsure too. But now the corrective properties of company A's head of developement, which is detected to have a high social proximity to his developers, could be applied to the filtering process of this incomming communication event.

A. Data pre-processing

Each actor shows connections in the social network through the various communication processes with other actors from the range of available communications media. Out of the communication relationships of individual actors three characteristic relationship parameters are able to be derived by every communication process in every type of communications media available. Additionally, it is possible through the communication behaviour of closely connected actors to connect them to clusters and to in turn derive the three characteristic behaviour parameters of the clusters. These three behaviour parameters are the times h, at which communications procedures take place, the distance d between the participating actors and subsequently occurring costs c. The distance data are deduced from the technical parameters of the appropriate communications medium or communication system. Typically available communications media example parameters are shown in table I. Afterwards this makes procedures usable for position determination [20] [21]. The media specific costs due to a communication process can be deduced from telephone charges or the volume of transmitted data. In order to get the abstract comparable costs of a communication process from various communications media the cost parameters for these considerations are standardized units and therefore comparable in the sense used here.

Each actor is given a 3 tuple as parameter data for the three characteristic behaviour parameter, see (2). The elements of this make up the parameter and can be expressed through their index.

$$T_{m_i} = \begin{pmatrix} \vec{h}(m_i) \\ d(m_i) \\ c(m_i) \end{pmatrix} \qquad (2)$$

Over a long period of time, a human actor displays time focal points upon which multiple communications have taken place. Capturing the time occurrences of the communication takes place in intervals in order to capture phases of increased communications incidences. This method is presented as a bar chart (see fig 5). A class wide of 60 minutes is proposed. Thereby a compromise between cancellation and complexity is given. Consequently a relationship is given between the division of the communications events in the different intervals and information about preferred communications points in time. As indicated in (3), the communications events are assigned to the 24 elements of the time vector $\vec{h}$.

$$\vec{h}(m_i) = \begin{pmatrix} \sum_{k=0}^{1} E_k(m) \\ \vdots \\ \sum_{k=23}^{24} E_k(m) \end{pmatrix} \qquad (3)$$

The captured position of the actors m_i, at every communications event, and his corresponding partner actor is processed to the distance d. In this way a prototypical range of all communications can be calculated out of the number of all communication processes. This takes place through the mathematical mean, here the distance of all communication processes totalled is divided by the set, see (4).

$$d(m_i) = \frac{\sum_{k=1}^{E_m(Count)} E_{mk}(Distance)}{E_m(Count)} \qquad (4)$$

Every communications instance has a cost c applied to it, in order to be able to classify the value of a communications instance, see

(5). Typically, this value is generated from the sending actor or his approximate surroundings. From the sum of all communication instances a prototypical cost value of the communication is calculated. This is represented through the mathematical mean, here the costs of all communication instances are totalled and divided by the set (see equation 4).

$$c(m_i) = \frac{\sum_{k=1}^{E_m(Count)} E_{mk}(Costs)}{E_m(Count)} \tag{5}$$

Subsequently the actor specific first part of the pre-processing clusters are searched for in the social network. For this the k-Means-procedure with the distance function given in equation 1 is used. Thereby the actors are connected to clusters which appear, through the communication behaviour, to be closely linked. After the clustering is carried out, each cluster C_i is assigned with the actors $C_{i,<acteurs>}$. According to the actors parameters three cluster parameters $\vec{h}(C_i)$, $d(C_i)$ and $c(C_i)$ are represented. These result from the arithmetic mean of the actor's parameters in the cluster. Thereby the parameter values of the actors in the cluster are added and divided by the set of the actors in the appropriate cluster. In this way the prototypical actor value for the cluster is generated. This 3 tuple is shown in (6).

$$T_{C_i} = \begin{pmatrix} \vec{h}(C_i) \\ d(C_i) \\ c(C_i) \end{pmatrix} \tag{6}$$

The captured values T_{m_i} and T_{C_i} can have their lines addressed through the index.

B. Evaluating the communications incidences

With the gathered and pre-processed data from the observed user behaviour, a simple test for spam or ham can be carried out upon the arrival of an external communication event E_r, on an actor in the UCS m_i. Thus, the actors and the cluster specific parameters are viewed as equal valued statements. With the linking up of individual behaviours, the affiliated actors and through the cluster analysis uncovered connection, the data basis for the decision is considerably enlarged and possible evaluation errors can be minimized. The cluster bound actors are socially affiliated and display similar communications behaviour. In this way the cluster specific statement confirms or corrects the actor specific statement. The ratio, between the new and individual communication event $current$ and the previously determined average value all, will be calculated for each of the three parameters $\vec{h}$, d and c. To evaluate a single communication event a non-naive approach is chosen, which means the result of the relationship comparison drives against the value *1.0*. This means the smaller the result's absolut value of the subtraction, the lower the probability of spam. This is represented by a subtraction of the evidence of a relationship comparison of one. In (7), the number of communication events in the class (time slice) of the current communication process $h_{current}$ is compared against all other classes according to the number of E_m contained.

$$h_{result}(E_r(m_i)) = 1.0 - \frac{\frac{T_{m_i}(1,current)}{T_{m_i}(1,all)} + \frac{T_{C_i}(1,current)}{T_{C_i}(1,all)}}{2} \tag{7}$$

In (8) the mean distance value from the recipient actor and his cluster are tested against the value of the received communication events.

$$d_{result}(E_r(m_i)) = 1.0 - \frac{\frac{T_{m_i}(2,current)}{T_{m_i}(2,all)} + \frac{T_{C_i}(2,current)}{T_{C_i}(2,all)}}{2} \tag{8}$$

In (9) the mean cost value of the previous communication of the recipient actors and their clusters are tested against the value of received communication processes.

$$c_{result}(E_r(m_i)) = 1.0 - \frac{\frac{T_{m_i}(3,current)}{T_{m_i}(3,all)} + \frac{T_{C_i}(3,current)}{T_{C_i}(3,all)}}{2} \tag{9}$$

C. Summary of the filter results

In order to summarize the three different filter results into a result value, a specific term must be used that allows it on the one hand to contain the total result, and on the other hand takes into account the characteristics of the individual filter levels. According to size, or rather construction of the social network in a database, various individual results can be achieved using evaluations metrics. A simple scoring method is based on experience values existing above the reliability of individual metrics in the usage context of the UCS. If it is recognized that the participant results invalidate, the spam level of a communication instance $E_{i,r}$, the weighting factors a, b and c not equal to 1.0 can be chosen. The individual metric results are multiplied with the scale factor and the product totalled.

$$Spamlevel(E_{i,r}) = a * h_{result} + b * d_{result} + c * c_{result} \tag{10}$$

For making a decision about a current $E_{i,r}$, whether it is spam or not, a threshold must be defined. The value to be used here is an individual nature, according to whichever risk of *false positives* appears to be acceptable.

D. Filter Position within a total context

The results from the filters described here can not reliably decide on a positive spam detection alone. Because of the multiplicity of possible connections a result value is only an additional indicator in the collection of all evaluation processes. Therefore a linking with the other (media specific) filters of individual communications media is allways necessary. Typically there are the hard criteria, such as firewall or listing procedures, which without cooperation with other filters reach a valid conclusion and soft criteria, such as content evaluation processes. For the final evaluation result of a communications process the results from all filter processes in this paper must be run together with the other processes. Typically a weighting according to reliability as well as personal settings of the associated communication participant will take place in the UCS.

III. Further Aspects

A telephone system's calculations data is taken and investigated for the characteristics of the three parameters $\vec{h}$, d and c. The data contains only the call placed by internal participants to unknown external numbers. The capture resulted originally for the purposes of billing only, not for the new procedure presented here. According to German data privacy law, the collection of unnecessary personal data is not allowed without agreement of every affected participant. There are no telephone calls between two or more known internal participants available, so that no social network in the appropriate sense can be spanned. The pure simulation of an artificial user group, in which each user generates the three parameters ($\vec{h}$, d and c) by random processes, would lead to results that are far away from reality. The following results presented in Fig. 4 to 6 originate from October 2009 data, the resulting analysis was carried out with standard tool boxes from Mathworks MatLab. There are 9298 communication events from 408 different call numbers available, a total of 42,211 charge units were used with an average speaking length of 287.48 seconds. The Fig. 5 to 7 were created using all available data thereby representing the total relationships of all participants. Clearly recognizable are working hours and weekends, as during that time little appreciable private call activity at the university takes place, see Fig. 5.

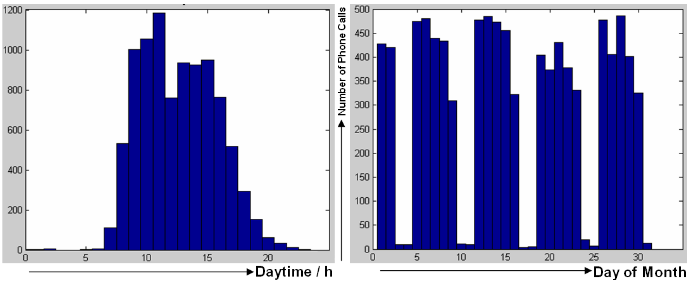


Figure 5. Call distribution over time of day and month

A large proportion of calls are shorter than five minutes and involve minimal cost, based on the charging units, see Fig. 6.

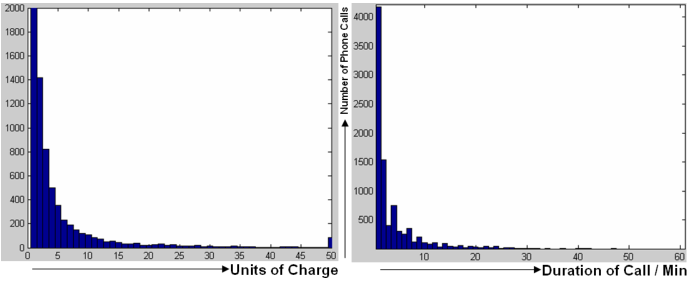


Figure 6. Call distribution over fee and time

The international calls are distributed over only a few target countries. This characteristic could be derived to a criterion for filtering by distance values, see Fig. 7.

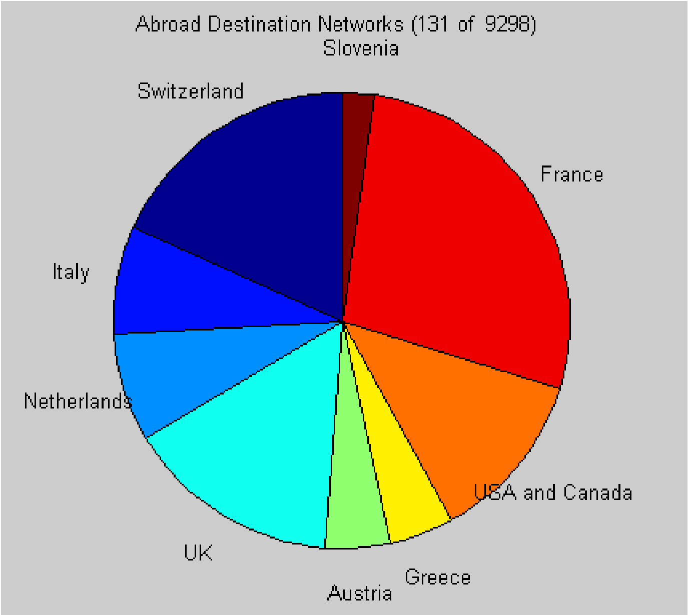


Figure 7. Abroad destination networks over a month

Through the broad range of the appropriately investigated parameters it is evident, that the procedure introduced here can give strongly conclusive results.

In order that these methods do not have to only be used with external databases an experimental UCS is being worked on at Pforzheim University, which unites the communications media of telephony and e-Mail and offers the possibility of the analysis introduced here. It will therefore be possible to evaluate the described procedures in a situation approximating reality and to variably test the parameters of different evaluation processes.

IV. Final remark

Proof of the effectiveness of the presented methods can only, untill now, be given from theoretical experiments as at the moment no data fulfilling the assumed conditions (social network capable) for this theory, is available. Because the decision procedures are based using the three parameters in (7), (8) and (9) on hard limitations, incorrect decisions on these limitation values are not unlikely. Real enviroments impose to implement an imprecise decision threshold, for example through the simple one-dimensional variance $VAR(X) = \sum(X - E)^2$ of the appropriate parameter X, with the arithmetical mean as expectation value E. The effectiveness of constant purported and thereby potentially suboptimal number of clusters through purported quantities of hard partitioned k-Means-procedure is also to be investigated using the available applicable data. Both input values from position and cost estimates will show different exactitude according to origin and communication instance. In combination with several relationship parameters and their histories these parameter input characteristics should be insignificant.

The procedure presented in this paper from three captured parameters of all conceivable communication media (time use behaviour, distance from communication partner and incurred - if only abstract - costs) in combination with individual and sum total behaviour as a reciprocal correction is a *new type* of idea in the battle against spam.

Acknowledgment

This research was made possible through Professor Dr.-Ing. Frank Niemann (Engineering) and the Pforzheim University as well as the Federal Ministry for Education and Research. The data represented in this paper does not necessarily represent the opinion of the Pforzheim University or the Federal Ministry for Education and Research.

References

[1] The Spamhaus Project, *The Definition of Spam*, February 2010, Checked September 2010 http://www.spamhaus.org/definition.html.

[2] Cisco IronPort, *2008 Internet Security Trends - A report on Emerging Attack Platforms for Spam, Viruses and Malware*, Checked September 2010 http://www.ironport.com/securitytrends/.

[3] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and Stefan Savage, Proceedings of the 15th ACM CCS, *Spamalytics: An Empirical Analysis of Spam Marketing Conversion*, October 2008, Checked September 2010 http://www.icsi.berkeley.edu/pubs/networking/2008-ccs-spamalytics.pdf.

[4] J. Klensin, *RFC 5321: Simple Mail Transfer Protocol (SMTP)*, October 2008, Checked September 2010 http://tools.ietf.org/html/rfc5321.

[5] Wikipedia, *Greylisting*, 4th February 2010, Checked September 2010 http://en.wikipedia.org/w/index.php?title=Greylisting&oldid=341929685.

[6] Rhyolite Software LLC, *Distributed Checksum Clearinghouses*, Summer 2008, Checked September 2010 http://www.dcc-servers.net/dcc/.

[7] Dr. S. Ritterbusch, *Die Mathematik des Bayes Spamfilters*, Checked September 2010 http://www.math.kit.edu/iag1/~ritterbusch/seite/spam/de.

[8] Cisco Systems Inc., *Description of the SenderBase Network*, Checked September 2010 http://www.senderbase.org/about.

[9] Cisco IronPort, *The SenderBase Network - Overview*, Checked September 2010 http://www.ironport.com/pdf/ironport_senderbase_overview.pdf.

[10] Federal Trade Commission, *The CAN-SPAM Act: A Compliance Guide for Business (Facts for Business)*, September 2009, Checked September 2010 http://www.ftc.gov/bcp/edu/pubs/business/ecommerce/bus61.shtm.

[11] The German Federal Ministry of Justice, *Gesetz gegen den unlauteren Wettbewerb*, 2004, Checked September 2010 http://www.gesetze-im-internet.de/englisch_uwg/index.html.

[12] The Spamhaus Project, *TRegister of Known Spam Operations (ROKSO)*, February 2010, Checked September 2010 www.spamhaus.org/rokso/.

[13] J. Rosenberg, et. al., *RFC 3261: Session Initiation Protocol (SIP)*, June 2002, Checked September 2010 http://tools.ietf.org/html/rfc3261.

[14] Heise-Newsticker, *Erste größere Attacke gegen deutsche VoIP-Nutzer*, September 2008, Checked September 2010 http://www.heise.de/security/meldung/Erste-groessere-Attacke-gegen-deutsche-VoIP-Nutzer-207400.html.

[15] D. Jansen, *Einfuehrung in die Netzwerkanalyse: Grundlagen, Methoden, Forschungsbeispiele*, August 2006 (3. Auflage), Vs Verlag,
ISBN-13: 978-3531150543.

[16] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining (Adaptive Computation and Machine Learning)*, October 2001, The MIT Press ISBN.

[17] H.-Y. Lam and D.-Y. Yeung, *A Learning Approach to Spam Detection based on Social Networks*, 2007, CEAS07 Fourth Conference on Email and AntiSpam, September 2010 http://www.ceas.cc/2007/papers/paper-81.pdf.

[18] P. O. Boykin and V. Roychowdhury, *Personal Email Networks: An Effective Anti-Spam Tool*, April 2005, IEEE Computer, Vol. 38, No. 4, pages 61-68.

[19] The Spamhaus Project, *The World's Worst Spam Producing Countries*, March 2010, Checked September 2010 http://www.spamhaus.org/statistics/countries.lasso.

[20] Wikipedia Encyclopedia, *Signaling System 7*, April 2010, Checked September 2010 http://en.wikipedia.org/w/index.php?title=Signaling_System_7&oldid=346781094.

[21] J. A. Muir and P. C. van Oorschot (Carleton University, Technical Report), *Internet Geolocation and Evasion*, April 2006, Checked September 2010 http://www.ccsl.carleton.ca/~jamuir/papers/TR-06-05.pdf.

Formalisation of Mediation Protocol for Web Services Composition with ACME/ARMANI ADL

Raoudha Maraoui
Faculty of Sciences of Monastir
Tunisia
maraoui.raoudha@gmail.com

Mohamed Graiet
MIRACL, ISIMS, Tunisia
Mohamed.graiet@imag.fr

Mourad Kmimech
MIRACL, ISIMS, Tunisia
mkmimec2@iutbayonne.univ-pau.fr

Mohamed Tahar Bhiri
MIRACL, ISIMS, Tunisia
tahar_bhiri@yahoo.fr

Béchir El Ayeb
Faculty of Sciences of Monastir
TUNISIA
Ayeb_b @yahoo.fr

***Abstract*—SOA (Service Oriented Architecture) defines a new Web Services cooperation paradigm in order to develop distributed applications using reusable services. The handling of such collaboration has different problems that lead to many research efforts. In this paper, we address the problem of Web service composition. Indeed, various heterogeneities can arise during the composition. The resolution of these heterogeneities, called mediation, is needed to achieve a service composition. In this paper, we propose a sound approach to formalize Web services composition mediation with the ADL (Architecture Description Language) ACME. To do so, we first model the meta-model of composite service manager and mediation. Then we specify semi formal properties associated with this meta-model using OCL (Object Constraint Language). Afterwards, we formalize the mediation protocol using Armani, which provides a powerful predicate language in order to ensure service execution reliability.**

Keywords- Web Services Composition; Mediation; Transactional Web Services; Formalization; ACME/ARMANI ADL; reliability.

I. INTRODUCTION

The recent evolution of Internet technologies expands the role of the Web from a simple data support to a middleware for B2B (Business to Business) applications. This new Internet wave is guided by the concept of Web services. However, it is necessary to combine a set of atomic service to answer for more complex requirements [1]. The problem we are interested in is how to ensure a reliable Web service composition. By reliable, we mean any compositions where all instances are correct in the sense that they meet designer's requirements, and especially in case of component failure. But, despite the organization of the composition into steps, the Web services composition has many heterogeneity problems. The resolution of these heterogeneities, called mediation, is needed to achieve a reliable service composition. In this paper, we formalize a reliable service composition based on non-functional Web Services properties. To do so, we describe the protocol mediation using the ACME of architectural concept style and Armani [17], to detect architectures software disparities.

This paper is organized as follows. In Sections 2 and 3, we present the Web services modeling related works, and then describe our formalization approach of Web services composition, respectively. In Section 4, we study the Web services meta-model and we propose a new composite service meta-model. Afterwards, we present in Section 5 the informal and semiformal specification of transactional properties. In Section 6, we propose a new architecture. In Section 7, we present our case study: a travel agency application. Finally, we conclude the paper by summarizing the main results and describing our futures woks.

II. RELATED WORKS

Many efforts have been provided to allow a usable and acceptable Web services composition. These efforts have been implemented by several composition standard and approaches and vary between those that aspire to become industry standards to those that are much more abstract. There are several formalisms for modeling Web services composition. We can cite the Petri nets, contracts, graphs, [2] , [3] UML (Unified Modeling Language), and ADLs. Each approach has advantages and disadvantages. For example, modeling using Petri nets is sound, has an intuitive graphical representation, and very visual. This approach is relevant but does not use the power of Petri nets for the composition verification. It does not model inputs and outputs of services. Another approach [4] used the concept of contracts, which

ISBN: 978-1-61208-105-2

are graph transformations rules. They are specified by assertions expressing the the parties' obligations and rights. This approach remains inadequate if we want to make a dynamic or semi-automatic service composition. In our work, we try to formalize Web services compositions with ADL, an architecture description language which describes such formal process. It is recognized that UML does not describe software architecture within the meaning of ADL [5]. Even if you can use profiles to give the ADL characteristics [6], this approach limits his strong reusability property. Therefore, our approach is inspired by ADL. Yet most approaches ignore the specification of non-functional properties such as security, dependency, or transaction management. We try in this work to formalize Web services compositions with an architecture description language by implementing the protocol mediation and encouraging a large proportion of non-functional properties namely transaction management. In the next section, we present our method of formalization that derives from an MDE (Model Driven Engineering) approach which is based on the use of the ADL ACME / Armani.

III. PROPOSED APPROACH

In order to check the Web services composition, we use an MDE-based approach (Fig. 1).

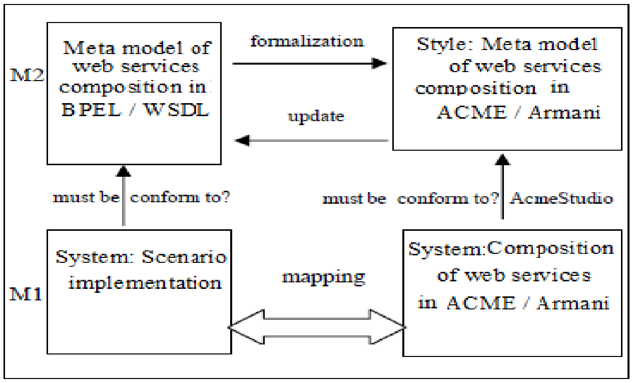

M2
Meta model of web services composition in BPEL / WSDL
formalization
update
Style: Meta model of web services composition in ACME / Armani
must be conform to?
must be conform to? AcmeStudio
M1
System: Scenario implementation
mapping
System:Composition of web services in ACME / Armani

Figure 1. An overview of our services composition checking approach applied to the web service model.

Indeed, we distinguish two levels M2 and M1. The M2 level describes the Web services composition meta-model and its formalization in Acme/Armani while the M1 level describes the services model. We aim to check its conformity with its meta-model.

For that purpose, we transform this service model into Acme/Armani through the M2 level formalized in Acme/Armani. The M1 level is conform to the M2 level if it checks the coherence of rules described in the M2 level and the specific rules described in the M1 level. This is checked thanks to the AcmeStudio environment, which enables the evaluation of the Armani constraints [8].

Indeed, to achieve the formalization of web service composition in ACME and check the consistency of this composition, we describe the meta-model of web service composition (M2) using the concept of architectural style of ACME. A web services composition in M1 is described using the concept of ACME system. Level M1 is said to be conform to M2 if it satisfies the consistency rules described in M2 in addition to specific rules outlined in M1.

Our approach of components assembly checking has several advantages:

- It could be applied to several components models.
- It allows validating (see the labeled arrow updated on Fig. 1) the coherence rules described on the M2 level of the considered component model. Indeed, the completeness of these rules must be considered as well on the theoretical level as on the practical level through a test activity. Representative test models based on functional testing can be established in order to validate the coherence of the suggested rules thanks to the AcmeStudio execution environment.
- The expressiveness power of Acme/Armani is higher than the UML/OCL which is considered as an alternative to our approach.

IV. META-MODELING OF COMPOSITE SERVICE

In this section, we offer an overview of the services composition that defines a meta-model of composite service. This meta-model reifies all reliable characteristics of a service composition. It identifies their interdependencies, allows a comprehensive understanding of the mechanism composition and provides the ability to reuse our meta-model, which is independent of application domains or specific technologies. The construction of our meta-model is based on the modification of various properties of a service composition. Each of its properties is clearly identified and defined. Moreover, our meta-model is built as an extension of the meta-service model of OASIS (Organization for the Advancement of Structured Information Standards) [10] and W3C (World Wide Web Consortium). As, an atomic service, a composite service inherit all properties [11]. A composite service is a composition of one or several services: services: the services' constituents.

We allocate these services constituents to business services and management services of the composite (Fig. 2):

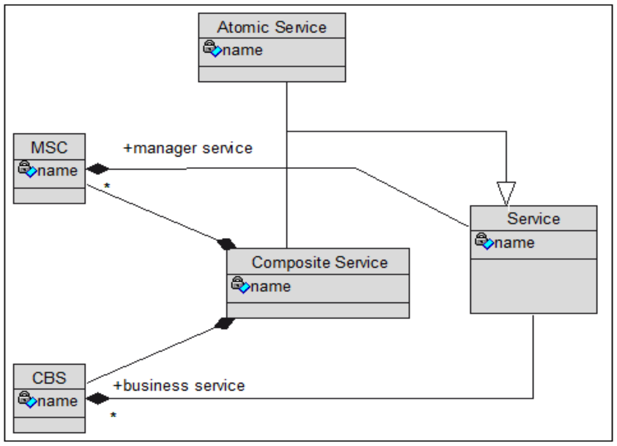


Figure 2. A meta-model of composite service.

- Business services: These services provide their functionality without global knowledge of the composition. The business services are grouped in the composite service business or CBS.
- Manager Services Composite MSC: These are specialized services in the management of the composition logic. They manage the other components and services, which have a comprehensive understanding of the composition. The service managers are grouped in the manager service composite or MSC [12].

The MSC meets all services managers who are totally transparent to users. It is the invisible part of the composite, in charge of the composition logic. Inspired by services composition existing work, we can abstract four main roles that are described in Fig. 3:

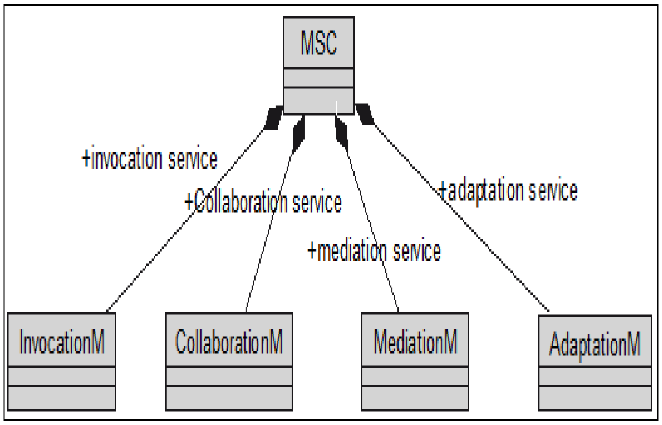


Figure 3. A meta-model of Manager Service Composite.

We focus mainly on the definition of MSC and more specifically on the mediation manager.

A. Web services mediation

The resolution of heterogeneities between Web services is critical to the achievement of the composition of these services. Indeed, the composition would lead most of the times to failure without a mediation between the functioning of services and data exchanged between them. In general, mediation is to resolve conflicts between stakeholders to ensure successful interactions. Furthermore, no current approach offers a comprehensive solution to the mediation protocol for Web services composition. Our work aims to answer to this lack of clarity. We are interested in a classification proposed by [7]:

- The integration level of Web services: aims to resolve all the heterogeneities between the non-functional properties.
- The adaptation level interface: aims to resolve all the heterogeneities of the service properties described in a WSDL document
- The data level mediation: aims to resolve all the heterogeneities of the service of data exchanged between the composed Web services.

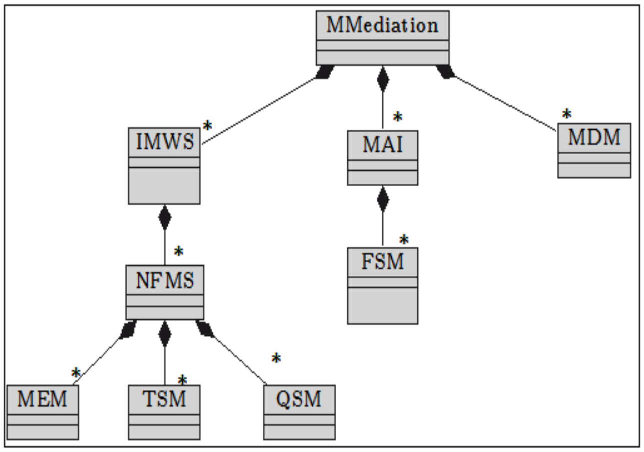


Figure 4. A meta-model of manager mediation

However, we can go further into the analysis of the meta-model and extract other properties to solve all kinds of heterogeneity. These properties included the specific non functional properties such as:

- The sequences message exchange (MEM).
- The transactional properties: They are managed by the Transaction Service Manager (TSM).
- Quality of service (QSM): This term includes nonfunctional properties, such as availability, speed, and cost

B. The transactional patterns mediators

Moreover, we introduce in our mediation the concept of transactional pattern, which is a point of convergence between workflow patterns and ATMs (Advanced Transactional Models) [14], one can express the logic of business processes, and the other can define the reliability of the executions. We also show their use to define and ensure service reliability compounds. For example, we use the ANDJoin pattern [15] that describes a class of interactions where a service will be activated after the termination of other services (Fig. 5).

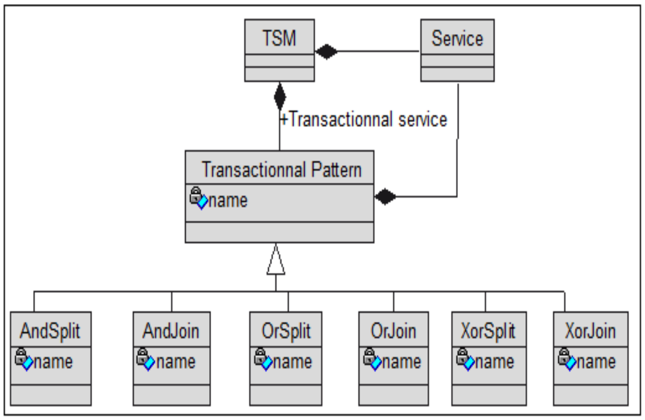


Figure 5. Transactional patterns.

ISBN: 978-1-61208-105-2

C. *Composition of transactional Web services*

In this section, we show how we combine a set of transactional Web services to offer a new more complex value-added service. To manage the coordination of service components of a Transactional Composite Service (TCS), a composed service defines preconditions for external transitions (Fig. 6). These preconditions specify how the service responds to state of other services and how it can influence their behavior. Thus, a transactional web service can be set up as the couple of all components of its services and all preconditions set on their external transitions [13].

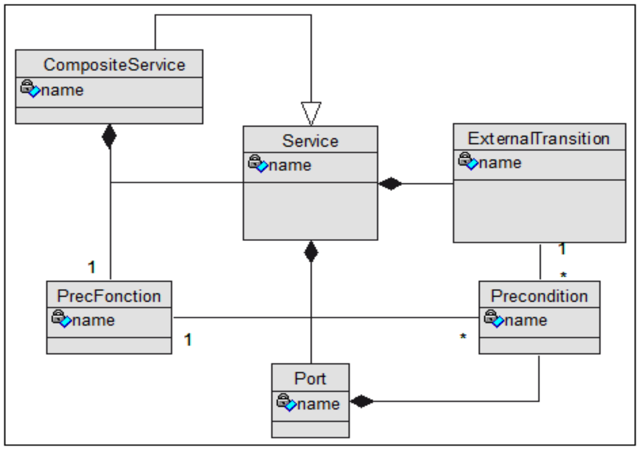


Figure 6. Definition of a transactional composite service.

Then, we show in Fig. 7 how these preconditions can express a level of abstraction above dependencies between services. These dependencies in turn define the control flow and the transactional flow of the service compound.

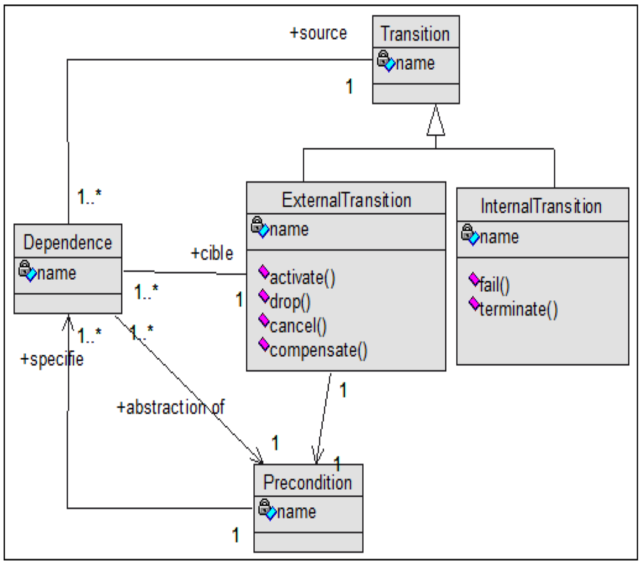


Figure 7. The preconditions to express a level of abstraction above dependencies between services.

The internal transitions that we consider in our approach are fail (), terminate () and external transitions are activate (), drop (), cancel () and compensate ().

- *Dependencies between services components of a TCS:*

The preconditions express the form of dependency relations (successions, alternative, etc) between service components, that is to say how services are coupled and how the behavior of some services can influence the others. In general, a dependency of S1 on S2 exists if the initiation of a transition (internal and external) of S1 can be triggered from external transition of S2. The management of these dependencies includes the definition of 5 types of dependencies: activation, alternative, abandonment, compensation and cancellation.

V. SPECIFYING PROPERTIES OF WEB SERVICES FROM THE PROPOSED META-MODEL

A. *Non functional transactional properties*

It is necessary to make a choice among various nonfunctional properties for each system as it is often impossible to fully satisfy all. We have chosen to highlight the transactional approach by the interest it provides. In addition if you want to move towards more rigorous, it is possible to complete this vision chart needs through the appropriate use of pre and post conditions expressed textually with OCL [16].Thus the semi-formal specification of some OCL constraints described informally as follows:

- In the component type AndSplit mediator, any port service type must have a pre-condition equal to active.

Context MedAndSplit

InvPortServiceTerminate:Self.ports → oclIsTypeOf

(PortTWSService)implies Forall(p : PortWSServicejp:P rec

== activate)

B. *Structural properties*

Although our framework focuses on the specification of transactional properties related to non-functional mediation for web service composition, it is clear that the formalization of these properties generates other properties related to the structure and the operation of composed Web services. Among the structural properties of our style, we can cite:

- Every component in the system must satisfy to be made a Web service client, mediator or service.

Context System

InvServiceType:Self.service → oclIsTypeOf

(CompTWSClient) ORoclIsTypeOf(CompTWSService)

ORoclIsTypeOf(CompTWSMediator)

- In the Component CompTWSMediator, there must at least two ports, a port of entry and an output port.

Context CompTWSMediator
Inv AtLeast2ports:Self.port → size () >=2

C. Fonctional properties

A specific style shows sequences of operations. Among the functional properties of our style, we can cite:

- A mediator AndSplit type specifies tha a set of services will be activated after the termination of another service.

Context AndSplit
Pre:SCN.PortTWSClient.Prec==terminate
Post:FB.PortTWSService.Prec==activateAND
HR.PortTWSService.Prec==activate

VI. A NEW ARCHITECTURE STYLE:WSM

By studying the deployed systems, there is a number of architecture which are not limited to one style only use. This is the case for our style that works in client/server roles style and symmetrical drawing some specific pipe/filter style. This WSM style (Web Service Mediation) has three components: clients, servers and mediators. They all play the role of a service with certain features. The Ombudsman is the link between the actors who are clients and servers. Clients and servers can communicate only with the mediators. There is no direct connection between the different clients of the system or between different servers. They use SOAP (Simple Object Access Protocol) as the communication protocol in order to exchange structured data regardless the programming languages or operating systems. The WSM style is an interaction model application implementing connections to perform a Web services composition. This style is not specific to a domain, it is rather generic in order to increase the level of reuse and adapt it to any field. In fact this advantage goes to the ACME ADL that allows these users to formalize their own styles.

A. The ADL ACME

The ADL ACME [17] [18], developed at Carnegie Mellon, is a common foundation for architecture description languages. It aims to enable the exchange of architectural specifications across different ADLs. ACME is based on seven types of entities to describe architecture: components, connectors, systems, ports, roles, representations, and rep-maps (map representation). Moreover, it provides a rather powerful predicates language called Armani [19] with functions appropriate to the field of software architecture. The Armani language allows describing architectural properties in the invariant or heuristics forms attached to any architectural element (component, family, system, connector, etc.). Such properties are achievable within the AcmeStudio environment [20]. In the same way, the ADL Acme supports the type concept. One can define types of architectural elements (component type, connector type, role type, port type and style type). The concept property of Acme used in the type and instance levels allows attaching nonfunctional properties to the architectural elements. Lastly, Acme provides basic types (int, float, boolean and string) and type builders (enum, record, set and sequence).

B. Formalisation of the mediation service for the Web services composition with ACME

Our work began with the improvement of an existing style. We have studied the work of [21] dealing Web services composition without mediation approach, or control over the execution of flow of services. The added mediation approach is used to increase the interactions reliability between services and ensured proper implementation through transactional patterns and connectors that represent mediators. We define in our WSM style five types of connectors that inherit from ConnTWS which is connector type of Web service and represents the five types of dependencies mentioned above. The connector ConnTWS contains rules that detect inconsistencies and show that the connector should have only two roles. Fig. 8 shows an example of an activation connector, which specifies a fundamental property to ensure the activation dependency. This property ensures that for any role r1 attached to a port P1, and for any role r2 attached to a port p2 , the two roles are different, the port p1 must be a precondition equal to ”terminate”. Therefore, to ensure this property the port p2 should be equal to a pre-condition ’activate’ and vice versa.

```
// Definition of Activation Connector
Connector Type ConnTWSAct extends ConnTWS with
{
Rule CondActivation = invariant forall r1 : Role in
self.ROLES |
Forall r2 : Role in self.ROLES |
Forall p1 : PortTWSClient in r1.ATTACHEDPORTS |
Forall p2 : PortTWSService in r2.ATTACHEDPORTS|
(r1 != r2 AND attached (r1, p1) AND attached (r2, p2))
-> (p1.Prec == terminate AND p2.Prec == activate) OR
(p2.Prec == terminate AND p1.Prec == activate) ;}
```

Figure 8. The ACME descriptions of the activation connector.

In addition, this style cans be used to detect the mismatches between web services. Thus, rules are defined, illustrated in Fig. 9. The first rule states that all the elements found in a system of this style must meet the requirement of being one of three component types CompTWSClient, or CompTWSService or CompTWSMediateur.

The second rule checks that if two components are connected one of them must be of mediator type and the third shows that the control flow is formalized as a composition between the AndSplit mediator on one hand and activation connector on the other. Indeed, if the component AndSplit exists it must necessarily be attached to an activation connector.

VII. CASE STUDY

We present in this section a scenario to show how this style can be used in ACME Studio to detect inconsistencies. The example shows a web travel organization application. The client specifies its requirements in terms of destination and choice of accommodation through the activity ”Specification of Client Needs” (SCN). This specification is then passed through mediation service AndSplit that describes how the services “Flight Booking” (FB) and ”Hotel Reservation” (HR) will be activated after SCN termination.

The organization Travel Service Online (TSO) described above, specifies a dependency of activation between SCN and HR services, denoted depAct (SCN,HR) under the activation condition, HR CondAct (HR) = SCN.terminate(). So HR will be activated after the termination of SCN. But the client component SCN has only Client type port according to the WSM specification style. In addition the mediator, AndSplit has an input service type port that can be assembled with the client port component SCN having a pre-condition “activate”.

```
143./// Configuration of few rules
144.Rule rule33 = invariant forall comp: Component in self.
    COMPONENTS satisfiesType(comp, CompTWSClient)
    OR satisfiesType(comp, CompTWSService) OR
    satisfiesType(comp, CompTWSMediateur)
145.Rule rule34 = invariant forall c1: Component in self.
    COMPONENTS | forall c2: Component in self.
    COMPONENTS|connected(c1,c2) ⟶
    (satisfiesType(c1, CompTWSClient) AND
    satisfiesType(c2, CompTWSMediateur)) OR
    (satisfiesType(c1, CompTWSService) AND
    satisfiesType(c2, CompTWSMediateur)) OR
    (satisfiesType(c2, CompTWSClient) AND
    satisfiesType(c1, CompTWSMediateur)) OR
146.(satisfiesType(c2, CompTWSService) AND
    satisfiesType(c1, CompTWSMediateur))
147.Rule rule35 = invariant exists c: Component in self.
    COMPONENTS  | declaresType(c, MedAndSplit) AND
    forall conn : ConnTWS in self.CONNECTORS |
    attached(c, conn)  ⟶   (satisfiesType
    (conn,ConnTWSAct));   }
```

Figure 9. The The ACME descriptions of few rules.

It also has two ports as client having”terminate” as pre-condition. A fundamental property was described in the activation connector and specifies that any assembly with a client port service must satisfy a dependency of activation, i.e., a precondition ”activate” and pre-condition ”terminate” on both sides of the connected ports. So given these properties checked during assembly AndSplit mediation service that has a service port ”activate” pre-condition with the SCN client service, it can only have one client port pre-condition ” terminate”. As a result, we check the function of a listed mediator AndSplit, which is to complete a service that is SCN client service. On the other side the mediator has the same role to enable other service that are the HR Service and FB using the same process as the AndSplit mediation service which can be linked with an activation connector. However, the different dependencies of activation, alternative, and cancellation have been fulfilled with the ADL ACME / Armani and fostered a reliable Web service composition through mediation. We note that Acme Studio puts warning triangles in architecture during the inconsistency detection process. These triangles are superimposed on pre signaling components or connectors, which indicate that one or more constraints are not met. In this case, it means that an architecture inconsistency has been detected and is localized around the connector or component as in Fig. 10.

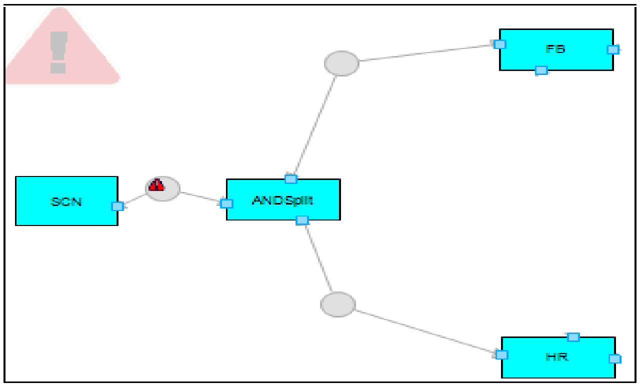
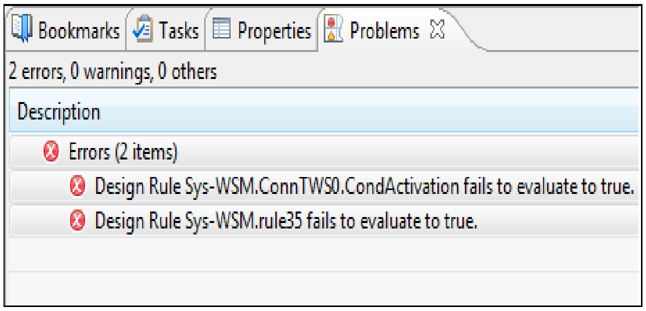


Figure 10. The initial system architecture with warning triangles showing where mismatches have been detected.

A triangle does not indicate what type of asymmetry is. This is why we should select the connector in question to find the reported failed rules. Fig. 11 shows this point of view of the activation connector between FB and services ANDJoin. The rule states that the activation connector fail to evaluate to true as shown in the figure and as consequence the activation dependence is failed, which then leads to failure of the entire system.



2 errors, 0 warnings, 0 others

Description

Errors (2 items)

Design Rule Sys-WSM.ConnTWS0.CondActivation fails to evaluate to true.

Design Rule Sys-WSM.rule35 fails to evaluate to true.

Figure 11. The initial system architecture with warning triangles showing where mismatches have been detected.

To process the ANDJoin mediator, it is necessary to satisfy the activation condition in the connector between FB and ANDJoin. To correct the detected inconsistency, we have to establish a good activation condition between these components by associating a precondition to enable ConfirmReqReserv port of the ANDJoin component.

VIII. CONCLUSION AND FUTURE WORKS

This work presented in a general framework to ensure a safe design and execution of software architectures specifically the web services composition. We could formalize this composition mechanism by implementing the mediation protocol and ensuring reliability advocated by specifying non-functional properties. To do so we use Acme to check assembling consistency of Web service composition. We address this issue by describing the Web services composition Meta-model (M2 level) using Acme style architecture. The checking of the structural and non-functional properties of the composition models exploits the AcmeStudio features of verifying invariants of an Acme model. In our future works we are considering the following perspectives:

- Using existing techniques developed by the Semantic Web initiatives to promote the automation of messages and the selection of mediator models.
- Using external analysis tools associated to AcmeStudio environment in order to reason on Web services composition structures: processing global properties from local properties.
- Developing systematic translation rules of Web service composition architecture through the M2 level provided in Acme style (WSM style) which would call upon an MDE approach.

REFERENCES

[1] F. Curbera, I. Silva-Lepe, and S. Weerawarana: On the integration of heterogeneous web service partners, IBM T. J. Watson Research Center, August, 2001. [retrieved: June, 2010]. http://www.research.ibm.com/people/b/bth/OOWS2001/curbera.pdf

[2] R. Hamadi and B. Benatallah: A Petri Net-based Model for Web Service Composition, in School of Computer Science and Engineering, The University of New South Wales, In Proceedings of the 14th Australasian Database Conference (ADC'03), CRPIT 17, pp. 191–200, Australian Computer Society, Adelaide, Australia, February, 2003.

[3] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella: Automatic Composition of e-Services, Proceedings of the First International Conference on Service-Oriented Computing (ICSOC), pp. 43–58, 2003.

[4] R. Heckel: Towards contract based testing of web service, in Electronic Notes in Theoretical Computer Science 116, pp. 145–156, 2005.

[5] N. Medvidovic and N. R. Taylor: A classification and comparison framework for software architecture description languages. IEEE Transactions on Software Engineering, 26 (1): pp. 70–93, January 2000.

[6] M. Graiet: Contribution à une démarche de vérification formelle d'architectures logicielles, thèse de doctorat, Université Joseph Fourier, 25 Octobre 2007.

[7] M. Mrissa: Médiation Sémantique Orientée Contexte pour la Composition de Services Web, thèse de doctorat, Université Claude Bernard Lyon I UFR Informatique, pp. 15–36, 2007.

[8] M. Kmimech, M. Tahar Bhiri, M. Graiet, and P. Aniorté: Checking component assembly in Acme: an approach applied on UML 2.0 components model, In 4nd IEEE International Conference on Software Engineering Advances (ICSEA'2009), Portugal, IEEE Computer SocietyPress, Septembre 2009.

[9] M. Rouachid: Une approche rigoureuse pour l'ingénierie de compositions de services Web, thèse de doctorat, Université Henri Poincaré, Nancy, pp. 31–34, 2008.

[10] OASIS (2008), Service component architecture assembly model specification version 1.1. http://www.oasis-opencsa.org/. [retrieved: August 10, 2010].

[11] OpenGroup (2009). Soa source book. http://www.opengroup.org/projects/soa-book. [retrieved: June, 2010].

[12] M. Oussalah: Vers une meilleure compréhension de la composition de services par Méta Modélisation d'un service composite, 4th Francophone Conference on Software Architectures, CAL'2010, Pau-Paris, March 2010.

[13] S. Bhiri, C. Godart and O. Perrin: Patrons transactionnels pour assurer des compositions fiables de services web, Technique et Science Informatiques 28(3): pp. 301–330, 2009.

[14] S. Bhiri: Reliable Web services composition using a transactional approach, International Conference on e-Technology, e-Commerce and e-Service (EEE'05): pp. 22–30, 2005.

[15] W. M. P. van der Aalst, A. P. Barros, A. H. M. ter Hofstede, and B. Kiepuszewski: Advanced Workflow Patterns. In O. Etzion and Peter Scheuermann, editors, 5th IFCIS Int. Conf. on Cooperative Information Systems, number 190 in LNCS, pp. 18–29, Eilat, Israel, September 6–8, 2000.

[16] J. Warmer and A. Kleppe: The Object Constraint Language: Precise Modeling with UML, AddisonWesley, 1998.

[17] D. Garlan, R. T. Monroe, and D. Wile: Acme: An Architecture Description Interchange Language, Proceedings of CASCON 97, Toronto, Ontario, November, pp. 169–183, 1997.

[18] D. Garlan, R. T. Monroe, and D. Wile: Acme: Architectural Description of Composed-Based Systems, Gary Leavens and Murali Sitaraman, ed.s Kluwer, 2000.

[19] D. Garlan, R. Monroe, and D. Wile: Acme: Architectural Description of Component-based. Capturing software architecture design expertise with Armani. Technical Report CMU-CS, pp. 98–163, Carnegie Mellon University School of Computer Science, 2001.

[20] Group 2006, http://www.cs.cmu.edu/~acme/Acme Studio/ [retrieved: August 12, 2010].

[21] C. Gacek and C. Gamble: Mismatch Avoidance in Web Services Software Architectures, Journal of Universal Computer Science, vol. 14, no. 8, pp. 1285–1313, 2008.

WEB Services for Ubiquitous Mobile Device Applications

Mihai Barbos
IT&C Department
SC IPA SA
Bucharest, Romania
e-mail: mihaibarbos@ipa.ro

Eugen Pop
IT&C Department
SC IPA SA
Bucharest, Romania
e-mail: epop@ipa.ro

Abstract— A WEB service is as a self-describing, self-contained software module available via a network, such as the Internet, which completes tasks, solves problems, or conducts transactions on behalf of a user or application. WEB services constitute a distributed computer infrastructure made up of many different interacting application modules trying to communicate over private or public networks (including the Internet and WEB) to virtually form a single logical system. Mobile WEB services target embedded devices. In other words, they enable handheld devices to interact with servers in a standardized way, regardless of operating systems, platforms, and programming languages. WEB services provide good opportunities for developing ubiquitous mobile client applications, allowing the delivery of information to users anytime and from anywhere. This paper brings to focus some general considerations required in designing and building WEB services that target ubiquitous mobile applications as consumers. The research work presented here was carried out under the UbiPOL FP7 research project, which aims to develop a ubiquitous platform for policy making, funded within the grant agreement nr. 248010. The main achievement of this work is the analysis of several WEB service frameworks and choosing the most appropriate one for developing the UbiPOL platform.

Keywords - WEB services; mobile devices; ubiquitous applications

I. INTRODUCTION

The “always on” vision of mobile Internet access has became a reality with the nearly ubiquitous coverage provided by cellular networks. Communications speeds have also increased significantly with the advent of 3G mobile networks that enable data rates supportive of real time video. In addition, using the competing access technologies such as wireless, the user can experience Internet data rates similar to those available with broadband connectivity in the fixed networks [4].

The increasing use of mobile terminals and infrastructure makes it possible the communications and information access from any location at any time. The convergence of mobile and WEB service technologies enables new services and business models, and accelerates the development of mobile and fixed Internet technologies [1]. The mobile industry is poised to take advantage of the benefits of interoperability that WEB services provide [2]. Interoperable messages can reduce the time and costs of business integration, creating opportunities for the adoption of WEB services technologies.

The present paper describes the technology and the building blocks needed to be put together in mobile networks, that can wirelessly delivered WEB content [7]. Included are extensive coverage of the network elements, languages used to represent browser content, communication protocols, network services and related software components that are used in the operation of such networks [5], [8]. The UbiPOL platform services will include user location tracking, security schemes, content personalization approaches, privacy mechanisms, etc. .

While on the move, a mobile user faces many challenges such as mobile terminal’s limited screen size, restricted input capabilities, battery power constrains and air time costs. There is where knowledge of a user context can be leveraged to drive and personalize the interaction between user and the Internet server, so as to ease the communication exchange and focus the delivery of WEB content to the user.

First, the paper presents and analyzes the tendencies and characteristics of the mobile device market. Then, the JAVA ME platform is presented as a very useful and performance tool for mobile application development. In fact, JAVA ME seems to be the nost ubiquitous application platform for mobile devices, which complies with the UbiPOL objectives. Following these considerations, the WEB services clients in JAVA ME are also presented. Several Java WEB service stacks and frameworks that will be considered in the implementation of UbiPOL are presented next. Finally, a section for conclusions is present at the end of the paper.

II. THE MOBILE DEVICE MARKET

The mobile devices market trends are closely related to the significant increase of the bandwidth necessity and data traffic on the cellular or wireless channels. Nowadays, the network and service infrastructure domain is facing many important changes like:

- Explosive growth of data rates and capacities;

- The emergence of massive data and content delivery and consumption;

- Evolution towards a converged architecture that has dramatically increased the permutations and combinations of services and usages between people, devices, media, and even between real and virtual worlds;

- User involvement in defining her/his communication sphere has led to much more customization, personalization and users becoming content producers.

An explosion in number of mobile Internet devices is expected, evolving towards trillions of connected devices, M2M, within the Internet of objects.

Globally, mobile data traffic will double every year through 2014, increasing 39 times between 2009 and 2014. Mobile data traffic will grow at a compound annual growth rate (CAGR) of 108 percent between 2009 and 2014, reaching 3.6 exabytes per month by 2014 [24]. The UbiPOl project aims to use this opportunity to involve the citizens in the policy making process.

According to Gartner Inc. Report, the worldwide mobile phone sales totalled 286.1 million units in the second quarter of 2009, a 6.1 per cent decrease from the second quarter of 2008, according to Gartner, Inc. [3] (see Table 1). Smartphone sales surpassed 40 million units, a 27 per cent increase from the same period last year, representing the fastest - growing segment of the mobile - devices market (see Table 2).

TABLE I. WORLDWIDE MOBILE TERMINAL SALES END USERS IN 2Q09 (THOUSANDS OF UNITS)

Company	2Q09 Sales	2Q09 Market Share (%)	2Q08 Sales	2Q08 Market Share (%)
Nokia	105,413.3	36.8	120,353.3	39.5
Samsung	55,430.2	19.3	46,376.0	15.2
LG	30,497.0	10.7	26,698.9	8.8
Motorola	15,947.8	5.6	30,371.8	10.0
Sony Ericsson	13,574.2	4.7	22,951.7	7.5
Others	65,260.2	23.0	57,970.6	19.0
Total	**286,122.7**	**100**	**304,722.3**	**100**

TABLE II. WORLDWIDE SMARTPHONE SALES TO END USERS IN 2Q09 (THOUSANDS OF UNITS)

Company	**2Q09 Sales**	**2Q09 Market Share (%)**	**2Q08 Sales**	**2Q08 Market Share (%)**
Nokia	18,441.0	45.0	15,297.9	47.4
Research In Motion	7,678.9	18.7	5,594.2	17.3
Apple	5,434.7	13.3	892.5	2.8
HTC	2,471.0	6.0	1,330.8	4.1
Fujitsu	1,249.0	3.0	1,071.5	3.3
Others	5,688.2	13.9	8,085.8	25.1
Total	**40,962.8**	**100.0**	**32,272.7**	**100.0**

As mobile devices continue to converge with consumer electronics, vendors find themselves (and their devices) locked in a battle over similar market segments, similar buyer demographics and similar product concepts. Mobile users have a wide choice of capable device types to fulfill their mobile communication and computing needs. Studies highlighed key device segment trends and market considerations across many mobile device categories, including mobile handsets and handset accessories. Since wireless connectivity continues to be incorporated into new device segments, the impact of emerging services and technologies is in close relation to their market potential in mobile devices [4].

The remarkable technical performances of the mobile devices make them available for a great variety of applications: social networking, consumer and, business applications, content & delivery platforms, messaging, browsers, operating systems (OS), and users interfaces (UI).

III. JAVA ME "THE MOST UBIQUITOUS APPLICATION PLATFORM FOR MOBILE DEVICES

Java Platform, Micro Edition (Java ME) provides a robust, flexible environment for applications running on mobile and other embedded devices—mobile phones, personal digital assistants (PDAs), TV set - top boxes, and printers. Java ME includes flexible user interfaces, robust security, built - in network protocols, and support for networked and offline applications that can be downloaded dynamically [5]. Applications based on Java ME are portable across many devices, yet leverage each device's native capabilities.

In order to provide ubiquitous deployment across different mobile operating systems and device types the Java ME platform was selected for development of front end components for the UbiPOL platform [6]. An inventory of mobile application development tools was realized. For each mobile application development tool, the following issues were analyzed and specified: system's requirements, installation procedures, available device emulators, Java ME API, etc.

The following tools are components of the mobile application development environment for UbiPOL:

- Java ME SDK 3.0, [15],[16];
- Nokia S60 3rd Edition SDK for Java, [17];
- BlackBerry JDE 5.0 [21];
- LG SDK 1.5 for the Java ME Platform ; [18];
- MOTODEV SDK for Java ME v3.0 [19];
- Samsung Java SDK 1.1.2, [20];
- Sony Ericsson SDK 2.5.0.6 for JavaME Platform[22].

The SDK from SUN was chosen as the default development tool for UbiPOL mobile applications identified as necessary. Compiling, building, running, testing and debugging mobile applications will be done using this tool. The other tools have been deployed only for test purpose, in order to ensure UbiPOL mobile applications compatibility with other device types from leading market manufacturers.

The Java ME Platform SDK 3.0 form SUN provides device emulation, a standalone development environment, and a set of utilities for rapid development of Java ME applications. On Windows, Java ME SDK 3.0 is the successor of the Java Wireless Toolkit 2.5.2 and Java Toolkit 1.0 for CDC. The Java ME SDK 3.0 is available for Windows XP and Vista 32 - bit, and for Mac OS.

The Java ME SDK 3.0 includes several emulators to allow running, testing and debugging applications under different scenarios (different device screen size, different input device configuration – key board, touch screen, both –, external events generator support –location events, device orientation change –, number of colours, different Java ME API support etc.) The device emulators available in the SUN's Java ME SDK are: ClamshellCldcPhone1, DefaultCldcJtwiPhone1, DefaultCldcJtwiPhone2, Default CldcMsaPhone1, DefaultCldcMsaPhone2, DefaultCldc Phone1, DefaultCldcPhone2, DefaultFxPhone1, DefaultFx TouchPhone.

The Java ME SDK from SUN offers a great number of Java ME API's, configurations and profiles as follows: configurations (CLDC 1.0, CLDC - 1.1), profiles (MIDP - 2.0), Java ME API's (ex. JSR 172 – WEB Services Specifications, JSR 179 – Location API, and so on), available for each emulator included in the SDK from SUN.

IV. WEB SERVICES CLIENTS IN JAVA ME

The J2ME WEB services API (WSA) extends the Java2 Platform, ME to support WEB services, and was developed within the Java Community Process, as JSR 172. The API's two optional packages standardize two areas of functionality that are crucial to clients of UbiPOL WEB services: remote service invocation and XML parsing.

WSA is designed to work with J2ME profiles based on either the Connected Device Configuration (CDC) or the Connected Limited Device Configuration (CLDC 1.0 or CLDC 1.1). The remote invocation API is based on a strict subset of J2SE's Java API for XML - Based RPC (JAX - RPC 1.1), with some Remote Method Invocation (RMI) classes included to satisfy JAX - RPC dependencies. The XML - parsing API is based on a strict subset of the Simple API for XML, version 2 (SAX2).

The core specifications and application - level protocols that define WEB services are promoted by the WEB Services Interoperability Organization (WS - I), and governed by the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS) [8]. The four key standards that specify how to create, deploy, find, and use WEB services, are presented in the following table.

TABLE III. WEB SERVICES STANDRADS

WEB services standards	**Description**
Simple Object Access Protocol (SOAP) 1.1	Defines transport and data encoding
WEB Services Definition Language (WSDL) 1.1	Defines how remote services are described
Universal Description, Discovery, & Integration (UDDI) 2.0	Defines how remote services are discovered
Extensible Markup Language (XML) 1.0, and XML Schema	Defines the Extensible Markup Language (XML) and XML Schema

The goal of WSA is to integrate fundamental support for WEB services invocation and XML parsing into the device's runtime environment, so developers don't have to embed such functionality in each application.

To make the interpretations of the standards easy, WS - I has defined a set of conformance rules called the WS - I Basic Profile, version 1.0. JSR 172 conforms to the Basic Profile. JSR 172 specifies standardized client - side technology to enable J2ME applications to consume remote services on typical WEB services architectures, as Figure 1 illustrates:

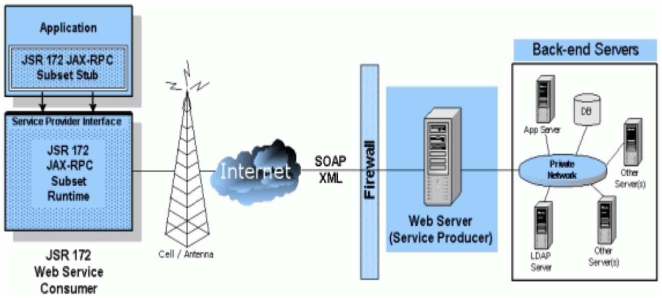

Application
JSR 172 JAX-RPC Subset Stub
Service Provider Interface
JSR 172 JAX-RPC Subset Runtime
JSR 172 Web Service Consumer
Cell / Antenna
Internet
SOAP XML
Firewall
Web Server (Service Producer)
Back-end Servers
App Server
DB
Private Network
Other Server(s)
LDAP Server
Other Server(s)

Figure 1. J2ME in a typical WEB service architecture

At a high level, this WEB service architecture has three elements:

- A network-aware application residing on a WSA-enabled wireless device. The application includes a JSR 172 stub that uses the JSR 172 runtime to communicate with the network.

- The wireless networks, the Internet and the corresponding communication and data - encoding protocols, including binary protocols, HTTP, and SOAP/XML.

- A web server, acting as the service producer, typically behind one or more firewalls and a proxy gateway; the web server often provides access to back - end applications and servers on a private network [7].

A typical JSR 172 based application is a smart client based on the Mobile Information Device Profile (MIDP) or the Personal Basis Profile (PBP), with business - specific logic, user interface, persistence logic, and life - cycle and application - state management. To handle XML documents, the application can employ the JAXP subset API. To consume WEB services, it can use the JAX - RPC subset API, employing JSR 172 stubs and the runtime. In devices such as cell phones, typically the application and the JSR 172 stub reside in the device's memory, while all the JSR 172 elements, along with the underlying profile and configuration, are embedded in the device itself.

At the center of JSR 172 operations is the runtime, with its service provider interface, which enables the stubs to perform all the tasks associated with invoking an RPC service endpoint:

- Set properties specific to an RPC invocation;
- Describe the RPC invocation input and return values;

- Encode input values;
- Invoke the RPC service endpoint;
-decode and return to the application any values that the service endpoint returns.

In the Figure 2 is presented how a typical JSR 172 based application is organized.

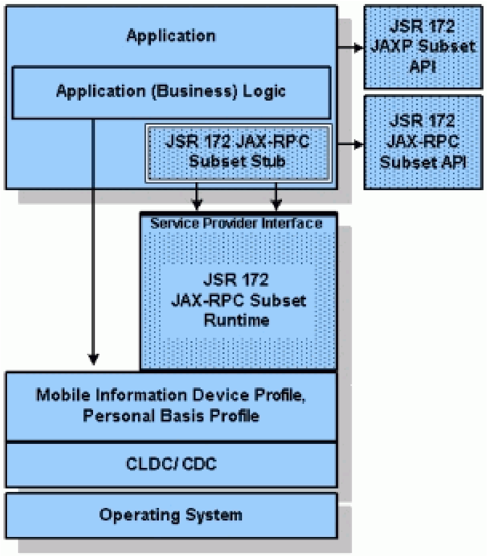

Application
Application (Business) Logic
JSR 172 JAX-RPC Subset Stub
JSR 172 JAXP Subset API
JSR 172 JAX-RPC Subset API
Service Provider Interface
JSR 172 JAX-RPC Subset Runtime
Mobile Information Device Profile, Personal Basis Profile
CLDC/ CDC
Operating System

Figure 2. A typical JSR 172 based application architecture

There are many WEB services useful for developing Java ME client applications. Among them, Google Maps API WEB services is a collection of HTTP interfaces to Google services providing geographic data for maps client applications. Google Maps offers REST services that allow accessing its data with simple HTTP requests, so they can be easily integrated into mobile applications [13].

The developer must sign up, [23] to get a key (API_KEY, a simple string) that he/she will use for all the queries to Google Map services.

The static maps WEB service allows retrieving single images that can be used in mobile applications, because it not requires Javascript or any dynamic page loading.

The static maps service supports different image formats (png32, GIF, JPG) and customizable image size that can be used for all purposes. The developer must retrieve an URL with an HTTP request, that must include some parameters like: the map's center geographical coordinates (latitude and longitude), the image format and size, the zoom level (the zoom range is from 0 to a maxim level of 19) and the API_KEY [14].

Geocoding is the process of converting an usual address into geographical coordinates, that can be used, for example, to place markers or position the map. The Google Geocoding API is another REST service that provides access to a geocoder via an HTTP request, so it can be integrated into mobile applications. Additionally, "reverse geocoding" can be performed, that means the converse operation (turning coordinates into addresses).

Location API (JSR - 179) allows Java ME applications to get the user geographic location. The implementation of this technique in the device can be based on a GPS technology, using the mobile phone network (with Cell ID), etc.

V. WEB SERVICES FRAMEWORKS AND PROTOCOLS

WEB services are being widely deployed to facilitate interoperability across different hardware and software implementation, machine architectures and application programming interfaces (API's) [9].

Creating effective mobile WEB services requires an architecture that addresses issues related to identity management, security, the machine readable description of WEB services and methods for discovering WEB services instances.

The XML Protocol work is the foundation for a WEB Service framework within which automated, decentralized services can be defined, deployed, manipulated and evolved in an automated fashion. This framework provides a structure for integration and a foundation for protocols that will support the needs of such service - oriented applications. The goal is a scalable, layered architecture, one that can appropriately meet the needs of both simple and extremely robust high - volume deployments. As with other Web technologies, the focus is on enabling ubiquitous interconnectivity of entities and organizations dispersed throughout the world. The WEB services framework focuses on supporting application - to - application integration between entities having disjoint platforms, management, infrastructures and trust domains. Using the framework, a model for describing, discovering and exchanging information can be realized, that is independent of application implementations and the platforms on which applications are developed and deployed.

The UbiPOL platform will be realized using frameworks and stacks that allow the WEB services implementation using Java language. As implementation options for UbiPOL platform the following WEB service frameworks have been considered: Axis 1.x, Axis2, CXF, Glue, JBossWS, XFire (1.2), Metro, OracleAS 10g.

Their general features and WS related JSR standards, as a criteria list approach, are presented synthetically in the following tables.

Apache Axis is an open source, XML based WEB service framework, It consists of a Java and a C++ implementation of the SOAP server, and various utilities and APIs.

Apache Axis 2 is a core engine for WEB services. It is a complete re - design and re - write of the widely used Apache Axis SOAP stack.

Apache CXF is an open source services framework, suitable for building and developing services using frontend programming APIs, like JAX - WS and JAX - RS. These services can comply with protocols such as SOAP, XML/HTTP, RESTful HTTP, or CORBA.

GLUE Java WEB services, delivered by Mind Electric, is a WEB service toolkit for Java programmers that offers an easy way to implement SOAP messaging.

For UbiPOL, **the Glue server** can have two parts: a Java class that performs the business logic and a Java class that starts GLUE's HTTP Server and publishes the business logic object as a SOAP service.

JBossWS is a WEB service framework developed as part of the JBoss Application Server. It implements the JAX - WS specification that defines a programming model and run - time architecture for implementing WEB services in Java [12].

XFire is an open source Java SOAP framework built on a high performance, streaming XML model. XFire includes support for WEB service standards, an easy to use API, Spring integration, JBI support, and plugable bindings.

The Metro WEB services stack is an open source tool developed by Sun Microsystems. It incorporates the reference implementations of the JAXB 2.x data - binding and JAX - WS 2.x WEB services standards, along with other XML - related Java standards.

Oracle Application Server 10g R3 WEB Services provides a new runtime infrastructure supporting J2EE 1.4 WEB services. Figure 3 provides an architectural overview of this new infrastructure [15].

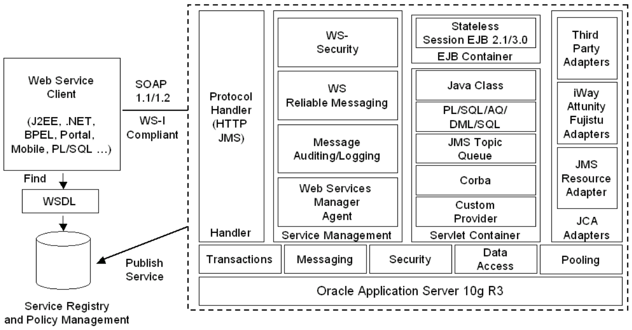


Figure 3. Oracle Application server 10g R3 (10.1.3.0.0) WEB Services Framework

Their general features and WS related JSR standards, as a criteria list approach, are presented synthetically in the following tables.

TABLE IV. WEB SERVICES FRAMEWORK GENERAL FEATURES

Feature	Axis 1.x, 2	Cxf	Glue	JBossWS XFire 1,2 Metro	Oracle AS 10g
Basic Profile Compliant	☑	☑	☑	☑	☑
Easily Create Services from POJOs	☑	☑	☑	☑	☑
Open Source	☑	☑	☑	☑	
RPC-Encoding	☑	☒	☑	☑ M1	☑
Spring Support	☑	☑	☒	☒	☒
REST Support	☒	☑	☑	☒	☑
IDEA Plugins	☒	☑	☑	☑	ANT
Eclipse Plugins	☒	☑	☒	☑	ANT
NetBeans Plugins	☒	☒	STP	☒	ANT
JDeveloper	☒	☒	☒	☒	☑
Hot Deployment	☒	☑	☒	☑	☑
Soap 1.1	☑	☑	☑	☑	☑
Soap 1.2	☑	☑	☑	☑	☑
Streaming XML (StAX based)	☒	☑	☑	☒	☒
WSDL 1.1 –Code (Client)	☑	☑	☑	☑	☑
WSDL 1.1 –Code (Server)	☑	☑	☑	☑	☑
WSDL 2.0 –Code (Client)	☒	☑	☑	☒	☒
WSDL2.0 – Code (Server)	☒	☑	☒	☒	☒
Client-side Asynchrony	☑	☑	☑	☒	BPEL
Server-side Asynchrony	☑	☑	☑	☒	BPEL
Policy-driven code generation	☒	☑	☑	☒	☒

TABLE V. WS RELATED JSR STANDRADS

Feature	Axis 1.x, 2	Cxf	Glue	JBossWS XFire 1,2 Metro	Oracle AS 10g
JAX-RPC	☑	☑	☑	☑	☑
JAX-WS	☑ A22	☑	☑	☑ M3	☑
JAX-RS	☑	☑	☑	☑	
JSR 181	☑	☒	☑	☑	☑
JSR 181 on Java 1.4	☑	☑	☒	☒	☒
SAAJ (1.2/1.3)	☒	☑	☑	☒	☑
JSR 109	☒	☑	☑	☑	ANT
JBI	☒	☑	☒	☑	ANT

Generally, all the previously WEB services Frameworks comply with the following transport protocols: HTTP, JMS, SOAP/JMS Spec, Jabber, SMTP/POP3, TCP.

The notes from Tables IV and V highlight some issues relevant to our objectives, as follows:

- M1 - in case of the Metro WEB service framework, the Remote Procedure Call encoding is available only through the JAX-RPC 1.1 APIs and removed JBI JSR;

- M3 - in case of Metro WS, JAX-RPC 1.1, JAX-WS 2.0 and JAX-WS 2.1 RI are combined together in Metro as well as JAXB 2.0 and JAXB 2.1. JAX-WS 2.0 and JAXB 2.0 functionality is available in Java SE 6 as well;

- A22 –in case of Axis 2 WS, there is not JAX-WS TCK compliant due to lack of JAX-WS tooling;

- BPEL – allow services interaction to be described easily and thoroughly by the WS – BPEL, an XML based programming language;

- ANT – is a Java based software tool useful for automating software build processes.

CXF supports a very wide range of connection possibilities, by using the Camel transport.

The specification from the Tables IV and V show that Metro is in conformance with the WS-I Basic Profile and it supports SOAP 1.1 messages relayed over HTTP as transport. It also provides support for WSDL 1.1 and XML.

Metro is an open source web service stack that is a part of the GlassFish project. It is included in Glassfish V3 and it is available under the CDDL and GPLv2 licence. Since Metro is compliant with the WS-I Basic Profile and the requirements of JSR 172, web services deployed with it can target Java Me applications running on mobile phones as service consumers. That is why Metro seems an appropriate web service stack option for UbiPOL.

OSGi technology is the dynamic module system for Java™. The OSGi Service Platform provides functionality to Java that makes Java the premier environment for software integration and thus for development [11].

ProSyst, an OSGi and Java pioneer, is a company entirely focused on open standards technology and was most actively involved in helping to create the OSGi specifications. It offers OSGi technology based products and services for Mobile Devices market [12].

VI. CONCLUSIONS AND FUTURE WORK

The following general advantages indicate WEB services as a suitable solution for UbiPOL:

WEB services are platform and language independent: WEB services provide interoperability between various software applications running on disparate platforms. They are not tide to any operating system, development platform or programming language. WEB services allow different applications from different sources to communicate with each other, without time - consuming or custom coding. This is because WEB services use open standards and protocols like SOAP, WDSL and XML. After deployment, this could provide other developers with the opportunity to include UbiPOL WEB services in their custom applications and deliver UbiPOL content and data merged with their own.

One solution for handling interaction between Java ME applications running on mobile phones and UbiPOL servers are WEB services. WEB services will be the logic tier components of the scalable UbiPOL system architecture.

There are several APIs that provide WEB service client support for the Java Me platform: WSA, WINGFOOT SOAP client, kSOAP2. The WSA specification was developed within the Java Community Process as JSR 172. The J2ME WEB Services API (WSA) extends the Java 2 Platform, Micro Edition to support WEB services. This is why it is included in many Java ME platform implementations by mobile phone manufacturers. WSA provides out of the box support for Java ME WEB service client applications on many mobile phones available on the market.

In order to develop Java ME WEB service client applications, UbiPOL must rely on an available API. Because WSA (JSR 172) provides out of the box support for Java ME WEB service client applications on mobile phones, it is the most appropriate option for now. Other third party APIs require including those APIs in the distribution file. This would lead to larger distribution files, which is not recommended. The other options may be implemented if they prove to be necessary.

WSA (JSR 172) conforms to the WS - I Basic Profile WEB service specification. It requires the use of SOAP 1.1, WDSL 1.1 and XML 1.0. The WEB service framework or stack that will be used for the deployment of UbiPOL WEB services on the server must comply with the same specification as WSA in order to ensure compatibility and interoperability.

Several WEB service frameworks were analysed: Axis 1.x, Axis2, CXF, Glue, JBossWS, XFire (1.2), Metro, OracleAS 10g.

The application server identified as required for the implementation of UbiPOL is Glassfish. Metro is the WEB service application stack bundled with Glassfish. It is in conformance with the WS - I Basic Profile WEB service specification. This makes Metro the best available option for now.

WEB services will be logic tire components of the UbiPOL system architecture, enabling the interaction between UbiPOL Java ME applications (presentation tier) and UbiPOL database servers (data tier) [17]. UbiPOL WEB services will be in conformance with the WS - I Basic Profile 1.0 specification. SOAP 1.1 conveyed over HTTP will be used for messages sent between Java ME applications and WEB services. As SOAP is an XML based protocol, the data conveyed between Java ME applications and UbiPOL WEB services will be XML serialised for both request and response messages. UbiPOL WEB services will make use of WDSL 1.1 to provide a “machine - processable” description of the operations supported, enabling client side code generation in many IDEs. The Metro WEB service stack and the Glassfish application server will be used to deploy UbiPOL WEB services. The WSA (JSR 172) API will be used in the implementation of UbiPOL Java ME WEB service client applications.

ACKNOWLEDGMENT

This paper represents part of the work realized by the authors within the UbiPOL FP7 European Union research project, dealing with e - participation of citizens in policy making and funded by grant agreement nr. 248010.

REFERENCES

[1] A. Pashtan, “Mobile WEB Services” Cambridge University Press 2005 pp. 5 – 30.

[2] M. Papazoglou, "What Are WEB Services?" WEB Services: Principles and Technology. Harlow, England Pearson/Prentice Hall, 2008. pp. 22- 32.

[3] Gartner Press Release on Worldwide Mobile Phones Sales http://www.gartner.com/it/page.jsp?id=1126812, [accessed: 30.08.2010] 20 August 2010

[4] P. M. Woo, K.Y. Seok, and Kyong - Ho “Migrating WEB Services in Mobile and Wireless Environments” International Journal of WEB Services Research, Volume 6, Number 2, April - June 2009 (pp. 1 - 19).

[5] D. Lizcano, J. Soriano, M. Reyes and J. Hierro “A user - centric approach for developing and deploying service front - ends in the future internet of services” International Journal of WEB and Grid Services (IJWGS) Vol 5, Issue 2 – 2009, pag. 155 - 191, doi: 10.1504/ IJWGS. 2009.027572

[6] A. Yamazaki, A. Koyama, J. Arai and L. Barolli “Design and implementation of a ubiquitous health monitoring system” International Journal of WEB and Grid Services (IJWGS) Volume 5, Issue 4 – 2009, pag. 339 - 355, doi: 10.1504/IJWGS.2009.030263

[7] E. Cerami, "Introduction to WEB Services."WEB Services Essentials, O'Reilly & Assoc., USA 2002, ISBN 0 – 596 - 0 224 - 6, pp. 10 - 50.

[8] G. Alonso and F. Casati “WEB Services - Concepts, Architectures and Applications”, Springer – Verlag – Berlin, Heidelberg 2004 ISBN 3 – 540 - 44008 - 9, pp. 124 - 149.

[9] L Richardson and S Ruby “RESTful WEB Services” Farnham O'Reilly 2007 pp. 47 - 67

[10] “Appache WS Wiki”; http://wiki.apache.org/ws StackComparison; [accessed: 30.08.2010];

[11] “OSGi Technology”; http://www.osgi.org/About/Technology; [accessed: 30.08.2010];

[12] “ProSyst is an OSGi and Java Pioneer”; http://www.prosyst.co; [accessed: 30.08.2010];

[13] “GoogleMap API WEB Services” http://code.google.com/apis/ maps/documentation/webservices/index.html; [accessed: 30.08.2010];

[14] “How to use Google Map Data in Mobile Applications” http://wiki.forum.nokia.com/index.php/How_to_use_Google_Maps_data_in _mobile_application; [accessed: 30.08.2010];

[15] “Java ME SDK 3.0” http://www.oracle.com/technetwork/java/javame/ downloads/ sdk30-jsp-139759.html; [accessed: 30.08.2010];

[16] “Java ME SDK 3.0 released” http://weblogs.java.net/blog/ 2009/04/22 /java-me-platform-sdk-30-released-goodbye-wtk-hello-java-me-sdk-part-2; [accessed: 30.08.2010];

[17] “Nokia S60 3rd Edition SDK for Java” http://www.forum.nokia.com/ info/sw.nokia.com/id/6e772b17-604b-4081-999c-31f1f0dc2dbb/S60; [accessed: 30.08.2010];

[18] “LG SDK 1.5 for the Java ME Platform” http://developer lgmobile.com/lge.mdn.tnd.RetrieveTNDInfo.dev?modType=T&objectType =T&menuClassCode=&saveFileName=&resourceNo=TND00000294&selectedType=&tabIndex=1#none; [accessed: 30.08.2010];

[19] ”]MOTODEV Studio for Java ME: Downloads” http://developer.motorola.com/docstools/motodevstudio/javame/download [accessed: 30.08.2010];

[20] “New Samsung Java SDK 1.1.2 - release 17th Nov 2009” http://innovator.samsungmobile.com/down/cnts/toolSDK.detail.view.do?platformId=3&cntsId=5640&listReturnUrl=http://innovator.samsungmobile.com:80/down/cnts/toolSDK.list.do%3FplatformId%3D3; [accessed 30.08.2010];

[21] ”What's new in BlackBerry Java application development 5.0” http://www.blackberry.com/developers/docs/5.0.0api/index.html; [accessed: 30.08.2010];

[22] ” Sony Ericsson SDK for the Java ME platform 2.5.0.6” http://dwgs3.sonyericsson.com/wportal/devworld/downloads/download/dw-99962-semcjavamecldcsdk2506?cc=gb&lc=en; [accessed: 30.08.2010];

[23] ”Sign up for the Google Maps Api” http://code.google.com/apis/maps /signup.html; [accessed: 30.08.2010];

[24] ” Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2009 - 2014” http://www.cisco.com/en/US/solutions/collateral /ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html; [accessed: 30.08.2010];

Development of Web 2.0 Applications using WebComposition/Data Grid Service

Olexiy Chudnovskyy, Martin Gaedke
Faculty of Computer Science
Chemnitz University of Technology
Chemnitz, Germany
olexiy.chudnovskyy@s2004.tu-chemnitz.de
martin.gaedke@informatik.tu-chemnitz.de

***Abstract*—Data integration and content publishing in terms of Linked Data is a complex and time-consuming task while developing Web 2.0 applications. Considering this problem separately from architecture design increases application maintenance effort and causes additional overhead to provide public access functions. In this paper, we present the WebComposition/Data Grid Service and its data management capabilities to meet demands of modern Web 2.0 applications. We show how to facilitate the application implementation and shorten development time by applying the Data Grid Service as Web Service-based storage solution.**

***Keywords*-REST; Linked Data; Web 2.0.**

I. Introduction

The classical approach while developing Web 2.0 applications foresees many steps beginning with problem analysis over data modeling, architecture design and ending with implementation and maintenance [1]. Consider the development process of a small Web 2.0 application. As an example, we create a small online-tool to support Scrum software development method [2]. In Scrum a product owner separates the project into stories, which are functionalities a client wishes from the application. Stories are implemented by the Scrum team during sprints - fixed periods of time, usually 2 or 4 weeks. The team divides the stories into small tasks and solves them by implementing the specified functionality. If problems occur, so called impediment requests are posted to the scrum master, who tries to solve them and cares about the smooth development process.

First we define entities and relationships using the UML class diagram from Fig. 1. Following the classical approach we use one of the Object-Relational-Mapping libraries (like Hibernate [3] or Microsoft Entity Framework [4]) to map the described classes and associations onto tables of a relational database. This way the application deal only with conceptual scheme and concentrates on business logic, abstracting from database read/write operations and communication details. With the help of a Model-View-Controller (MVC) Framework we develop the presentation level and implement navigation functions. Due to the simplicity of our example application it doesn't take much time to implement the business logic. The creation, edition and retrieval functions are usually automatically generated by modern MVC frameworks.

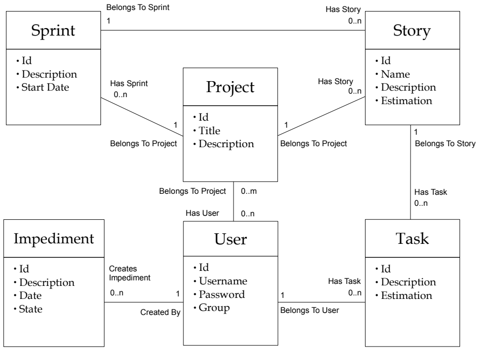


Figure 1. UML class diagram of scrum tool.

The created application works fine as a standalone tool, but still doesn't collaborate with other services or exchange any data. Assuming we would like to provide a Really Simple Syndication (RSS) feed [5] with newly added impediment requests, additional programming effort in implementation of a web-service is needed making this simple feature costs- and time-consuming. Publishing the information about ongoing projects in Resource Description Framework (RDF) format [6] requires new work again, transforming internal data into new representation and exposing it by implementing new web service methods. Further functionalities like public or new data representations become even more expensive due to the implementation and maintenance costs.

As we see, application development time and project costs could be decreased if collaboration with other services as well as publishing of content in terms of Linked Data would be considered in the planning phase. Addressing these problems after essential application functions are implemented, data is strictly modeled and manipulation methods are defined makes the further development inefficient and increases maintenance costs. Our solution acts as a web-based storage solution targeting common integration and data exchange needs of modern Web 2.0 applications and supporting the developer in implementation and maintenance of web-based applications. We show how schema-free data can be modeled using Data Grid Service (DGS) and manipulated in the RESTful way [7] (Section II). We illustrate the application of Data Grid Service as underlying storage engine (Section III) and discuss it respecting complexity and performance aspects (Section IV). We also present some related approaches in Section V.

II. WebComposition/Data Grid Service

In this section, we discuss the fundamentals of the WebComposition/Data Grid Service, present data modeling possibilities, access methods and internal architecture of the service.

A. Basic Principles

The Data Grid Service acts as a flexible and easy to integrate component providing wide information exchange and sharing possibilities. Focusing on the management of XML lists and corresponding metadata in a RESTful way, the service can be applied in a variety of scenarios with different requirements on discovery, presentation and integration of data. The concept of URI plays a decisive role in data access and manipulation methods, the variety of supported representation formats makes it easy to share the information and integrate it into existing applications. Though the service focuses on the maintenance of data in form of XML lists, further functionality such as content transformation using XSLT stylesheets, binary content or gateways to other data sources may be managed through extensions.

The logical view on the resources managed by Data Grid Service is described by a set of so called information stores (Figure 2). The information stores provide access to the resources inside and corresponding metadata. For example an information store may act as a single XML list, containing XML representation of people or publications and providing Create/Read/Update/Delete (CRUD) methods for item manipulation. The information stores, metadata and single items are references through URIs, service architecture allows items to contain further information stores or act as a gateway to other services or data sources. To create information stores within the Data Grid Service a corresponding HTTP request is made with descriptive information about the newly created information store. Configuring the stores using metadata allows not only the definition of functionality but also affects performance issues, e.g., XML lists may be internally stored either directly in separate files or in the database to maintain larger amounts of data. Moreover relationships between information stores can be configured to merge the contents and process the combined data.

B. Data Model

The WebComposition/Data Grid Service manages structured data in form of XML lists. Additionally service can handle lists of binary arrays, providing a fast and flexible storage solution for web application resources. Resource metadata in form of RDF statements can be created to annotate the stored information, connect it with related resources or configure access and manipulation methods. XSLT stylesheets are used to transform the content into another representation formats, such as RDF or JSON, or to organize the data as Atom or RSS Feeds.

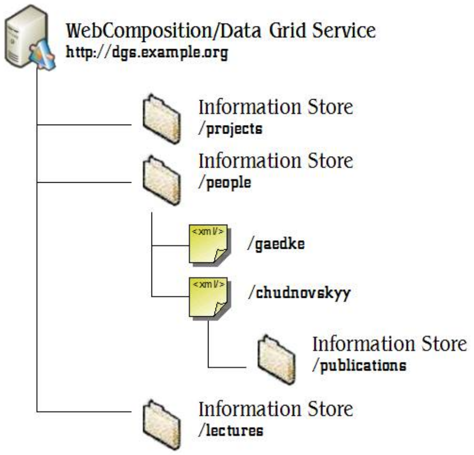


Figure 2. WebComposition/Data Grid Service. Logical view.

Resources in DGS may be created via HTTP in the RESTful way and configured for the future use by provided descriptive metadata. Depending on the type of the created resource its behavior and access methods are defined. In case of XML lists, single XML blocks may be added, retrieved or deleted. To validate the incoming data a XSD schema can be defined, specifying either the overall list structure or making restrictions on the incoming elements. The approach simplifies the maintenance of objects used by web applications and allows making changes into data structure without reorganizing dependencies or affecting other contents. In order to support the development of applications with focus on relationships between objects foreign keys and connections may be specified between lists providing a flexible access to subordinate items. The relationships are defined using RDF statements so that external services may consume this information to optimize their own discovery and integration functions.

C. Data Manipulation

The WebComposition/Data Grid Service is a web component designed in a RESTful architectural style, providing a number of resource discovery and maintenance functions. With a HTTP GET request on service metadata (/meta) the RDF description of existing information stores and current service configuration is retrieved

New information stores are added with a POST request on Data Grid Service URL providing description of the resource to be created. The type of the resource (list of XML elements, XSLT transformation, gateway to other services etc.) defines the allowed operations both on the resource itself but also on the subordinate items. Single list items are created with a POST request on the corresponding parent list URL:

```
POST /authors HTTP/1.1
Host: dgs.example.org
Content-Length: 89
Content-Type: text/xml

<author>
  <fname>Olexiy</fname>
  <sname>Chudnovskyy</sname>
  <city>Chemnitz</city>
</author>

HTTP/1.x 201 Created
Location: http://dgs.example.org/authors/5
```

Unique id's are assigned to the newly created store items so these can be later retrieved, updated or deleted with corresponding GET, PUT or DELETE methods. To make the item URI even more descriptive a URI Template [8] may be defined to map the incoming request onto predefined XPath expression selecting the appropriate items from the list:

```
POST /authors/meta HTTP/1.1
Host: dgs.example.org
Content-Length: 89
Content-Type: text/n3

@prefix meta: <http://www.webcomposition.net
     /2008/02/dgs/meta/>.
<http://dgs.example.org/authors>
meta:urlTemplate
[
 meta:url "authors/{value}";
 meta:xPath "/authors/student[sname='{value}']"
].
```

The newly created item would be then alternatively available under the URI: *http://dgs.example.org/authors/chudnovskyy*.

The metadata of XML list items may provide further information in RDF format about the resource e.g., its creation date or list creator.

The described approach simplifies the fast development of many Web 2.0 applications, e.g., blogs, online presentations or information sharing portals by providing a flexible and intelligent storage solution. Satisfying the needs of developers to model structured data, Data Grid Service exposes its content in a RESTful way, so the content may be immediately consumed by other applications and services.

To support applications based on the domains with many connections between items, the pre-configured relationships are used by the Data Grid Service to aggregate subordinate items. The relationship is defined through 4 obligatory and 3 optional attributes:

- *Source*: A URI of the information store within the Data Grid Service to act as a primary list, e.g., *http://dgs.example.org/authors/*
- *Target*: A URI of the information store within the Data Grid Service to act as a subordinate list, e.g., *http://dgs.example.org/publications*
- *Predicate*: A URI of RDF predicate to act as a foreign key, defining a connection between primary and secondary list items, e.g., *http://www.webcomposition.net/2008/02/dgs/meta/hasPublished*. Predicates are automatically stored in the metadata of the parent item.
- *URI*: The unique identifier for the relationship. In particular an URL within Data Grid Service domain is used to retrieve the relationship details, to modify or to delete it. The URL is provided by the service and is sent in the Location header to the client after creation.

A relationship defined through the obligatory attributes allows the service to process URIs after the following pattern:

```
http://{service_host}/{source_list_name}/
{source_item_id}/{target_list_name}
```

and as such, filtering only those items from target list that have a relationship to the parent list item *source_item_id* over the RDF property defined in *Predicate*-attribute. A POST request on the same URI is used to add new items to the subordinate list connecting it simultaneously with the given parent list item. An inverse operation to remove the relationship between items is performed using a DELETE request on the URI

```
http://{service_host}/{source_list_name}/
{source_item_id}/{target_list_name}/
{target_item_id}
```

An optional *Inverse-Predicate*-attribute can be specified to define a reverse relationship from the target list to the source list. A corresponding RDF statement is then automatically assigned to the child item metadata, acting as a foreign key to the parent list item. The approach improves both performance processing *n:m* relationships and lets the Data Grid Service process the URIs after the reverse pattern:

```
http://{service_host}/{target_list_name}/
{target_item_id}/{source_list_name}
```

In example above both publication(s) of some fixed author and also author(s) of some fixed publication can be retrieved with simple GET requests on the corresponding URIs. If many relationships between the same source and target list should be modeled, optional *Source* and *Target Aliases* are specified to resolve conflicts with already existing relationship definitions. The predicate of the relationship is then used to perform aggregation of target list items and response to the requests URIs like

```
http://{service_host}/{source_list_name}/
{source_item_id}/{target_list_alias}
```

or

```
http://{service_host}/{target_list_name}/
{target_item_id}/{source_list_alias}
```

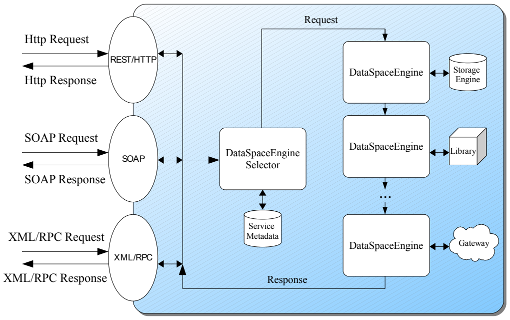
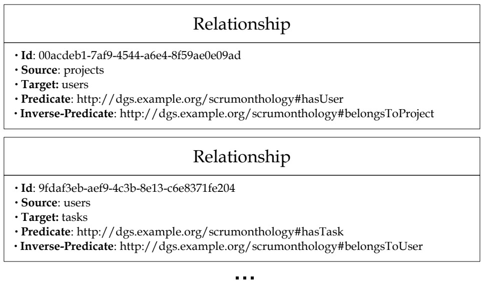


Figure 3. Data Grid Service architecture



Relationship

- **Id**: 00acdeb1-7af9-4544-a6e4-8f59ae0e09ad
- **Source**: projects
- **Target:** users
- **Predicate**: http://dgs.example.org/scrumonthology#hasUser
- **Inverse-Predicate**: http://dgs.example.org/scrumonthology#belongsToProject

Relationship

- **Id**: 9fdaf3eb-aef9-4c3b-8e13-c6e8371fe204
- **Source**: users
- **Target:** tasks
- **Predicate**: http://dgs.example.org/scrumonthology#hasTask
- **Inverse-Predicate**: http://dgs.example.org/scrumonthology#belongsToUser

...

Figure 4. Definitions of entity relationships.

The data modeling possibilities and processing functions let the developer concentrate on the contents and user interface of the web application supporting them by flexible and intelligent storage solution. The ease of service integration and data retrieval shortens both development of prototypes and real applications. As the content is immediately available in form of XML for consumption and integration into existing applications, implementation of additional web services or API to access the application data is unnecessary. The variety of representations formats and target clients is supported through configurable transformation of output content. E.g., author may provide an RSS Feed of his newly posted publications simply providing the Data Grid Service a XSLT stylesheet, that should be applied on the XML document returned to the request on *http://dgs.example.org/authors/chudnovskyy/publications*. The same way one creates RDF graphs or JSON representation from the data stored in Data Grid Service.

D. Data Grid Service Internals

The flexibility of the service is achieved by integrating new components, so called Data Space Engines, handling the incoming requests with predefined URI patterns (Figure 3). The request is first analyzed by the *DataSpaceEngine*-Selector component to determine the information store type of the requested URI.

The processing of the request is done afterwards by a chain of Data Space Engines, providing the specific behavior of the information store. The Data Space Engines may complete different tasks, such as authorization, resource versioning, data manipulation or gateway functionality. 3rd party libraries, components, storage engines and services may be used to accomplish the task. Following Data Space Engines are currently implemented:

- *XmlDataSpaceEngine* - main component providing the basic functionality on XML lists. Both lists and XML items are created using corresponding HTTP requests. Metadata is maintained for stored resources, containing RDF statements describing the contents and relationships to other lists or items. URI Templates and relationship definitions are resolved and processed by the component.
- *XSLTDataSpaceEngine* - transforms the requested XML resource according to the defined XSLT transformation. XML lists or single items can be used to create an alternative representation of contents, e.g., Atom or RSS feeds, RDF graphs, JSON representation or HTML pages. The behavior is configured through the list metadata.
- *BinaryDataSpaceEngine* - manages lists of binary content, automatically extracting meta data from known formats and storing it using common RDF vocabularies.

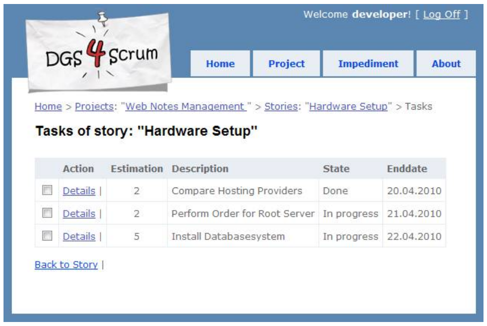
The well defined interface allows developers to extend the functionality of the service implementing further functionality, e.g., synchronizing the stored contents with other data sources or restricting access to specific resources.

III. WebComposition/Data Grid Service in Use

In this chapter we apply the Data Grid Service as a storage solution to the example application discussed in Section I. To represent the entities, we start with defining 6 XML lists representing the entities (classes) of the UML model, i.e., their URIs */projects*, */stories*, */sprints*, */users*, */tasks* and */impediments* and corresponding schemas to perform data validation. We also describe the associations by submitting the corresponding information to the Data Grid Service (Figure 4).

Each relationship is specified in the service's metadata through a set of RDF statements, identifying source and target XML lists as well as connection predicates. The associations between entity objects (depending on the direction) are now represented by URIs as follows:

- */projects/*{*project id*}*/users* &
  /users/{*user id*}*/projects*
- */projects/*{*project id*}*/stories* &
  /stories/{*story id*}*/projects*
- */users/*{*user id*}*/tasks* &
  /tasks/{*task id*}*/users*
- ...



Welcome **developer**! [Log Off]

Home > Projects: "Web Notes Management " > Stories: "Hardware Setup" > Tasks

Tasks of story: "Hardware Setup"

	Action	Estimation	Description	State	Enddate
□	Details |	2	Compare Hosting Providers	Done	20.04.2010
□	Details |	2	Perform Order for Root Server	In progress	21.04.2010
□	Details |	5	Install Databasesystem	In progress	22.04.2010

Back to Story |

Figure 5. Example application based on Data Grid Service.

The definitions made above are enough to maintain the application data, to create new entities and to connect them. The content manipulation is performed using GET, POST, PUT and DELETE methods. As in the approach from Section 1 we implement the UI layer and authorization logic using a MVC Framework (ASP.NET MVC [9]). In contrast, implicit support for data exchange and integration is given through the RESTful architecture of Data Grid Service. Transforming XML lists using XSLT allows other applications to consume the data in a suitable for them format. For example we created a simple XSLT transformation to provide a RSS feed with information about newly added impediment messages. The same we exposed the RDF representation about current projects, stories and tasks.

IV. Evaluation

A. Performance

To test the performance of the Data Grid Service regarding relationship processing functions we evaluated the service response time while retrieving subordinate XML list items within a simple relationship. The evaluation was performed by measuring the service response time on a local machine after request for all users within a specific project (as in the example application from the section I) using the URI *http://dgs.example.org/projects/5/users*. The number of items within users list varied from 100 to 5000, which corresponds to the objects count in the middle-size Web 2.0 application. The measures were performed on a service, hosted in ASP.NET Development Server on a PC with Intel Core2 Duo 2.66 GHz CPU 3 GB RAM and WD Velocity Raptor HDD (10000 rot/min). The results in Figure 6 show the linearity of response time function due to the straightforward implementation of XML filtering procedures using XPath. The current implementation gives acceptable results for smaller amounts of data, but should be revised for larger XML lists and faster response times. Currently we consider usage of RDF Tripple Stores, caching techniques and XML databases to meet the requirements of larger Web 2.0 applications and to optimize the overall service performance.

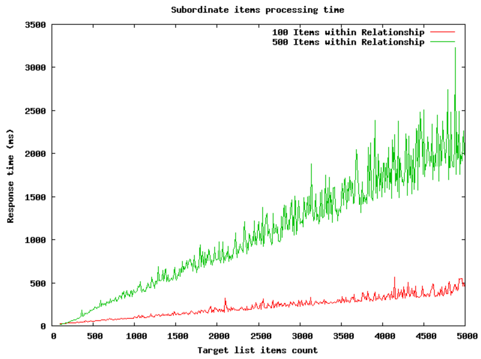


Figure 6. Measurements of service response time.

B. Development Process

We notice that the overall application complexity decreases due to the implicit resource oriented architecture and integration capabilities of Data Grid Service. The implementation of additional web services in order to expose application content or offer public data manipulation functions becomes redundant due to the RESTful architecture of Data Grid Service and its loose coupling with the application. The publishing of content in terms of Linked Data is implicitly supported through XSLT transformations and may be anytime reconfigured without changing the application code.

V. Related Work

During the last years a large number of distributed non-relational data storage solutions appeared. While meeting many web-specific requirements, the solutions concentrate on these main problems:

- High availability. An uninterrupted access to the stored data is especially important in Web 2.0 (business) applications to serve the customers around the world and to any time.
- Scalability. The time delay of read/write functions is stable even if maintenance routines are running or the number of clients emerges.
- Simple data modeling. The built-in support for key-value pairs or schema-free content simplifies the implementation of data driven Web 2.0 applications.

Following we discuss some interesting solutions providing the mentioned functionality and that are related to our approach.

- *CouchDB* - The Apache CouchDB Project [10] is a document-oriented storage solution accessible over HTTP in the RESTful style. The documents maintained by CouchDB are objects containing a variable number of named fields. The absence of document schemas makes the solution flexible for often data structure changes and new document types. Availability and robustness aspects are greatly solved, the manipulation of content requires solid JavaScript skills, the modeling of relationships between documents requires mixing objects and their metadata, the retrieval functions must be manually defined.

- *WCF Data Services* - The Microsoft WCF Data Services [11] is a part of .NET Framework, enabling developers to expose the data on the Web in a RESTful way. The contents are addressed through URIs and may be retrieved in different formats like JSON or XML. The data is described using Microsoft Entity Data Model based on the Entity-Relationship-Model. The service offers an easy traversal of collections, items and relationships, the underlying storage may be either a relational database, such as Microsoft SQL Server or any other data source accessed with custom implementations of data source provider component.
- *Amazon S3* - Amazon Simple Storage Service (S3) [12] provides essential functionality to maintain data over the Web being accessed both through SOAP and over HTTP in the RESTful style. While Amazon S3 is used to store unstructured data, it is often accompanied by Amazon Simple DB [13] offering a storage solution to access structured information and objects metadata. The data stored in Simple DB is schema-free and is automatically indexed to optimize query operations.
- *Google Data API* - The Google Data API provides access to the data stored in Google products such as Spreadsheets or Calendar using Google Data Protocol [14]. Besides client libraries are available for many programming languages, abstracting the conceptual schema from the data serialization formats used for data transport. The second version of protocol used is fully compliant with AtomPub RFC 5023 [15]. The data access and management functions are fast and flexible, additional methods to retrieve or update partial entities are implemented.

The presented approaches concentrate mainly on scalability and performance issues, but do not provide built-in functions to annotation the content with metadata as well as to transform it into alternative representation formats, e.g., RDF/XML. However these issues are essential for data exchange and integration in modern Web 2.0 applications. In contrast, Data Grid Service simplifies the development of Web 2.0 applications by providing an implicit support for the mentioned functionalities.

VI. Conclusion and Outlook

In this paper, we presented the WebComposition/Data Grid Service as a web-based storage solution to meet the needs of modern Web 2.0 applications and to support the developer in the implementation and maintenance process. Particularly Data Grid Service facilitates the Web 2.0 application development by

- Fast and flexible data management methods
- Content manipulation in RESTful way
- Data annotations in RDF format
- Rich data modeling capabilities and schema-free data structures
- Support for different data representation formats (XML, JSON, RDF, N3)
- URI templates and associations handling for easier data manipulation
- Flexible architecture to extend the functionality

Existing applications take advantage of the built-in support for resource annotation and sharing capabilities of the service. In combination with further WebComposition components like Data Grid Service List Manager [16] (DGSLM) user input elements, e.g., XHTML forms can be automatically generated to manipulate the contents directly from the external web application. To secure single lists or items within the Data Grid Service WebComposition/ Identity Federation System (idFS) [17] may be used as identity provider component. Both DGSLM and idFS have been successfully tested and are used as embedded modules on the web page of Distributed and Self-organizing Computer Systems research group to publish the information about publications, projects and lectures. In order to improve service performance and shorten response times we are currently working on indexing the contents and caching techniques to avoid redundant parse procedures and accelerate the content delivery process. To support the collaboration between single service instances publish/subscribe mechanism is being developed in current research projects. We are also working to provide iteration and pagination functions to simplify content discovery and navigation process.

References

[1] T. O'Reilly. (2005, September) What is web 2.0? design patterns and business models for the next generation of software. http://oreilly.com/web2/archive/what-is-web-20.html. Last Access: 04.07.2010.

[2] K. Schwaber and M. Beedle, *Agile Software Development with SCRUM*. Prentice Hall, February 2002.

[3] R. F. Beeger, A. Haase, S. Roock, and S. Sanitz, *Hibernate: Persistenz in Java-Systemen mit Hibernate und der Java Persistence API*, 2nd ed. Heidelberg: dpunkt, 2007.

[4] A. Adya, J. A. Blakeley, S. Melnik, and S. Muralidhar, "Anatomy of the ado.net entity framework," in *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2007, pp. 877–888.

[5] *RSS 2.0 Specification*, RSS Advisory Board, http://www.rssboard.org/rss-specification. Last Access: 04.07.2010.

[6] G. Klyne and J. J. Carroll, “Resource description framework (rdf): Concepts and abstract syntax,” World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004. [Online]. Available: http://www.w3.org/TR/rdf-concepts/

[7] L. Richardson and S. Ruby, *RESTful Web Services*. O’Reilly Media, Inc., May 2007.

[8] J. Gregorio and M. Handley, *URI Template*, http://tools.ietf.org/id/draft-gregorio-uritemplate-03.txt. Last Access: 04.07.2010.

[9] J. Galloway, S. Hanselman, P. Haack, S. Guthrie, and R. Conery, *Professional ASP.NET MVC 2*. Wiley Publishing, Inc, June 2010.

[10] J. C. Anderson, J. Lehnardt, and N. Slater, *CouchDB: The Definitive Guide Time to Relax*. O’Reilly Media, Inc., 2010.

[11] M. Corporation, *WCF Data Services*, http://msdn.microsoft.com/en-us/data/bb931106.aspx. Last Access: 04.07.2010.

[12] Amazon, “Amazon s3 developer guide,” Amazon, Tech. Rep., 2010. [Online]. Available: http://aws.amazon.com/documentation/s3/

[13] D. Robinson, *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, SimpleDB and SQS Web Services enables you to reach business goals faster*. London, UK, UK: Emereo Pty Ltd, 2008.

[14] Google, *Google Data API*, http://code.google.com/intl/de-DE/apis/gdata/. Last Access: 04.07.2010.

[15] J. Gregoric and B. de hOra, *RFC 5023 - The Atom Publishing Protocol*, http://tools.ietf.org/html/rfc5023. Last Access: 04.07.2010.

[16] R. Sommermeier, A. Heil, and M. Gaedke, “Lightweight data integration using the webcomposition data grid service,” in *First International Workshop on Lightweight Integration on the Web (Composable Web’09) in conjunction with the 9th International Conference on Web Engineering (ICWE 2009)*, San Sebastian, Spain, 22.-26. Jun 2009, pp. 30–38.

[17] M. Gaedke, J. Meinecke, and M. Nussbaumer, “A modeling approach to federated identity and access management,” in *WWW ’05: Special interest tracks and posters of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 1156–1157.

Business Protocol Monitoring

Samir Sebahi, Mohand-Said Hacid
Université de Lyon
Université Claude Bernard Lyon 1
LIRIS CNRS UMR 5205
France
{samir.sebahi | mohand-said.hacid}@liris.cnrs.fr

Abstract—**Because it is never sure that a business process successfully tested or statistically checked will have the expected behaviour during its execution, it is necessary to bring verification to the execution phase, by continuously observing and checking the correct behaviour of business processes during run-time. In this paper, we propose a new monitoring framework to monitor business protocols. We provide a monitoring language called BPath, which is an XPath-based language for both expressing and checking temporal and hybrid logical properties at run-time, making visibility on business process external behaviour by expressing and evaluating statistical queries over execution traces.**

Keywords-monitoring; business process; business protocol; XPath; hybrid logic

I. Introduction

The advent of web services and Service Oriented Architecture (SOA) has made a considerable progress in the way applications are developed and used, leading to the opening of new borders for information systems, with more automation of tasks, complex and multiple interconnection scenarios between applications within the same system and across different systems. In this context, the task of checking correctness of business processes at run-time becomes particularly challenging.

Currently, the common practice for developing service-based systems is to employ the SOA paradigm [1], which enables composition of services into business processes in a particular order and according to a set of rules to provide support for business processes.

Two features characterizing SOA have retained our attention and guided our investigation towards building an approach for monitoring business processes: SOA uses a message-based communication model, and most of specifications and languages used in SOA are XML based.

Based on these considerations, we designed and developed a new monitoring framework based on message abstraction. This abstraction is called business protocol [2]. We provide an extension of XPath [3] to accommodate verification issues. The resulting language (called BPath) is also a query language that can be used to track and make visibility on business process execution.

The paper is organized as follows: In Section II, we present some related works. Section III presents the concept of business protocol, and presents a monitoring scenario. Section IV describes architectural and design principles of our approach for monitoring. In Section V, we present our monitoring language. Then, we show its applicability to monitoring in Section VI. Finally, we conclude in Section VII by summarizing our work and identifying some extensions.

II. Related Work

A lot of research works have been proposed in the last years to monitor business processes. Some of them are directly related to our work. Baresi and Guinea [7] proposed a language (WSCoL) for specifying constraints on execution by defining a set of monitoring rules for both functional and non-functional constraints with the capability of setting the degree of monitoring at run-time such as: validity time frame, priority and a set of certified providers, for which monitoring may be omitted. Also, it enables specifying expressions over process variables and supports a set of built-in functions, logical and mathematical operators, and quantification. This work was extended in [8] by providing a support for specification and checking of temporal properties at run-time like with our monitoring framework. In [9], both business process behaviors and monitoring properties were stated as event calculus predicates, which is a logic-based formalism representing actions and their effects. Then, monitoring properties are checked in a post-mortem way against the stated behaviors and the recorded behavior in execution log at runtime, making the monitoring framework non intrusive regarding the execution of the business process, which is also the same case in our monitoring framework. The authors in [10][11] proposed monitoring languages that are built on top of XPath. [10] Proposed an approach to the monitoring of business processes specified in BPEL. A visual language, called Business Process Query Language (BPQL), with query capabilities, over BPEL processes, was introduced. XQuery expressions are generated, in the same way that graphical notations help business process designers generate specification code, using dedicated icons for each activity. Hallé and Villemaire [11] proposed an approach for monitoring web services choreography by means of XQuery [12] engine. Linear temporal logic properties are translated into an equivalent XQuery expression. Then, it is evaluated over XML message traces representing the choreography. Our monitoring framework is distinguished by using a simple messages based abstraction, and an expressive hybrid logic based language.

III. BUSINESS PROTOCOL

The purpose of a business protocol is essentially to specify the set of conversations (sequence of messages) that are supported by a business process [2]. Formally, we define a business protocol as a tuple $P = (S, s_0, F, M, T)$ where:

- S is a finite set of states the process goes through during its execution.
- s_0 is the initial state.
- F represents the finite set of final states.
- M is a set of messages.
- $T \subseteq S{\times}S{\times}M$ is the set of transitions, where every transition is labelled with a message name and its polarity, when a message is consumed by the protocol, the transition is assigned the polarity sign(+), and when it is produced by the protocol, the transition is assigned the sign (**-**).

In order to give an intuitive idea about our monitoring approach, let us consider the following scenario, inspired from [9], of an online Car Rental System (CRS) shown Figure 1.

CRS offers a car location service: whenever a rent car request is received *(RentCar)*, the availability of the requested car will be checked. If it is not available, then a list of cars will be sent to the client, otherwise, the requested car is reserved, and a confirmation message is sent to the client *(CarReservation)*. Then, the client will send her/his bank information (*BankInfo*), which will be validated, before sending the *keys*. After returning the keys, the client receives a payment confirmation *(BankConfirmation)*. But, in case the bank information is not valid, *CardRejected* message will be sent to the client and the process instance is completed.

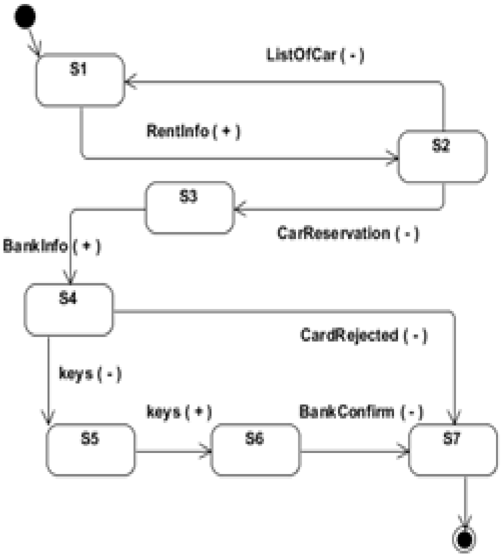


Figure 1. CRS business protocol

To show how our monitoring framework is able to monitor different kinds of properties and queries, we propose to consider the following list which should be continuously evaluated at run-time:

- **P1:** if a client's bank information is rejected, he should not get a car reservation before one hour.
- **P2: a** client should not get a car reservation when the keys are taken by another client.
- **Q1:** calculates the average time to perform a car reservation.
- **Q2:** counts the number of rejected bank information.

IV. THE OVERALL ARCHITECTURE

Figure 2 depicts the main components of the monitoring framework. First, a BPEL business process external behaviour is represented by means of a business protocol. Then, monitoring properties and queries are formulated using BPath monitoring language (presented in Section V).

At run-time, all incoming or outgoing messages will be captured by the business protocol monitor component before reaching their original destination. The process engine as well as the monitoring framework will publish the execution and monitoring events respectively, which will be stored in the execution log.

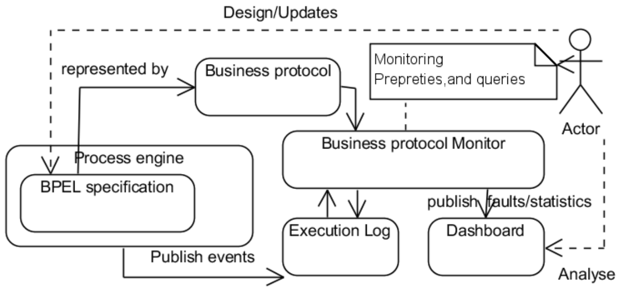


Figure 2. Monitoring framework

The execution log is of two types: state log, generated by the business protocol monitor, and event log generated by the process engine. On the basis of these generated execution logs, a checker component will check the correctness of the current execution and a Business activity monitor component will evaluate the specified statistical query to return statistical indicators on the execution of the process, and then both of these monitoring results will be published on a dashboard.

Additionally, the monitoring framework provides a set of business protocol execution events (see TABLE I.), to capture and control the exchanged messages, but also to specify when verification tasks should be performed.

For instance, to perform verification every time a message is received, we write:

```
OnMessageReceived (EventArgs  e){
    Check a property( $P_i$)
}
```

Or after a message is sent, as follows:

```
 OnMessageSent(EventArgs e){
    Check a property( $P_j$)
}
```

In the first case, a business process will be blocked until the verification is done. But, in the second case, verification task will not block the execution of the business process.

TABLE I. BUSINESS PROTOCOL EVENTS

Protocol Events	Description
OnEvent	Fires every time an event listed in this table occur.
OnNewInstance	Occurs when a new instance is started
OnNewState	Occurs when a state is entered
OnMessage	Occurs when a message sent or received
OnMessageReceived	Occurs when a message is received
OnAnknwonMessage	Occurs when a received message is not defined in the protocol
OnUnexpectedMessage	Occurs when a received message is defined in the protocol, but not expected from the current state
OnMessageSent	Occurs when a message is sent
OnTransition	Occurs when a transition from a state to another state happen
OnEndInstance	Occurs when an instance is ended

V. MONITORING LANGUAGE

In what follows, we consider that an execution of a business process as a sequence of states, independently of the fact that a business process can have different process instances, or parallel activities inside the same process instance.

The main idea behind our monitoring language (BPath) is first to consider a sequence of states representing the execution of a system as a special kind of tree. Each node represents a possible state, and its child node represents the direct next state. Then try to reuse the widely used language in the area of service based systems, which is XPath, as both a verification language and a query language.

So, BPath is built on top of XPath, and evaluated over a special tree of nodes (each node has only one child node, and no sibling nodes) forming a linear structure. BPath accommodates the notion of static and dynamic attributes and allows variable assignment inside path expressions. BPath offers a mean to express properties in first order hybrid logic. First order Hybrid logic [6] is an extension of first-order modal logic that makes it possible to name states and to assert that a formula is true at a named state.

A. BPath Syntax

A BPath formula is built according to the following abstract syntax:

$\varphi ::= T \mid T = T \mid P(T, \ldots, T) \mid \text{not } \varphi \mid \varphi \text{ and } \varphi \mid \varphi \text{ or } \varphi \mid (\varphi) \mid @_s\varphi \mid \downarrow s,\varphi \mid \downarrow_\pi s,\varphi \mid \downarrow_T x,\varphi \mid \exists x\ \varphi \mid \forall x\ \varphi$

$T ::= \pi \mid x \mid c \mid f(t_1,\ldots,t_n) \mid \pi/@q \mid s \mid s/@q$

$\pi ::= \text{Axis}::N \mid (\pi) \mid \pi[\varphi] \mid \pi / \pi \mid$

$N ::= n \mid *$

Axis ::= child | descendant | self | descendant-or-self| parent | ancestor| ancestor-or-self

Where: $x \in$ FVAR (a set of first-order variables), $n \in$LAB (a set of first-order constants), We define a function *lablel*: W→LAB, such that for each element of *W* associates an element of LAB, $s \in$ SVAR (a set of state variables), $q \in$ATTS (a set of unary function symbols, called static attributes) ∪ ATTD (a set of unary function symbols, called dynamic attributes) ∪ FUN (a set of one or more arity functions).

To simplify some expressions, we consider that *"π/child::N"* can be written as "π/N", that *"self::*/@q"* can be simply written as *"@q"*, and that "not (φ) ∨ α" can be abbreviated as "φ→ α".

B. BPath semantic

BPath formulas are interpreted in first-order modal models $M\ (W, R, D, I_w)_{w \in W}$ with constant domains such that: *W* is a set of nodes (or states){w_1, w_2...}, R is a linear modal relation on *W*. *D* is the interpretation domain. *(W, R)* is the modal frame.

For every $w \in W$, (D, I_w) is an ordinary first-order model such that:

- $I_w(c) = I_{w'}(c)$, for all w, w' ∈ W, c∈ CON.
- $I_w(q) \in D$, for q ∈ ATTS ∪ ATTD ∪ FUN.
- $I_w(P) \subset D^k$, for P a k-ary predicate symbol.
- $I_w(\pi) \subset W$ for π a path expression.

To interpret formulas with free variables, we define an assignment function *g* such that:

g: SVAR × FVAR → W × D

$g(x) \in D$ if $x \in$FVAR or $g(x) \in W$ if $x \in$ SVAR.

Given a model and an assignment *g*, the interpretation of the term *t*, denoted by $\overline{t}$ is defined as:

- $\overline{x} = g(x)$ for $x \in$ FVAR
- $\overline{s} = g(s)$ for $s \in$ SVAR
- $\overline{c} = I_w(c)$ for $c \in$ CON, for some $w \in$ W
- $\overline{s/@q} = I_{g(s)}(q)$ for *s* a state variable, and $q \in$ ATTS ∪ ATTD ∪ FUN.
- $\overline{\pi/@q} = \{I_{w1}(q)\ ,\ I_{w1}(q)\ ,\ldots\ I_{wn}(q)\ \}$such that $w_1, w_2 \ldots w_n \in I_w(\pi)$, for some $w \in$W.

We also define an assignment $g^{x1..xn}_{d1\ldots dn}$ such that:

$$g^{x1..xn}_{d1\ldots dn}(y) = g(y) \text{ for } y \neq x_i, \text{ and } g^{xi..xn}_{d1\ldots dn}(x_i) = d_i.$$

This means that d_1 is assigned to x_1, d_2 to x_2…and for each *y* not in{$x_1 \ldots x_n$}, $g^{x1..xn}_{d1\ldots dn}$ is the same as *g*.

The satisfaction relation of a BPath expression is defined as follows:

$M, g, w \models t \Leftrightarrow I_{M,w,g}(t) \neq \emptyset$.

$M, g, w \models P(t_1, \ldots, t_n) \Leftrightarrow (\overline{t1}, \ldots, \overline{tn}) \subset I_w(P)$

$M, g, w \models t = u \Leftrightarrow \overline{t} = \overline{u}$, where: t and *u* are terms.

$M, g, w \models \text{not } \varphi \Leftrightarrow M, g, w \not\models \varphi$.

$M, g, w \models \varphi \text{ and } \psi \Leftrightarrow M, g, w \models \varphi \wedge M, g, w \models \psi$.

$M, g, w \models \varphi \text{ or } \psi \Leftrightarrow M, g, w \models \varphi \vee M, g, w \models \psi$.

$M, g, w \models @_s\varphi \Leftrightarrow M, g, g(s) \models \varphi$ for s ∈SVAR.

$M, g, w \models \downarrow s,\varphi \Leftrightarrow M, g_w^s , w \models \varphi.$

$M, g, w \models \downarrow_\pi s,\varphi \Leftrightarrow M, g_{w1,w1\ldots wn}^{s,s[1]\ldots s[n]} , w \models \varphi.$ Where $w_{1\ldots}w_n \in I_{M,g,w}(\pi)$.

$M, g, w \models \downarrow_T x,\varphi \Leftrightarrow M, g_T^x , w \models \varphi.$

$M, g, w \models \exists x\ \varphi \Leftrightarrow M, g_w^x , w \models \varphi.$ for some $d \in D$.

$M, g, w \models \forall x\ \varphi \Leftrightarrow M, g_d^x , w \models \varphi.$ for all $d \in D$.

The interpretation of a path on the model *M*, starting from the state-node *w*, and given an assignment *g* is defined as follows:

$I_{M,w,g}(\pi_1 | \pi_2) = I_{M,w,g}(\pi_1) \cup I_{M,w,g}(\pi_2).$
$I_{M,w,g}(\pi_1 \cap \pi_2) = \{w' |\ w' \in I_{M,w,g}(\pi_1) \wedge w' \in I_{M,w,g}(\pi_2)\ \}.$
$I_{M,w,g}(\pi_1/\pi_2) = \{w'' | w' \in I_{M,w,g}(\pi_1)_w \wedge w'' \in I_{M,w',g}(\pi_2)\}.$
$I_{M,w,g}(\pi\ [\varphi]) = \{w' | w' \in I_{M,w,g}(\pi) \wedge M,g,w' \models \varphi\ \}.$
$I_{M,w,g}(\mathit{self}::N) = \{w|\ \mathit{label}(w) = N \vee N = *\ \}.$
$I_{M,w,g}(\mathit{child}::N) = \{\ (w' |\ (\ wRw' \wedge \forall w''\ wRw'' \rightarrow w'Rw)) \wedge (\mathit{label}(w') = N \vee N = *)\ \}.$
$I_{M,w,g}(\mathit{descendant}::N) = \{w' | wRw' \wedge (\mathit{label}(w') = N \vee N = *)\ \}.$
$I_{M,w,g}(\mathit{descendant\text{-}or\text{-}self}::N)_= I_{M,w,g}(\mathrm{descendant}::N) \cup I_{M,w,g}(\mathit{self}::N).$
$I_{M,w,g}(\mathit{parent}::N) = \{\ w' |\ (\ w'Rw \wedge \forall w''\ w''Rw \rightarrow w''Rw')) \wedge (\mathit{label}(w') = N \vee N = *)\ \}.$
$I_{M,w,g}(\mathit{ancestor}::N)_= \{w' | w'Rw \wedge (\mathit{label}(w') = N \vee N = *)\ \}.$
$I_{M,w,g}(\mathit{ancestor\text{-}or\text{-}self}::N) = I_{M,w,g}[\mathrm{ancestor}::N] \cup I_{M,w,g}[\mathrm{self}::N].$

C. From BPath to XPath

To be evaluated, a BPath expression will be translated into a standard XPath expression, extended with two functions: *Set*, and *Get*, which allow to assign variables and to retrieve their values respectively.

The following table shows, the concrete BPath syntax, and how it is translated to XPath.

TABLE II. BPATH TO XPATH

BPath Abstract Syntax	BPath Concrete Syntax	Translation to XPath 1.0
x (a free variable)	$x	Get($x, self::*)
$@_s\varphi$	$s [φ]	Get(φ, $s)
$\downarrow_T x,\varphi$	$x:=T,φ	Set($x,t,self::*) and φ
$\downarrow s, \varphi$	$s*, φ	Set($s, self::*,self::*) and φ
$\downarrow_\pi s, \varphi$	$s:= π, φ	Set($s, π ,self::*) and φ
s/@q	$s/@q	Get(q, $s)
π/@q (q∈ FUN ∪ ATTD)	π/@q	Get(q, π, self::*)

Quantified expression cannot be expressed in XPath 1.0 [3]. It is possible by using XPath 2.0 [4], as follows:

$$\exists x\ \varphi\text{: some \$x in D satisfies } \varphi$$
$$\forall x\ \varphi\text{: every \$x in D satisfies } \varphi$$

Listing 1 presents the *Get* and *Set* functions. We suppose that '*Eval()'* is a function provided by the framework to evaluate an XPath expression. *ɵ(q,w)* is a function returning the value of a dynamic attribute, *g* is an array storing variables and theirs values.

```
Set($x, t, w){g[x]=Eval(t,w), return true ;}
Get(φ , $s) :{  return Eval(φ, g[s])}
Get($x, w){return g[x];}
Get(q, $s) { return Get(q, g[s])  }
Get(q, π, w) { return sequence:{Get(q,w') / w'∈ Eval(π,w)}
Get(q, w) { if q ∈ ATTS ∪ FUN return Eval (q,w) else  if
(q∈ATTD) return ɵ(q,w) }
```

Listing 1 *Get* and *Set* functions

D. Linear Temporal Logic with BPath

Linear temporal Logic is a special type of modal logic: it provides a formal system for qualitatively describing and reasoning about how the truth values of assertions change over time [5]. LTL provides four future operators with the following meanings: X(φ): φ should be true on the next state, F(φ): means that φ should be true at least once in the future, G(φ): φ should be true every time in the future, φ U ψ: φ has to be true at least until ψ, which is true now or in the future. These operators can be represented in BPath as follows:

- X(φ): child::*[φ].
- F(φ): descendant-or-self::*[φ].
- G(φ): not(descendant-or-self::*[not(φ)]).
- φ U ψ: $x*, F($y*, $x[F($y=self ::* ∧ ψ) ∧ G (F ($y=self ::*) → φ)]).

VI. APPLICATION SCENARIO

In this section, we will show, through a concrete execution scenario, how BPath can be used to monitor a business process execution.

Let us assume that the car rental system manages three cars *(RedCar, GreenCar, BlueCar),* and receives requests from three clients *(John, Mark and Bob),* that we consider as web services interacting with the CRS business process: First, John sends a request for red car. His credit card will be rejected, but he tries again and gets the car reservation. Mark requests a green car, gets a reservation and keys, and then receives a payment confirmation after returning the keys. Bob requests the same car as Mark and obtains a reservation.

At run time, messages exchanged between different instances of the process and external partners will be captured and stored in the event log.

Definition 1: An event log is a collection of entries el = (name, (key=value), (key=value)...., InsId, T), where: *name* is the name of the event, (key=value) ...are list of items and their values contained within the event, *InsId* is an Instance identifier of the process instance concerned with the event, *and T* is a timestamp recording the time the event occured. Listing 2 shows an example of an event log, generated from the supposed execution scenario of the CRS business process.

```
L1 : RentInfo: ClientInfo=John; CarInfo=RedCar, InstId=1, T=1
L2: CarReservation: carReserved=yes, InstId=1, T=3
L3: CardRejected: cardInfo=798799979879, InstId=1, T=5
L4: RentInfo: ClientInfo=Mark; CarInfo=GreenCar, InstId=2, T=8
L5: CarReservation: carReserved=yes, InstId=2, T=10
L6: BankInfo: cardInfo=798799979879, InstId=2, T=12
L7 : RentInfo: ClientInfo=John; CarInfo=BlueCar, InstId=3, T=15
L8: CarReservation: carReserved=yes, InstId=3, T=17
L9: Keys: keysOut=KY123, InstId=2, T=19
L10: RentInfo: ClientInfo=Bob; CarInfo= GreenCar, InstId=4, T=22
L11: CarReservation: carReserved=yes, InstId=4, T=24
L12: Keys: keysIn=KY123, InstId=2, InstId=2, T=26
L13: BankConfirm: payeConfirmed =yes, BankTransation=Trans0001,
InstId=2, T=28
```

Listing 2 Event log

Additionally, the business protocol will generate events related to transition from a state to another state, when a message is received or sent to or by an instance of the process. These events are stored in the state log.

Definition 2*:* A state log (SL) is an XML tree of nodes (states-nodes): w_1, w_2, w_3...where w_2 is the unique child node of w_1, w_3 the unique child node of w_2, etc. Each state-node has a name $s_j \in LAB/\ s_j=label(\mathrm{w_i})$,, and two attributes, *InsId* (instance identifier)∈ATTS, and *T (*timestamp*)*∈ ATTS.

Listing 3 shows the states log generated from the supposed execution scenario of the CRS business process.

A BPath expression will be evaluated over the state log. But as we can see, state log does not contain a lot of information about the execution, because the real events are stored in the event log. Execution information can be retrieved and linked to a state-node through dynamic attributes.

```
<S1  InstId="1" T="0">
 <S2 InstId="1" T="2">
  <S3 InstId="1" T="4">
   <S4 InstId="1" T="6">
    <S1 InstId="2" T="7">
     <S2 InstId="2" T="9">
      <S3 InstId="2" T="11">
       <S4 InstId="2" T="13">
        <S1 InstId="3" T="14">
         <S2 InstId="3" T="16">
          <S3 InstId="3" T="18">
           <S5 InstId="2" T="20">
            <S1 InstId="4" T="21">
             <S2 InstId="4" T="23">
              <S3 InstId="4" T="25">
               <S6 InstId="2" T="27">
                <S7 InstId="2" T="29"/>
               </S6>
              </S3>
     {…}
</S1>
```

Listing 3 State log

In BPath, the value of a dynamic attribute at state-node *w* is defined by a function *θ (q, w)*, which extracts the last value of *q* from the event log, before that state node w occurs, as follows:

```
θ(q,w):
Begin
   Let q∈ el1 / el1 ∈ Event log ∧ el1.InstId=Eval(@InstId, w) ∧
   el1 .T<Eval(@T, w) ∧ ∀ el2  ∈ Event log: q∈ el2 ∧ el2.InstId=
   el1.InstId ∧  el2.T<Eval(@T, w))→ el2.T< el1.T;
   return q.value;
End
```

For instance, the following BPath expressions, when evaluated at T>4, will return:

- *S1/S2/@ClientInfo={John}.*
- *S1/S2/S3/@ClientInf={john}.*
- *S1/S2/S3/@ carReserved={yes}.*

Now, the monitoring properties and queries presented in Section III can be expressed using BPath as follows:

a) Check that in case where credit card of a client is rejected, the client should wait one hour to be able to get a car reservation. We formulate this property in BPath as follows (P1):

G(self::S7[$S7*, @CardRejected →
not(F(self::S3[@CleientInfo=$S7/@ClientInfo and (@T-
$S7/@T)<60]))]).

In this property, we check that every time in the future a credit card of a client is rejected (can be checked at state *S7*), the concerned client should not get a car reservation (we check a state *S3* following the previous *S7*), knowing that the elapsed time (between *S3* and *S7*) is less than one hour.

b) A client should not get a car reservation when the keys are taken by another client. This property can be expressed using BPath as follows (P2):

G (self::S5[$S5*, F(self::S3[$S3*, @CarInfo =
$S5/@CarInfo] → $S5[F(slef::S7[@CarInfo =
$S5/@CarInfo and = @T< $S3/@T])]).

In this property we express that whenever keys of a car is sent (at state *S5*). Then, every time in the future where a reservation for the same car is done (at state *S3*), it should be the case that the keys of this car were returned before (if there exist a state *S7* after *S5* but before *S3*, where the keys of the car are returned)

As we can see from the previous execution log, the properties (P1, P2) are violated respectively at:

- L8 (see event log): when John obtains a car reservation, knowing that his credit card was rejected less than one hour ago (see L3).
- At line L11: the green car was reserved for Bob (at L11), but this car is still assigned to Mark (L9), and the keys of the car are returned by Mark only after (L11), exactly at (L12).

BPath is also a query language that can be used to return statistical indicators on the execution of a business process:

c) Calculating average time to make a car reservation (Q1):

sum(descendant-or-self::S1[$S1*, descendant::S3[$S3*, (@InstId=$S1/@InstId)]]/@($S3/@T-@T)) div count(descendant::S3).

In this query we start by calculating the sum for all process instances of the time to reach the state *S3* (the reservation state) from the state *S1* (the start state), then dividing the obtained sum on the number of reservations. We use two functions (sum and count) to respectively calculate the sum and the number of elements of a sequence.

d) Count the number of rejected credit cards we write in BPath (Q2):

Count(descendant-or-self::S7[@CardRejected]).

The previous list of monitoring properties and queries provides an overview on how to use BPath to monitor business processes. Additional functionalities can be expected when using BPath within XQuery, and by adding new built-in functions.

VII. Conclusion and Future Work

In this work, we provided a framework for business protocol monitoring. First, we have presented the business protocol abstraction. Then, we have presented BPath, the underlying monitoring language. Finally, through a case study, we have shown how the monitoring framework can be used to monitor business protocol. To summarize, we have developed a monitoring framework that mainly displays the followings features:

- The use of a simple messages based abstraction.
- A single expressive language for expressing both monitoring properties, and queries. However BPath is familiar to those who already use XPath language.
- Monitoring properties and queries can be dynamically specified during the execution of the process,
- Non-intrusive monitoring framework, because it is executed in completely separated way from the business process.

Our future work will be devoted to the design of methods to analyze and explain the reason of the deviations, and move towards resolving them as soon as they occur.

References

[1] A. Metzger and K. Pohl, “Towards the Next Generation of Service-Based Systems: The S-Cube Research Framework,” Advanced Information Systems Engineering, 2009, pp. 11-16.

[2] B. Benatallah, F. Casati, and F. Toumani, “Analysis and Management of Web Service Protocols,” Conceptual Modeling – ER 2004, 2004, pp. 524-541.

[3] J. Clark and S. DeRose, XML Path Language (XPath) Version 1.0, W3C, 1999.

[4] M. Kay, D. Chamberlin, J. Robie, M.F. Fernández, J. Siméon, S. Boag, and A. Berglund, XML Path Language (XPath) 2.0, W3C, 2007.

[5] E.A. Emerson, “Temporal and modal logic,” Handbook of Theoretical Computer Science, 1995, pp. 995--1072.

[6] P. Blackburn and M. Marx, “Tableaux for Quantified Hybrid Logic,” Automated Reasoning with Analytic Tableaux and Related Methods, 2002, pp. 259-286.

[7] L. Baresi and S. Guinea, “Towards Dynamic Monitoring of WS-BPEL Processes,” Service-Oriented Computing - ICSOC 2005, 2005, pp. 269-282.

[8] L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, and P. Spoletini, “A Timed Extension of WSCoL,” Web Services, IEEE International Conference on, Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 663-670.

[9] K. Mahbub and G. Spanoudakis, “Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience,” IN ICWS ’05, 2005, pp. 257--265.

[10] C. Beeri and A. Eyal, “Monitoring business processes with queries,” IN VLDB, 2007.

[11] S. Hallé and R. Villemaire, “Runtime monitoring of web service choreographies using streaming XML,” Proceedings of the 2009 ACM symposium on Applied Computing, Honolulu, Hawaii: ACM, 2009, pp. 2118-2125.

[12] D. Chamberlin, J. Snelson, J. Robie, and M. Dyck, XQuery 1.1: An XML Query Language, W3C, 2009.

Service Planning in Multi-Layer Networks Considering Physical Constraints

Shu Zhang, Lothar Kreft and Ulrich Killat
Institute of Communication Networks, Hamburg University of Technology
Email: s.zhang, kreft, killat@tu-harburg.de

***Abstract*—In the daily work of network operators, some traffic engineering tasks are often encountered, e.g., to create new logical links over the physical layer considering the efficient utilization of network resources; to establish new end-to-end paths across the network with minimum cost in order to support emerging data transfer services; to install physical disjoint paths for some critical services where fault tolerance is desired, etc. Since these tasks are by nature interrelated, we propose an integrated optimization framework to solve them as a unified planning problem. Both an Integer Linear Programming model and a Simulated Annealing based optimization method are discussed in this paper. Because optimization in multi-layer networks is known to be much more complicated than that in a single layer, special care has been taken in our model to alleviate the scalability problem. The framework has been implemented as a commercial tool for traffic planning. The numerical tests have shown that the corresponding tasks in real scale network can be efficiently handled.**

***Keywords*-Physical Disjoint; SRLG; Traffic Engineering; ILP; Simulated Anealing**

I. Introduction

One of the major challenges in short term network management is to establish a number of new end-to-end paths with dedicated resource assignment for the emerging request of data transfer services, using the currently available spare resources in the network. A network operator usually does not handle each service directly. Instead, a fixed path with a bulk of bandwidth allocated at each hop is provided to an aggregation of individual data transfer services with some common properties, e.g., same source and destination, similar QoS requests, and same protection mechanism. From the operator's point of view, the aggregation of services is considered as an abstract *demand* to be routed across the network.

The state of the art solution is to calculate a minimum-cost solution at the moment for each upcoming demand using a shortest path algorithm. However, such a greedy approach is known to be sub-optimum in case of multiple demands, because the demands are interrelated due to the competition for common network resources. The situation becomes even more complicated when multiple network layers are taken into consideration. In practical solving approaches nowadays, the consideration of inter-layer relations is mostly intuitive or based on personal experiences of the planner. This may result in worse solutions than planning in a single layer only. Furthermore, some critical user requests could be left unguaranteed. A typical problem raised by network operators is to ensure physical disjoint paths for the protection of important services in a multi-layer setting. This problem and our solution based on a two-layer reference model will be extensively discussed in this paper.

A great number of researches have been carried out on multi-layer optimization problems, typical works includes [1][2][3] in which multi-layer problems are implicitly discussed in the background of WDM networks, and [4][5] where general purpose multi-layer planning models are suggested. One common conclusion is that multi-layer optimization is much more complicated than single layer ones, and to strictly model multi-layer data structures may result in prohibitive scale when dealing with practical problems. In this paper, we propose an approach which reduces the multi-layer model as much as possible into a single layer one. We will formulate a core TE model for the standard single layer planning problem, where the critical multi-layer features appear as extra constraints, and the non-critical features are shifted into heuristic algorithms executed before or after the main TE process.

The rest of the paper is organized as follows. Section II introduces the problem settings and the Integer Linear Programming (ILP) formulation of our Traffic Engineering model. In Section III, a greedy planning method and its extension to a Simulated Annealing (SA) based method are presented as alternative heuristic solutions. Some numerical results in solving the ILP and the heuristic models for a test scenario are presented in Section IV. The conclusions are given in the last part.

II. The Problem Definition and the ILP Model

A. Problem Setting

Let's consider the task to route a set D of end-to-end demands (with given resource requests) across a network described by a graph $G(N, E)$, where N is the set of nodes and E is the set of links. The objective is to find a minimum-cost solution where all constraints are held. Without loss of generality, our multi-layer network model is defined by following properties:

1) There are two layers in the network, physical layer $G_{phys}(N_{phys}, E_{phys})$ and logical layer $G_{log}(N_{log}, E_{log})$. Since logical nodes are a subset of physical nodes($N_{log} \subseteq N_{phys}$), we define $N_{phys} = N$;

2) The logical layer occupies a part of the physical resources. Therefore, there are some remaining free resources at both physical and logical layer.
3) Free resources in the logical layer can be directly used to route an end-to-end demand, while free resources in the physical layer must be converted into logical links before being used to route any demand.
4) The routing of each logical link in the physical layer is known; new logical links can be arbitrarily created when there are enough resources in each of its physical hops.
5) The settings of all existing logical links and routed demands remain constant. The capacity of logical links, as well as the routing of demands and logical links cannot be changed.

Note that the last property originates from the practical request of the network operators. The purpose of this conservative constraint is to make sure that no active services could be disturbed due to the accommodation of new demands. Consider another extreme case: Free reconfiguration of all logical links is allowed. In this case, we can setup an analytical model where all free resources in logical links are returned to the corresponding links at the physical layer. After this step, all logical links can be safely removed from the graph since they can no longer influence the routing decisions. Finally, the optimization will be carried out in a topology identical to the physical network, and thus becomes equivalent to a single layer TE problem. After the optimization, we only have to modify the related logical links in the original network according to the solution. With this approach, the sub-optimality of resource utilization due to the ”bundle effect” is not an issue, and may therefore result in better resource effectiveness than our definition above. But, such kind of solution may require a large number of reconfigurations, which is a tedious task and in many cases a major source of error.

In the following part of this paper, we will focus on our problem setting with the properties presented above, in a network G defined as follows.

$$\begin{aligned} G &= G_{phys} \cup G_{log} = G(N_{phys} \cup N_{log}, E_{phys} \cup E_{log}) \\ &= G(N, E_{phys} \cup E_{log}) \\ &= G(N, E),\ E = E_{phys} \cup E_{log} \qquad (1) \end{aligned}$$

This equation explains our attempt to convert most multi-layer optimization features into a single layer model, as shown in Fig.1. Both logical and physical links in Fig.1a which have spare resources at the moment are represented by an *abstract* link in Fig.1b, with capacities equal to their spare resources. In the real operation, spare resources on a physical link must be organized into logical link(s) to be eligible for the routing of demands. However, since the operation of creating a new logical link on a selected physical segment is not an optimization issue, it is taken off from our optimization model without compromising the optimality. Here, we consider the spare resources also as *abstract*, and no longer differentiate between physical and logical links that can eventually be used to support demands.

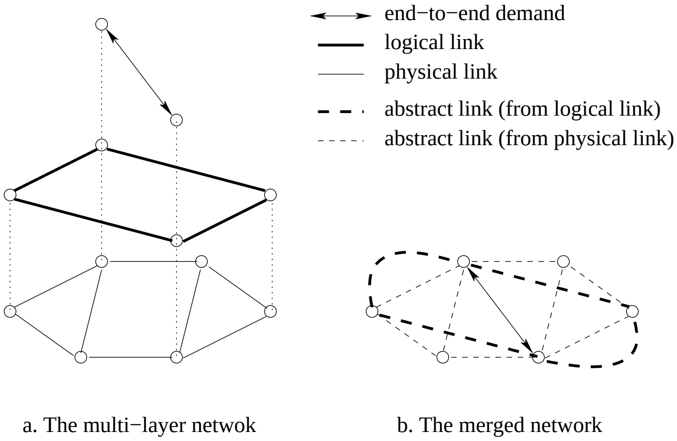

end−to−end demand
logical link
physical link
abstract link (from logical link)
abstract link (from physical link)
a. The multi−layer netwok
b. The merged network

Figure 1. Merge of the physical and logical layer

Note that both Eq.1 and Fig.1 are missing the information of the routing of logical links over physical links, which must be specially modeled.

Following the requests of network operators, we define 2 types of end-to-end demands:

1) Type 1 (D_1): requiring a single end-to-end path with dedicated resource allocation.
2) Type 2 (D_2): requiring a pair of physical disjoint end-to-end paths with dedicated resource allocation along both paths, so that any single failure in the physical layer can be tolerated.

The objective of the optimization is to find a routing solution for each demand, while the link load does not exceed the limit of the available capacity, and the cost due to resource consumption is minimized. In our planning model, all available free resources at any layer become resources in the abstract links of the merged network, based on which a traffic engineering algorithm is carried out. Eventually, the creation of logical links is accomplished according to the results of the optimized routing of demands.

B. The Routing of Demands

In order to establish an end-to-end connection for type 1 demand, a set of flow continuity equations are established. We define a set of binary variables:

$$x_1(i,e) = \begin{cases} 1 & \text{if demand } d \in D_1 \text{ traverses link } e \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

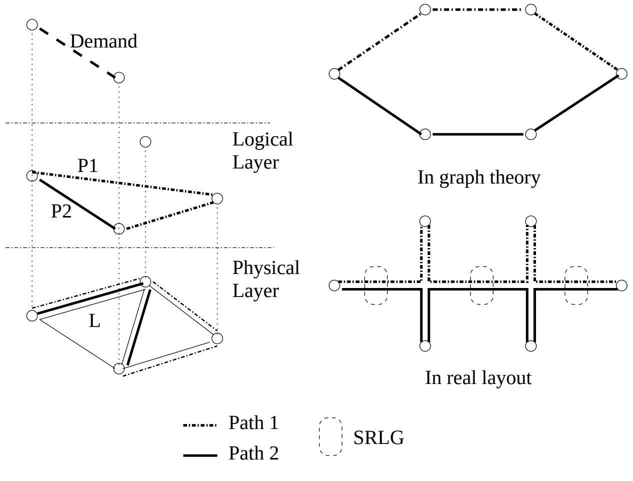
Assume $u, v \in N$ are the end nodes of demand i, and $e(m,n)$ denotes a directional link e from node m to n, then the flow continuity constraint is as follows:

$$\begin{aligned}\delta_{u,j} + \sum_{e(m,j)\in E} x_1(i,e) &= \sum_{e'(j,n)\in E} x_1(i,e') + \delta_{v,j},\ \forall j \in N \\ \delta_{a,b} &= \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Eq.3 means that the traffic entering any node j must be equal to that leaving the node, with the exception at the source and destination nodes of the demand.

For type 2 demands, the flow continuity equations are in principle the same. Here, a pair of disjoint paths for each demand is required. We will show that the physical disjointness in the multi-layer model can be well modeled by disjoint conditions of nodes and Shared Risk Link Groups ($SRLG$) [8][9][10] in the layer-merged model.

While node disjoint condition is obvious, an $SRLG$ may originate from two cases in a multi-layer network. The first case is due to the routing of logical links over the same physical link, as shown in Fig.2a. A single event which destroys the physical link L will also destroy logical link P_1 and P_2. Since they share the risk of the same event, we state that the set of links $\{L, P_1, P_2\}$ forms a shared risk link group S. Consider disjoint paths calculation: If one path traverses one of the links in a risk group, the other path should avoid taking any of the links in the same group. Given the routing of all logical links, we can find as many risk groups as the number of physical links in the network, each containing a physical link and all the logical links routed over it. These groups can be obtained by a deterministic analysis process, denoted as *routing related groups*.



a. Non−disjoint due to routing of logical links

b. Non−disjoint due to cable layout

Figure 2. Situations of non-disjoint paths

The second case originates from cable layout. As shown in Fig.2b, although a pair of disjoint paths can be calculated, it may still be risky due to the layout of the physical links. I.e., because cables can be placed in the same bundle or same duct, a single event that destroys one cable should also destroy all others in the same place.

Considering both cases in the optimization model, a preprocess before running the main optimization is required: firstly, an analysis should be made to find out all such bundles/ducts, referred to as a *risk area*; then the *routing related groups* traversing the same risk area are merged to become an SRLG. Note that any standalone physical link is considered as an SRLG with only one physical link. With this model, the physical disjoint constraint in a multi-layer problem is converted to the equivalent condition in the merged single-layer network: the two paths of a type 2 demand should not traverse the same SRLG.

To establish a pair of paths for every type 2 demand, we define a similar binary decision variable as Eq.2:

$$\begin{aligned} x_2(i,e,p) &= \begin{cases} 1 & \text{the } p\text{th path of } d(\in D_2) \text{ traverses } e \\ 0 & \text{otherwise} \end{cases} \\ p &\in \{1,2\} \end{aligned} \tag{4}$$

Under this definition, the flow continuity condition is very similar Eq.3, with the same equation set for every $p \in \{1,2\}$, while the SRLG disjointness of the paths is guaranteed by Eq.5:

$$\begin{gathered} x_2(i,e,p) + x_2(i,e',p') \leq 1 \\ \forall i \in D, \forall p,p' \in \{1,2\}, p \neq p', \forall e,e' \in S, \forall S \end{gathered} \tag{5}$$

The above constraint means that $x_2(i,e,p)$ and $x_2(i,e',p')$ cannot both be 1, which implies the path p and p' of the demand i must not traverse links in the same SRLG S. However, Eq.5 does not check disjointness of paths at each nodes. The following two equations ensures node disjointness, where the variable $u(i,n,p)$ tracks the intermediate nodes of a path (see Eq.6), and the node-disjointness constraint (see Eq.7) is similar to Eq.5.

$$\begin{aligned} u(i,n,p) \text{ is binary, }, &\qquad \forall i \in D_2, n \in N, p \in [1,K] \\ x_2(i,e(m,n),p) \leq u(i,n,p), &\qquad \forall i \in D_2, e \in E, p \in [1,K] \\ n \text{ is not an end node of } i & \qquad\qquad\qquad\qquad\qquad\qquad (6) \\ \sum_{p\in\{1,2\}} u(i,n,p) \leq 1, &\qquad \forall i \in D_2, n \in N \qquad\qquad (7) \end{aligned}$$

Note that the SRLG and node-disjointness equations discussed in this section are not restricted to a pair of disjoint paths, the formulation can be naturally extended to the K disjoint paths model.

C. Other Constraints and the Optimization Objective

1) Link utilization: Link utilization should not exceed capacity, where $R(i)$ is the resource required by demand i, and $Cap(e)$ indicates the spare capacity at link e.

$$\sum_{i_1\in D_1} x_1(i_1,e)\cdot R(i) + \sum_{i_2\in D_2, p\in\{1,2\}} x_2(i_2,e,p)\cdot R(i_2)$$

$$\leq Cap(e),\ \forall e \in E \tag{8}$$

2) Objective: As in standard TE problems, we wish to minimize the total cost to support the demands. Let $Price(i,e)$ indicate the cost of demand i taking a unit of resource at link e, then the objective function can be formulated as:

$$\begin{aligned} &Minimizing: Cost\\ &Cost = \sum_{i_1 \in D_1, e \in E} x_1(i_1,e) \cdot R(i) \cdot Price(i_1,e) +\\ &\sum_{i_2 \in D_2, e \in E, p \in \{1,2\}} x_2(i_2,e,p) \cdot R(i) \cdot Price(i_2,e) \end{aligned} \tag{9}$$

With this optimization objective and all above constraints, the default model to solve the TE problem has been established. It is a linear problem and can be solved by an LP solver. Since we have made no compromise on any feature, the optimum solution of the problem can be obtained.

3) Relaxations: For the default model, all the constraints hold strictly. If there are any violations, e.g., the network resource is not sufficient to support all demands, or a pair of strictly disjoint paths does not exist for some type 2 demands, the result of the optimization will be ”infeasible”. Practically, a planner may wish to know more. One of the typical FAQ is: We know it is hard to keep all constraints, but if we tolerate some violations, what can still be achieved?

Here we discuss two kinds of tolerances. The first one is to allow some demands eventually be left unrouted. Define a binary variable $e_x(i)$ as in Eq.10, and a punishment cost $C_{ex}(i)$ to indicate the increment of total cost if demand i cannot be routed.

$$e_x(i) = \begin{cases} 1 & \text{if demand } i \text{ is routed}\\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Now, the flow continuity condition (Eq.3) should be slightly modified, so that the incoming/outgoing traffic is no longer guaranteed to be 1 at the end nodes of each demand. Instead, it depends on the value of $e_x(i)$:

$$\begin{aligned} \delta_{i,u,j} + \sum_{e(m,j) \in E} x_1(i,e) &= \sum_{e'(j,n) \in E} x_1(i,e') + \delta_{i,v,j},\\ \forall j \in N,\ \delta_{i,a,b} &= \begin{cases} e_x(i) & \text{if } a = b\\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

The other tolerance is related to the physical disjointness. We define two integer variables: $e_s(i,S)$ indicates that the paths of demand i traverse the same SRLG S, and $e_n(i,n)$ indicates the paths of demand i traverse the same node n, both taking 0 for no violation and positive integer values for so many times of violation. Besides, punishment costs $C_{es}(i,k)$ and $C_{en}(i,n)$ represent the increment of cost when these kinds of violation happen. We need to modify the SRLG and node disjoint constraints (Eq.5, 7) as follows:

$$x_2(i,e,p) + x_2(i,e',p') \quad \leq \quad 1 + e_s(i,S), ... \tag{12}$$

$$\sum_{p \in \{1,2\}} u(i,n,p) \quad \leq \quad 1 + e_n(i,n), ... \tag{13}$$

In the objective function, we set that each violation brings an extra punishment cost in addition to the regular cost. Therefore, the optimization of minimizing the objective can proceed in the direction of reducing violations too. The formulation is as follows:

$$\begin{aligned} & Minimizing: Cost + CostT\\ Cost &= \text{... (as defined in Eq.9)}\\ CostT &= \sum_{i \in D_1 \cup D_2} (1 - e_x(i)) \cdot C_{ex}(i)\\ &+ \sum_{i \in D_2} \Big[\sum_{k \in S} e_s(i,k) \cdot C_{es}(i,k)\\ &+ \sum_{n \in N} e_n(i,n) \cdot C_{en}(i,n) \Big] \end{aligned} \tag{14}$$

The values of C_{ex}, C_{es} and C_{en} are very critical to the result of the optimization. The relationship between these punishment costs indicates the tradeoffs among several factors: to support more demands, to save cost, as well as to reduce the amount of rule violation. In our model, we have made the clear setting:

$$C_{es} \approx C_{en} \gg C_{ex} \gg Cost \tag{15}$$

Here, $Cost$ is the regular cost due to resource utilization. This setting implies that respecting the disjoint condition is most important. If we set C_{es} and C_{en} to $Infinite$, or remove the tolerance terms in Eq.12 and Eq.13, then no violation is allowed. Under this principle, a decision which can accommodate more demands is always better than any other solution with less regular cost but also less demands being routed.

III. The Simulated Annealing (SA) model as alternative solution

In the previous section, we have modeled every feature of the optimization problem as an ILP. Therefore, theoretically the optimum solution can always be obtained. However, the disjoint routing problem with SRLG constraints is proved to be NP-complete [9], and our numerical results that will be introduced in the next section also tend to confirm this property. As an alternative, we introduce our basic greedy algorithm for the same planning problem; then, the greedy algorithm is taken as the core element of a meta heuristic model, for which we chose the Simulated Annealing (SA) model. Since it aims at searching for a satisfactory solution rather than the optimum one, the SA approach can efficiently avoid the difficulties of an NP-complete problem.

A. The greedy algorithm

As discussed in the introduction, a straightforward solution to plan multiple demands in a network is to route all demands sequentially. This is a fully deterministic process, which can be expressed by the pseudo code of Algorithm 1.

Algorithm 1 Greedy TE Algorithm (GTA)

```
Given a fixed sequence $S$ of demands $D(i), i = 1..N_S$
for $i = 1$ to $N_S$ do
    for each link $l$ in the network do
        if $l$ has sufficient resource to accommodate $D(i)$ then
            set the cost of $l$ as its cost for $D(i)$
        else
            set the cost of $l$ as $Infinite$
        end if
    end for
    calculate a minimum-cost solution $sol(i)$ for $D(i)$
    if $sol(i)$ exists then
        record $sol(i)$ in the solution set $Sol$
        for each link $l$ in $sol(i)$ do
            add the cost of $l$ to cost $C$
            update resource utilization at $l$
        end for
    else
        record no solution for $D(i)$ in $Sol$
        add the punishment cost to $C$
    end if
end for
Return the solution set $Sol$ and cost $C$
```

Here, if $D(i)$ is type 2, then a pair of paths with minimum cost sum should be calculated. If the SRLG condition is not considered, know methods like Suurballe's algorithm [6][7] can guarantee the optimum solution. To the best of our knowledge, no heuristic algorithm has been found to be able to guarantee a minimum cost SRLG-disjoint solution. In our greedy algorithm, the *trap avoidance* algorithm suggested by the authors of [10] is used.

B. The simulated annealing algorithm

Because each shortest path is adaptively calculated according to the available network resources at the moment, the above greedy method is to some extent optimized. The quality of the solution is generally better than the intuitive solution of a human planner. An open issue is that GTA (Algorithm 1) depends on a given sequence of demands. With a different sequence, a different set of paths and different overall cost will be obtained. A simple method to take care of the observation is to repeat the same operation: Randomly modify the sequence, and call GTA; the best solution that occurs in this process is taken as the final solution. We refer to this method as *Random Solution Generation (RSG)*.

However, according to our tests, if we combine GTA with some well-known meta heuristics, solutions with better quality (more demands accommodated, less cost) than that of RSG can be obtained in the same calculation time. Here, we present our model with simulated annealing which helps to decide on a suitable sequence for GTA. The pseudo code is shown in Algorithm.2.

Algorithm 2 Simulated Annealing based TE

```
Start with a current sequence $S$ and cost $C \leftarrow GTA(S)$
$S_{best} \leftarrow S$, $C_{best} \leftarrow C$
for reset $round = 0$ to $R$ do
    $S \leftarrow S_{best}$, $C \leftarrow C_{best}$
    for $schedulestep = 1$ to $N$ do
        sequence $S' \leftarrow neighbor(S)$
        $C' \leftarrow GTA(S')$
        if $C' < C$ then
            $S_{best} \leftarrow S'$, $C_{best} \leftarrow C'$
        end if
        the current temperature $t \leftarrow T(i)$
        if $P_{trans}(C, C', t)) > random()$ then
            $S \leftarrow S'$, $C \leftarrow C'$
        end if
    end for
end for
Return $S_{best}$ and $C_{best}$
```

The major functions in the algorithm are:

1) The function $GTA(S)$ is the greedy method Algorithm.1 with the given demand sequence S. The cost C of its solution is then passed to the simulated annealing process.
2) The SA algorithm will reset for R times. At the beginning of each reset, the current state is set to the best solution obtained so far, i.e., the sequence which brings the lowest overall cost.
3) The function $neighbor(S)$ is designed to move a fraction of randomly chosen demands in the sequence to the front of the modified sequence. Therefore, the new sequence S' is similar to the original S, and theoretically the whole solution space can be explored by this operation without preference of any specific pattern.
4) There will be N steps till the temperature drops from the initial T_{max} to 0. The function $T(i)$ controlling the temperature dropping according to the time i is referred to as a *cooling schedule*. Here, an exponential schedule $T(i) = \alpha^i T_{max}, 0 < \alpha < 1$ is chosen based on our tests.
5) If a neighbor state S' is better (with lower cost) than its original state S, then a state transition to S' will definitely take place. When S' is worse than S, the key idea of *simulated annealing* is to allow the transition according to probability, so that the searching process may have chances to let the current state move out from the *local optima*. The probability of a transition

to a worse state is reduced with the dropping of temperature, therefore the state tends to stabilize around some good solutions. When the temperature reaches 0, the SA becomes a pure greedy algorithm. According to our tests, the exponential transition probability formula suggested by Kirkpatrick et al.[11] has shown good performance, i.e., $P_{trans}(C, C', t) = exp(\frac{C-C'}{K_B t})$.

According to our tests, it is capable of obtaining a satisfactory solution within much less time than that required by the ILP model (Section IV), and the solution quality is also better than that obtained by running the *Random Solution Generation (RSG)* for the same time. Although the final solution obtained by SA is inherently sub-optimum. its quality is significant better than the path-oriented methods (even combined with ILP), in which the paths for each demand are selected from a set of pre-calculated candidates.

IV. Numerical Result

Here we show the optimization result obtained with the topology of X-WiN network [12], which is a German scientific research network with nodes located in its major cities. The physical layer of our test case consists of 54 major nodes and 81 links from X-WiN network. The capacities of links range from 1Gbit/s to 20Gbit/s, the same setting as established in X-WiN. Then, 100 logical links are randomly generated using the resource of physical links, which organizes 70% to 80% of the physical resources into the logical layer. Finally, 20% to 80% (random even distribution) of the total capacity at each logical link is marked as occupied to emulated the current network usage.

A test has been carried out to show the influence of SRLG conditions (in the above model, SRLGs are only due to logical links routed over the same physical link). We repeatedly generate type 2 demand with random source and destination nodes. The resource request of such demands are set to 0 so that resource shortage in the network is not a problem. At first, by ignoring all SRLG relationship and only considering the pure graph information like Fig.1b, we use Suurballe's algorithm to find a pair of disjoint paths for each demand. This is roughly what a planner can do in a single-layer TE model. Then, we check if such a solution violates the SRLG-disjoint condition. In our experiment of 10 thousand times random demands generation, 74.2% solutions obtained in this way are in fact non-disjoint, i.e., the 2 paths in a solution traverse at least one common physical link.

Now, we test the different solving approaches of the TE model. To avoid the mixed effect of performance measuring, the test is set to find a minimum cost solution for x type 2 demands, each with randomly generated source and destination. Violation of the disjoint condition is not allowed in this test. The resource requests of each demand are randomly generated and evenly distributed. The range of distribution is adjusted each time to let the overall demand slightly hit the bottleneck of network resources, i.e.: In the solution obtained by simulated annealing, roughly 90% of the demands can eventually be routed. Then, the result is compared to the greedy solution, as well as the optimum solution obtained by the ILP model. All methods follow the principle of accommodating as many demands as possible (Section II-C3).

The quality of the solutions obtained by the different solving approaches is shown in the following two figures. For ILP solutions, the result is obtained when the gap reaches ≤1%, i.e., at most 1% away from the optimum solution. Fig.3 shows the total number of demands routed in the final solution, and Fig.4 shows the comparison of average routing cost for each demands using the greedy solution as reference.

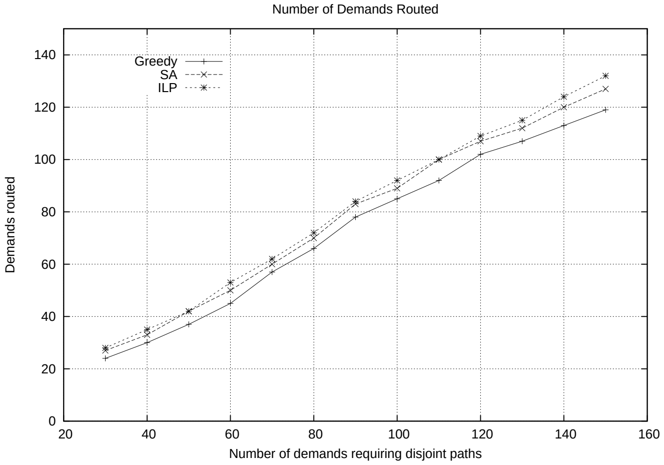


Figure 3. The average cost per routed demands by each method

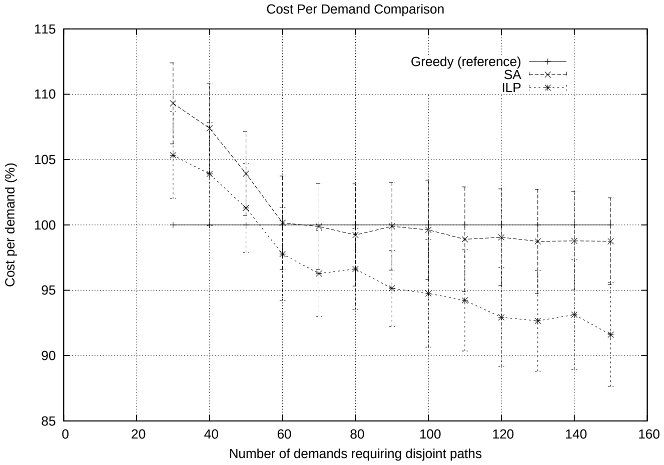


Figure 4. Number of accommodated demands by each method

From the results, it is clear that with the increment of the amount of demands to be considered in one planning task, more demands can be routed with even reduced average cost. The performance of SA is close to that of ILP, and significantly better than that of the greedy solution. Fig.5 shows the solving time of the ILP model and the simulated annealing model with respect to the amount of demands. The solving time of SA is multiplied by 10 to display the curve more clearly. The LP solver is ILOG CPLEX version 11.0.0, and we are using a PC with 3GHz processor and 1G memory.

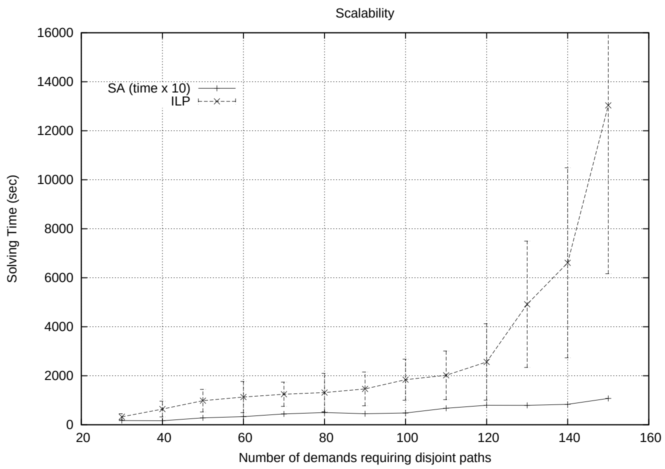


Figure 5. The solving time of the basic TE model

The SRLG disjoint routing problem has been proved to be NP-complete [9]. Indeed, Fig.5 shows a tendency of exponential growth of solving time of the ILP model with the increasing number of demands. The calculation time is also related to whether the bottleneck has been reached, and to which extent. When network resources are adequate, less calculation time (e.g., around 1 hour for 150 demands) is observed in our tests. In comparison, the simulated annealing model has demonstrated a linear increment of solving time due to its fixed cooling schedule and the linear scalability of the greedy sequential method $GTA(S)$ in its core (Algorithm 2). The execution time of the greedy method takes only several seconds within the tested range, therefore it is not shown in the figure.

V. Conclusion

In this paper, we discussed the traffic engineering problem of disjoint route allocation in multi-layer networks, and suggested an analytical model to take the disjoint conditions in physical layer into an integrated optimization approach. Our ILP model is introduced for the basic problem setting, and then extended to support choices like conditional violation of disjoint constraints and best-effort accommodation of demands. Our model has considered the scalability issue often encountered in the multi-layer planning. Besides, a simulated annealing based planning method has also been suggested as an alternative solution approach, aiming at obtaining satisfactory solutions when the solving time of the ILP model is prohibitively high.

References

[1] J. Hu and B. Leida, *Traffic Grooming, Routing, and Wavelength Assignment in Optical WDM Mesh Networks*, In Proc. of IEEE INFOCOM 2004, vol. 1, March 2004, pp.495-501.

[2] K.Zhu, H. Zhu and B. Mukherjee, *Traffic Grooming in Optical WDM Mesh Networks*, Springer Science 2005

[3] E. Kublilinskas and M. Pioro, *An IP/MPLS over WDM network design problem*, in Proc. of International Network Optimization Conference (INOC) 2005, Lisbon, Portugal, 20-23 March, 2005. volume 3, pp.718-725.

[4] S.Orlowski and R. Wessäly, *An Integer Programming Model for Multi-layer Network design*, Konrade-Zuse Center, Berlin. ZIB-Report 04-49 Dec.2004

[5] P. Belotti, A. Capone, G. Carello and F. Malucelli, *Multi-layer MPLS Network Design: The Impact of Statistical Multiplexing*, Computer Networks, Vol.52, Issue 6, pp.1291-1307, April 2008.

[6] J. W. Suurballe, *Disjoint Paths in a Network*, Networks vol.4 (1974) 125-145

[7] J. W. Suurballe and R. Tarjan, *A Quick Method for Finding Shortest Pairs of Disjoint Paths*, Networks, vol.14 (1984) pp.325-336

[8] D. Xu, Y. Xiong, and G. Li, *Trap Avoidance and Protection Schemes in Networks with Shared Risk Link Groups*, IEEE Network, 18(13):36-41, May-June 2004.

[9] J. Hu, *Diverse routing in optical mesh networks*, IEEE Transaction of Communications, Vol.51, pp.489-494, 2003.

[10] Lu Shen, Xi Yang and B. Ramamurthy *Shared Risk Link Group (SRLG) - Diverse Path Provisioning Under Hybrid Service Level Agreements in Wavelength Routed Optical Mesh Networks*, IEEE/ACM Transaction on Networking, Vol.13, No.4, August 2005, pp.918-931.

[11] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi *Optimization by Simulated Annealing*, Science, 13 May 1983: Vol.220. no.4598, pp.671-680

[12] German Research Network (Deutsches Forschungsnetz DFN), *The Scientific Network X-WiN*, http://www.dfn.de/xwin/

Using QoS for Relevance Feedback in Service Discovery: A Preliminary Empirical Investigation

Konstantinos Zachos, Neil Maiden
Centre for HCI Design, City University
London, UK
{kzachos, n.a.m.maiden}@soi.city.ac.uk

Glen Dobson, Pete Sawyer
Computing Department, Lancaster University
Lancaster, UK
{dobsong, sawyer}@comp.lancs.ac.uk

***Abstract*—Service-centric systems pose new challenges and opportunities for requirements processes and techniques. This paper describes our requirements-based service discovery tool that exploits an ontology-based quality specification mechanism to receive early feedback on candidate services that best match quality requirements. An empirical evaluation of the tool is presented that assesses the feasibility of the approach to filtering candidate services based upon quality using real-world scenarios from our industrial partners. The results reveal that commitment to a common ontology helps achieving the desired quality-based filtering.**

Keywords – Service Discovery; QoS; Quality-based filtering.

I. DEVELOPING WITH WEB SERVICES

Service-centric system engineering is an important emerging paradigm in which applications are constructed from reusable component services [1]. One important consequence for requirements processes is that service registries available over the Internet provide users with immediate access to elements of the solution space. In the SeCSE Project [4] we have developed tools and techniques to form and execute queries on service registries from requirements specifications.

SeCSE supports an iterative requirements discovery process as shown in Fig. 1 [5]. Requirements analysts form service queries from requirements specifications to retrieve web services compliant with the requirements. Descriptions of retrieved services are presented to analysts who use them to enable more accurate service retrieval in a cyclical retrieval and requirement refinement process.

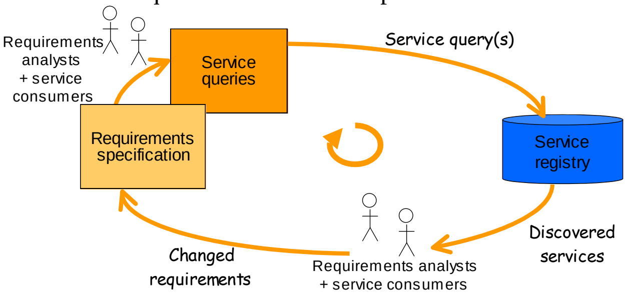

Requirements analysts + service consumers
Service queries
Service query(s)
Requirements specification
Service registry
Discovered services
Changed requirements
Requirements analysts + service consumers

Figure 1. SeCSE's Requirements Process

To ensure industrial uptake SeCSE's requirements process uses established techniques based on structured natural language. Analysts specify service-centric system behaviour with UML use case specifications and required system properties in a testable form with VOLERE shells [7]. The process extends the Rational Unified Process (RUP) [8] without mandating additional specification or service retrieval activities.

Our approach builds on Fischer et al.'s [6] observations about how design queries are incrementally improved by critiquing results from previous queries. Relevance feedback, as this is known, provides information about whether requirements can be satisfied by available services, to guide the analysts to consider alternative build, buy or lease alternatives or explore trade-offs to see whether most requirements can be met at acceptable cost by the available services.

To support this process of requirements-based service discovery we have also developed a means to represent service semantics. Relevance feedback may be used to inform analysts' formulation of requirements based on representations of service semantics held in service registries. Crucially, this may include information about non-functional service characteristics. Of particular interest are the set of characteristics that broadly fall into the area of Quality of Service (QoS). With SeCSE it is possible to get an early indication of the quality of available services and therefore the potential quality achievable in the final application. Along with information about discovered services' functionality, QoS information can be fed back into the requirements in order to help formulate requirements for acceptable QoS characteristics for the future system.

QoS has been identified [13] as a key element in the wider uptake of web services, particularly in a competitive market in third-party services [2]. Here, the consumer has no control over the service provision and implementation, and so assessing QoS is vital in establishing trust in a service. It is therefore desirable for the service consumer to be able to clearly state QoS requirements and identify the level of service compliance with those requirements. In recent years, we have reported new tools and techniques to increase requirements completeness from retrieved web services based on functional requirements [e.g. [14],[16],[17]]. In this paper we present new techniques to match non-functional requirements to service qualities during service selection using an ontology.

There are many QoS properties and for any property, there are often several metrics with which it may be expressed. This creates problems for QoS-informed service discovery since the analyst and service provider may express required and provided QoS properties in different ways.

Where different metrics are used false negatives will result from simple syntactic matching. As we have reported previously [15], a key feature of our tools is that these tools can tolerate such inconsistencies. When we last reported on this work, however, we had not yet performed an empirical evaluation of our work. To remedy this, this paper reports results from an industrial evaluation that was designed to answer our central hypothesis, that *the SeCSE service specification and SeCSE's requirements-based discovery tools are tolerant of mismatches between how a service provider specifies their services' QoS properties, and how a service consumer specifies their QoS requirements.* In other words, the analyst is insulated from needing to know the metrics and units used by different service providers, provided they (the analyst) use a recognized metric/metrics and unit/units when they specify their requirement.

Section 2 describes SeCSE's requirements-based service discovery environment and the QoSOnt ontology. Section 3 introduces the evaluation method, and Section 4 reports results from the evaluation. Section 5 uses this data to answer the central hypothesis and discusses threats to validity of the results. The paper ends with implications of the results for iterative requirements-based service discovery processes and service discovery algorithms to support this process.

II. Discovering and Selecting Services using QoS Requirements

To support SeCSE's requirements process we implemented the SeCSE service specification and discovery environment. It has 4 main components: (i) Service Registries – a federated and heterogeneous mechanism for storing service descriptions along with the Service Specification Tool with which service providers can populate the Service Registries; (ii) the UCaRE Requirements Component, which supports web-enabled specification of requirements and use cases, and formulation of service queries from these specifications; (iii) the Service Querying Component, which uses service queries to discover web services with different types of similarity; and (iv) the Service Explorer Component that displays the descriptions of retrieved services to enable analysts to understand, select between and use these services to discover new requirements.

A. The Test Scenario

To clarify our discussion we will present the test scenario for the evaluation. This has been taken from a system developed by SeCSE's industrial partners that integrates various services into a client within an in-car device. The services include map/point of interest services; weather services; car park booking services; telecoms services, and logistical services [3]. The use case that has been used for the purpose of this evaluation to test the hypothesis described in the introduction is partly shown in Table 1 –*Acquiring Weather Information* use case. Table 2 defines two simple associated requirements. The first, a functional requirement (FR61), specifies what the service shall do, and the second, the non-functional requirement (AR7), specifies desirable qualities of the service.

TABLE I. PART OF ACQUIRING WEATHER INFORMATION USE CASE

Name	Acquiring weather information
Précis	A driver is driving his car. The driver requests weather forecast information for a selected destination using the car's on-board weather forecast information service. The service retrieves information about weather forecasts for the selected destination based on the estimated arrival time. The service displays the information to the driver.
Actors	Driver, On-board weather forecast information service
Problem Statement	The driver needs information that is relevant and useful to him/her and his/her location.

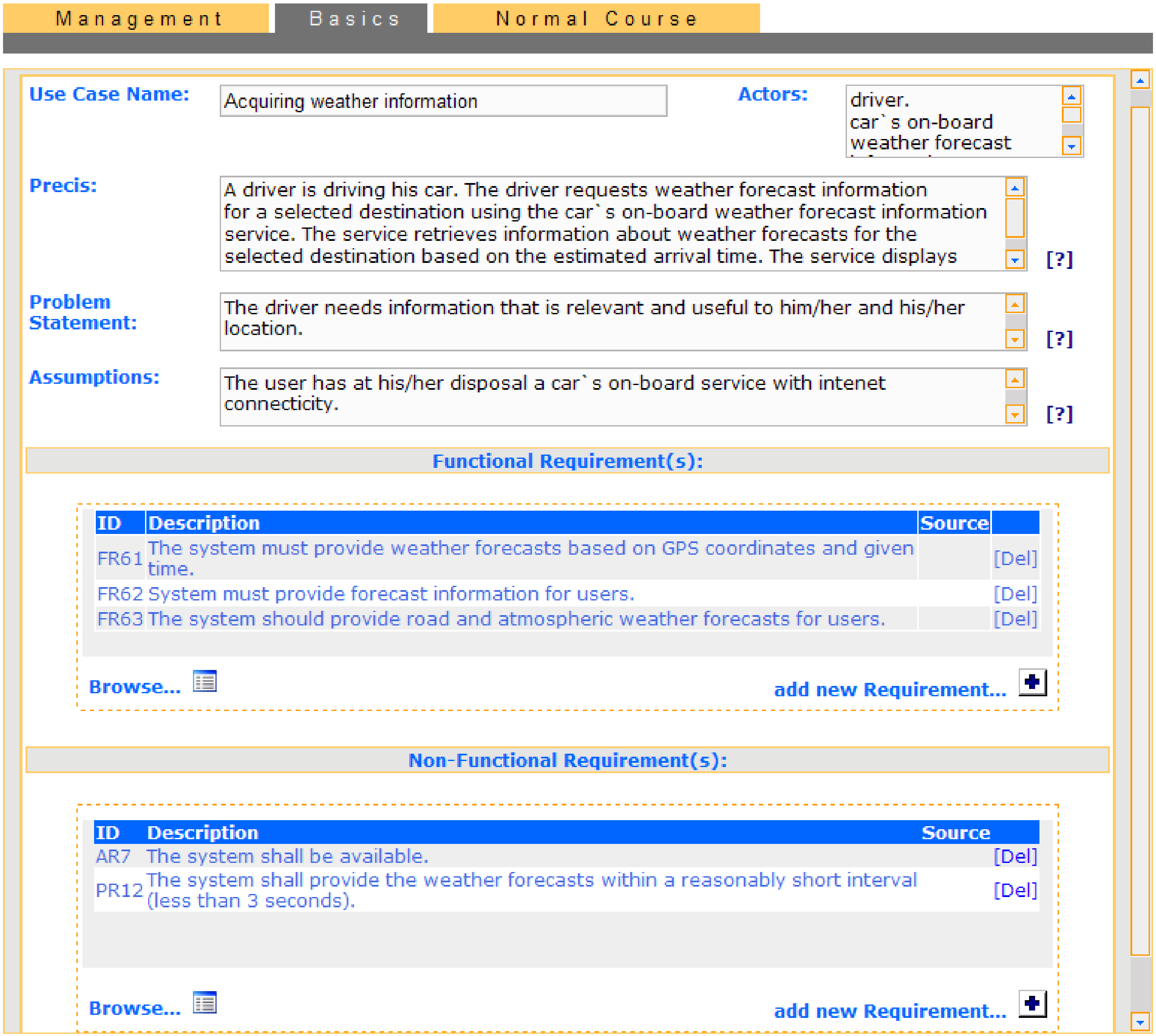
TABLE II. REQUIREMENTS ON THE ON-BOARD WEATHER FORECAST INFORMATION SYSTEM

FR61: The system shall provide weather forecasts based on GPS coordinates and estimated arrival time.
AR7: The system must be available remotely in order to be integrated into the in-car device.

B. The Service Registries

The SeCSE environment discovers web services specified within registries that link to service implementations that applications invoke. However, registries such as UDDI (Universal Description, Discovery and Integration) are inadequate for retrieving services using semantic criteria such as QoS and exception handling. To counter this shortcoming SeCSE has defined an extensible *faceted* specification mechanism [9]. Facets are projections over service properties and serve to partition a service specification according to the properties that the service provider wishes to make public. There are currently 10 facet types that can be used to describe a service. Those that are relevant here are *signature*, which mimics the WSDL (Web Services Description Language) specification of service binding and signature information and is always needed, a *description* facet which provides a brief natural language service description designed to aid discovery and selection by SeCSE's discovery tools, and the *QoS* facet used to describe a service's QoS characteristics. Although we commonly assume that the service provider is responsible for providing service facets, some could be provided or authenticated by third parties. This applies particularly to the QoS facet where the information may be the service's QoS that is claimed by the service provider, or it may represent data on the service's actual QoS as monitored by a third party.

Service discovery in SeCSE uses the description and QoS facets to retrieve web services [[15],[17]]. The role of the QoS facet in service discovery is to refine selection of services discovered using the description facet. Retrieved service descriptions are presented to requirements analysts and service consumers to enable them to select the most appropriate services based upon their QoS requirements. A full description of the part of the SeCSE conceptual model that relates to QoS and the QoS facet is provided in [15].



Use Case Name: Acquiring weather information

Actors: driver.
car`s on-board
weather forecast

Precis: A driver is driving his car. The driver requests weather forecast information for a selected destination using the car`s on-board weather forecast information service. The service retrieves information about weather forecasts for the selected destination based on the estimated arrival time. The service displays

Problem Statement: The driver needs information that is relevant and useful to him/her and his/her location.

Assumptions: The user has at his/her disposal a car`s on-board service with intenet connecticity.

Functional Requirement(s):

ID	Description	Source	
FR61	The system must provide weather forecasts based on GPS coordinates and given time.		[Del]
FR62	System must provide forecast information for users.		[Del]
FR63	The system should provide road and atmospheric weather forecasts for users.		[Del]

Non-Functional Requirement(s):

ID	Description	Source	
AR7	The system shall be available.		[Del]
PR12	The system shall provide the weather forecasts within a reasonably short interval (less than 3 seconds).		[Del]

Figure 2. Example use case specification specified in UCaRE

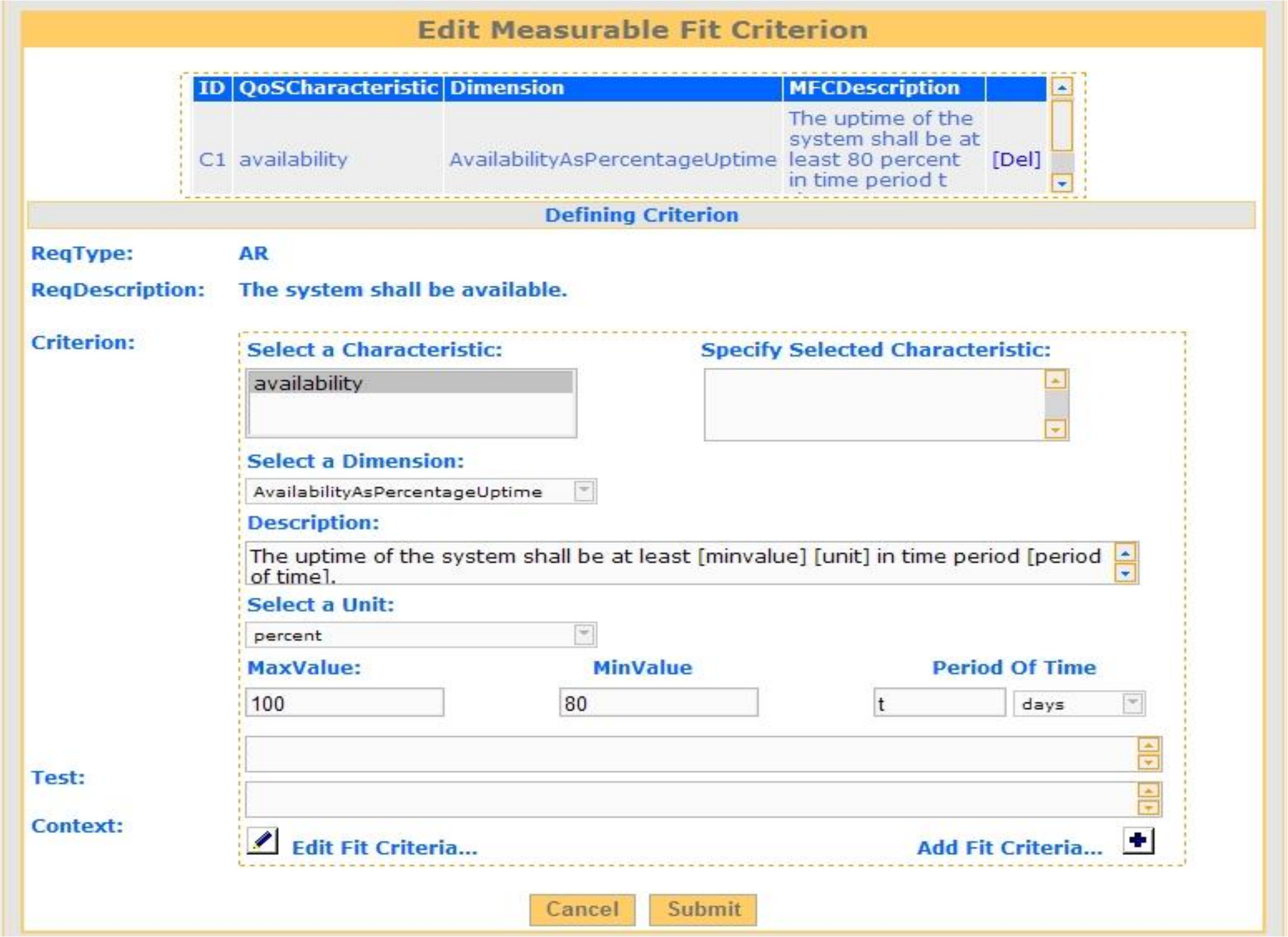
C. *The UCaRE Requirement Component*

Analysts express requirements for new applications using UCaRE [18], a web-based .NET application depicted in Fig. 2. A requirements analyst manages requirements and use cases through a web client.

At the start of SeCSE's requirements process, analysts work with future service consumers to develop simple use case précis that describe the required behaviour of the service-centric application. Table 1 includes the précis for the *Acquiring Weather Information* use case.

UCaRE supports the analyst in using the VOLERE shell to specify requirements such as AR7. VOLERE enables the analyst to specify the requirement's type, rationale, source, owner and importance scores. For non-functional requirements, UCaRE also supports the specification of measurable fit criteria (MFC) that are essential for selecting between discovered services on QoS criteria. MFCs are quantified goals that describe in detail how the system must behave in order to be deemed to have satisfied the requirement. While the description of the requirement is written in the language of the stakeholders, the MFC is written in a precise quantified manner so that solutions can be tested against the requirement. By stating requirements with measurable fit criteria that are aligned with the metrics of QoSOnt it is possible to use these to directly compare against the QoS facet. Fig. 3 shows the measurable fit criterion for the availability requirement AR7 that was used in the evaluation to filter the resulting list of candidate services relating to QoS.

The availability requirement in Fig. 3 shows that even if there was a broad agreement on the important characteristics of quality it is highly likely that a degree of translation would be required to allow different parties to compare QoS. For instance, even if exactly the same metric is used (say, mean time to repair) by both analyst and service provider it may be stated in any unit of time. Simple syntactic matching would lead to many false negatives in this case. Consider that the requirement AR7 of at least 80% availability (as percentage uptime over some period) has been specified for a weather service. Imagine that a specification of a candidate service includes a QoS property with a mean time between failure of 42 days and a mean time to repair of 1 hour. Without conversion capabilities the service's QoS property would not match the requirement and would give a false negative. However, with the integration of QoSOnt (section 2.5) mismatches of this kind are avoided. The metrics are first converted to compatible units (42 days = 1008 hours). The availability metric is then computed from these two component metrics 1008/(1 + 1008), allowing the inference that the availability of the service is 99.9%, thus that it meets the requirement.



Edit Measurable Fit Criterion

ID	QoSCharacteristic	Dimension	MFCDescription	
C1	availability	AvailabilityAsPercentageUptime	The uptime of the system shall be at least 80 percent in time period t	[Del]

Defining Criterion

ReqType: AR

ReqDescription: The system shall be available.

Criterion:

Select a Characteristic: availability

Specify Selected Characteristic:

Select a Dimension: AvailabilityAsPercentageUptime

Description: The uptime of the system shall be at least [minvalue] [unit] in time period [period of time].

Select a Unit: percent

MaxValue: 100

MinValue 80

Period Of Time t days

Test:

Context:

Figure 3. Measurable fit criterion for the availability requirement AR7

D. *Service Querying and Service Explorer Components*

In the SeCSE service discovery environment analysts can manipulate specified use cases and requirements to generate service queries. These are fired at service registries with the EDDiE service discovery engine [17] to retrieve web services from the same domain as the current problem. EDDiE implements advanced term disambiguation and query expansion algorithms to add different terms with similar meanings to the query using the WordNet online lexicon, thus increasing the number of web services retrieved from the registries.

The service discovery environment presents retrieved services to analysts and service consumers in the Service Explorer Component (as shown in Fig. 4). It orders services that attain a minimum threshold of relevance in a ranked order. The analyst can click on each service to view all properties of the service description facet and view each property next to its corresponding use case and requirement property to enable comparison. The Service Explorer component also provides functionality with which to select between retrieved services and refine the services available based on a service's QoS information.

E. QoSOnt as a means to filter and select candidate services

The SeCSE QoS Facet makes use of the QoS Ontology, QoSOnt [10] that implements the OMG model of QoS [12] using the OWL [11] ontology language. An ontology is a description of the concepts which exist in some domain and the relationships between them. QoSOnt does not just provide a set of terms but provides a machine interpretable model of QoS knowledge. The structure of QoSOnt, which is really three linked ontologies that model QoS properties, QoS metrics and units, is further described in [10].

Returning to our consideration of comparing the quality of candidate services during the requirements phase; determining whether a service supports certain metrics is of limited use without being able to compare the analyst's requirements against the services' capabilities. This is where the use of QoSOnt provides great advantages for this usage scenario.

It is through commitment to the same ontology (QoSOnt) in the specification tool, registry and discovery tool that comparison between service quality is made possible. UCaRE prompts the user to specify MFCs using terms directly from the ontology. This means that if the underlying ontology evolves then these changes will automatically be reflected in the tool. The SeCSE specification tool [9] uses the same approach, meaning that service QoS requirements are stated in compatible terms.

In practice this does mean that some level of agreement on a QoS vocabulary is a pre-requisite. However the use of an ontology easily allows the use of multiple synonyms and the expression of new concepts in terms of other existing concepts.

III. EVALUATION METHOD

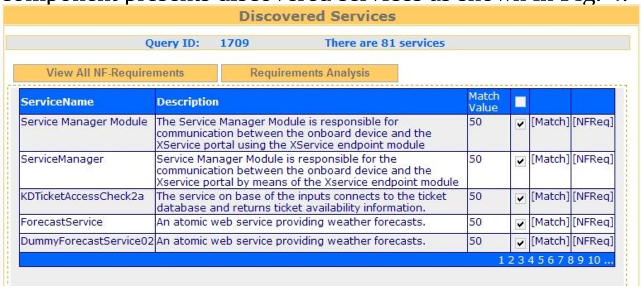
We made use of a scenario from SeCSE's industrial partners in the design of our experiment. For each service involved in the scenario the industrial partners specified and published a QoS facet and a description facet. They then described the corresponding use case using the UCaRE tool. With these artifacts in place it was possible to attempt requirements-based service discovery, i.e. matching the use case requirements against the service descriptions in the SeCSE registry.

In order to evaluate how useful this was we needed more data however. In particular, in order to assess whether the conversion capabilities provided by QoSOnt were allowing a greater number of candidate services to be compared, we needed more examples of QoS facets. At the same time, in order to be evaluated using UCaRE the description facet of the corresponding services had to remain largely the same. To achieve this we took the specifications of industrial partners and created dummy service specifications in which the description was unaltered, but the QoS facet was varied. In the dummy services the units of various metrics were altered to compatible units assuming a uniform random distribution. Where another way of expressing metrics was known, metrics were also re-expressed randomly in the dummy versions of the services. Having populated the registry with dummy services for each original service we then re-ran the discovery and QoS-based filtering process in UCaRE.

We focused the experiment on the QoS availability characteristic. Availability is a crucial service property for many applications, and the scenarios provided by our industrial partners reflected this. Further, there are several metrics that can be used to express availability and availability can be considered a compound of some metrics that actually represent other QoS characteristics. Finally, different metrics can use different units. As we described earlier, these factors all meant that a scenario in which availability was a crucial QoS characteristic had the potential to provide a test of our hypothesis.

A. Requirements-based Test Queries

A service query was formed from specification elements for the Acquiring Weather Information use case, including the availability requirement AR7 that provided the basis of the service query evaluation. The service query was then used to discover services in the SeCSE registry using EDDiE. Once generated and fired at the SeCSE service registry to retrieve similar web services, the Service Explorer component presents discovered services as shown in Fig. 4.



Query ID: 1709 There are 81 services

ServiceName	Description	Match Value	□		
Service Manager Module	The Service Manager Module is responsible for communication between the onboard device and the XService portal using the XService endpoint module	50	☑	[Match]	[NFReq]
ServiceManager	Service Manager Module is responsible for the communication between the onboard device and the Xservice portal by means of the Xservice endpoint module	50	☑	[Match]	[NFReq]
KDTicketAccessCheck2a	The service on base of the inputs connects to the ticket database and returns ticket availability information.	50	☑	[Match]	[NFReq]
ForecastService	An atomic web service providing weather forecasts.	50	☑	[Match]	[NFReq]
DummyForecastService02	An atomic web service providing weather forecasts.	50	☑	[Match]	[NFReq]

Figure 4. Discovered service descriptions shown in the Service Explorer

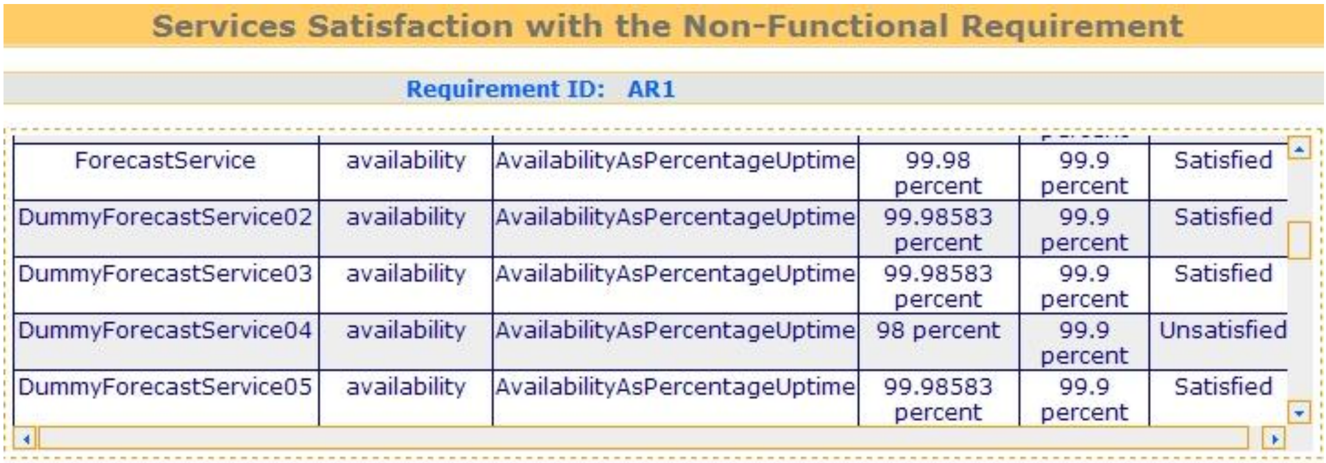
The Service Explorer orders services that attain a minimum threshold of relevance in a ranked order. Each service can be selected to view all properties of the service description facet and view each property next to its corresponding use case and requirement property to enable comparison.

The Service Explorer component also provides functionality with which to select between retrieved services and refine the services available based on a service's QoS information. Fig. 5 shows part of the list of services that have been evaluated based on the metrics. Satisfied indicates that the availability of a service is at least 99.9% and thus that it meets the relevant measurable fit criterion. On the other hand, Unsatisfied indicates that the availability of a service is below 99.9% and thus that it does not meet the relevant measurable fit criterion.

B. The QoS Specifications used in the SeCSE Registry

To evaluate whether QoSOnt can provide benefits to the discovery process, we seeded the SeCSE repository with 14 dummy specifications. These represented clones of the real weather forecast service, ForecastService, which acted as the baseline. Each clone specified its Availability QoS properties

ISBN: 978-1-61208-105-2



Services Satisfaction with the Non-Functional Requirement

Requirement ID: AR1

ForecastService	availability	AvailabilityAsPercentageUptime	99.98 percent	99.9 percent	Satisfied
DummyForecastService02	availability	AvailabilityAsPercentageUptime	99.98583 percent	99.9 percent	Satisfied
DummyForecastService03	availability	AvailabilityAsPercentageUptime	99.98583 percent	99.9 percent	Satisfied
DummyForecastService04	availability	AvailabilityAsPercentageUptime	98 percent	99.9 percent	Unsatisfied
DummyForecastService05	availability	AvailabilityAsPercentageUptime	99.98583 percent	99.9 percent	Satisfied

Figure 5. Discovered services evaluated based on QoS

differently from the baseline. As summarised in the following table, in some cases this involved the use of different units, in others the use of different metrics, i.e. percentage uptime, Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR).

Seven of the 14 dummy specifications used different QoS metrics to the baseline specification. Instead of directly specifying an Availability metric, the MTBF and MTTR properties were specified. QoSOnt can infer an availability measurement by combining other metrics, and the purpose of these seven specifications was to evaluate this ability.

Seven of the dummy specifications also used different units in order to specify the metrics. QoSOnt is able to convert between different units (for example, hours and minutes), and these specifications would help evaluate this. The baseline and dummy specifications are summarized in Table 3 below.

IV. Evaluation Results

Table 3 shows the expected outcome of performing a QoSOnt supported discovery activity on these specifications, based on the use case described in section 2.1 (99.9% availability). Table 3 also shows the converted availability value that QoSOnt should infer for the specifications that do not directly specify their availability. The final two columns show the expected and actual result from applying QoSOnt and the service discovery tools to the registry containing the set of service descriptions.

As shown in Table 3, by using the conversion functionality provided within QoSOnt UCaRE/EDDiE was able to determine a %ageUptime metric for 13 of the 15 Weather Forecast services. Seven of these services were found to satisfy the required Availability measurement as described within the use case. Dummy services 2, 5, 6, 7, 9, 11 and 12 all used different metrics and units, and QoSOnt successfully performed conversions to infer availability values for them. Dummy services 8 and 10 were not returned. In the case of number 8 this was because it only contained one metric, MTBFs, and this on its own was not enough to infer an availability value (MTTR being required as well). Dummy service 10, on the other hand, contained no metrics that could be used to derive an availability value.

We re-ran the experiment without support from QoSOnt. Here, only syntactic matching was possible and only 5 results were obtained. All 5 were for service descriptions that used the percentage uptime metric: the baseline service and dummy services 4, 13, 14 and 15. The other services that satisfied the requirement but expressed using a different metric (2, 3, 5 and 6) were not found. Overall, in this experiment, by not using QoSOnt only 43% of the service specifications that would actually satisfy the use case were discovered.

V. Hypothesis Revisited

We used results and data from the evaluation to demonstrate QoSOnt's ability to convert between metrics and units for any (QoS characteristic, metric, unit) tuple that is known to it. Hence, we have provided evidence to support our hypothesis, that *the SeCSE service specification and SeCSE's requirements-based discovery tools are tolerant of mismatches between how a service provider specifies their services' QoS properties, and how a service consumer specifies their QoS requirements*. Of course, this is only true where both service consumer and service provider use characteristics, metrics and units known to the ontology. However, integration of QoSOnt in SeCSE's service specification and service discovery tools make consistency easy to achieve in a way that is transparent to both actors. It is this transparency that is crucial if the vision of an open service marketplace, with all the diversity of practice that implies, is ever to be realised.

Clearly there are threats to results validity. One threat to the conclusion validity of the evaluation results is the sample size – 1 service query from 1 use case specification with 1 quality type fired at 1 registry. However the current small body of research into matching non-functional requirements to service qualities during service selection led us to run a formative-predictive evaluation to generate a first set of results to provide a framework and focus for more subsequent rigorous evaluation.

VI. CONCLUSIONS

In this paper we have described an empirical evaluation that assessed the feasibility of an approach to filtering candidate services based upon QoS. The approach in question relied upon the use of the QoSOnt ontology both in the QoS facets used to specify services and in the measurable fit criteria stated in service requirements. We have demonstrated that commitment to a common ontology did aid in achieving the desired QoS-based filtering. We have also begun to demonstrate the wider advantages of the use of such ontology. In particular the supporting conversion rules were found to avoid a number of cases, which would have produced false negative matches.

Our experiment showed that the number of comparisons, which were made possible only with conversion capabilities, could potentially be significant. In practice it is not clear how the use of different metrics and units would vary across the population, but in the absence of such knowledge we believe that our experiment gives us a first approximation of the effects of conversion in service selection.

Although more direct evaluations are needed to answer questions like ‘Do the QoS facet and QoSOnt assist QoS-critical service discovery?’ – evaluations that are planned to be undertaken in the near future – the results presented in this paper support our hypothesis that the SeCSE specification and requirements-based service discovery mechanisms are tolerant of mismatches between how a service provider specifies their services' QoS properties, and how a service consumer specifies their QoS requirements. By implementing the OMG model of QoS [12] in QoSOnt we have tried to ensure coverage of recognised QoS properties. New QoS properties may emerge that have particular utility for service-centric systems. QoSOnt will need to evolve as this happens and has been released as open source to help encourage a user community to play its role in maintaining its relevance to service-centric systems engineering.

REFERENCES

[1] I. Sommerville. “Service-Oriented Engineering” in Software Engineering, 8th Edition, Pearson Education, 2006, 743-770.

[2] J. Bloomburg, “Competitive SOA”, ZapFlash, 2007.

[3] SeCSE, “Enriched mixed scenario for final demonstration” Deliverable A6.D18, http://www.secse-project.eu/?page_id=102&page=3, (last accessed 10/11/09)

[4] EU Integrated Project 511680, “Service Centric System Engineering (SeCSE)”, http://www.secse-project.eu/, (last accessed 21/07/10)

[5] S.V. Jones, N.A.M Maiden, K. Zachos, and X. Zhu, “How Service-Centric Systems Change the Requirements Process”, Proc. REFSQ’2005 Workshop, 2005, pp.13-14

[6] G. Fischer, S. Henninger, and D. Redmiles, ‘Intertwining Query Construction and Relevance Evaluation’, Proc. CHI’91, 1991, pp. 55-62.

[7] S. Robertson and J. Robertson, Mastering the Requirements Process, Addison-Wesley-Longman, 1999.

[8] I. Jacobson, G. Booch, and J. Rumbaugh, ‘Unified Software Development Process’, Addison-Wesley-Longman, 2000.

[9] J. Walkerdine, J., Hutchinson, P. Sawyer, G. Dobson, and V. Onditi, "A Faceted Approach to Service Specification", Proc. 2nd Int’l Conf. on ICIW, Mauritius, 2007.

[10] G. Dobson, R. Lock, and I. Sommerville, “Quality of Service Requirements Specification using an Ontology” Proc. SOCCER at 13th Int’l RE Conf. (RE 05), 2005.

[11] D. L. McGuinness and F. van Harmelen (eds.) “OWL Web Ontology Language Overview”, W3C Recommendation, http://www.w3.org/TR/owl-features/, 2004, (last accessed 05/09/09)

[12] “UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics & Mechanisms”, OMG, 2004.

[13] G. Dobson and P. Sawyer: "Revisiting Ontology-Based Requirements Engineering in the Age of the Semantic Web", at Dependable RE of Computerised Systems, NPPs, 2006.

[14] K. Zachos and N.A.M. Maiden, ‘Inventing Requirements from Software: An Empirical Investigation with Web Services’, in Proc. 16th International Conference on RE, 2008.

[15] K. Zachos, G. Dobson, and P. Sawyer, 'Ontology-aided Translation in the Comparison of Candidate Service Quality', in Proceedings of SOCCER workshop at RE08, 2008.

[16] K. Zachos, N.A.M. Maiden, and R. Howells-Morris, ‘Discovering Web Services to Improve Requirements Specifications: Does It Help?’, in REFSQ, June 2008.

[17] K. Zachos, N.A.M Maiden, S.Jones, and X.Zhu, “Discovering Web Services To Specify More Complete System Requirements'” Proc. 19th Conference on CAiSE, 2007.

[18] K. Zachos, X. Zhu, N.A.M. Maiden, and S. Jones, “Seamlessly integrating service discovery into UML requirements processes” Proc. SOSE '06, 2006, 60-66.

TABLE III. EXPECTED AND ACTUAL OUTCOMES OF THE DISCOVERY ACTIVITY

ServiceName	**Specified Metrics**	**Specified Values**	**%age uptime**	**99.9% uptime?**	
				Expected	**Expected**
ForecastService (baseline)	%age uptime	99.98 %	*99.98 %*	Yes	Yes
Dummy02	MTTR, MTBF	1 Hour, 42 weeks	99.98 %	Yes	Yes
Dummy03	MTTR, MTBF	60 Minutes, 294 Days	99.98%	Yes	Yes
Dummy04	%age uptime	98 %	*98 %*	No	No
Dummy05	MTTR, MTBF	3600 Seconds, 42 Weeks	99.98 %	Yes	Yes
Dummy06	MTTR, MTBF	58 Minutes, 42 Days	99.90%	Yes	Yes
Dummy07	MTTR, MTBF	100000 Seconds, 1 Month	99.62 %	No	No
Dummy08	MTBF	30 Days	-	No	-
Dummy09	MTTR, MTBF	4 Hours, 28 Days	99.40 %	No	No
Dummy10	None specified	-	-	No	-
Dummy11	MTTR, MTBF	1 Week, 2 Months	89.68 %	No	No
Dummy12	MTTR, MTBF	10 Days, 1 Year	97.33%	No	No
Dummy13	%age uptime	99 %	*99 %*	No	No
Dummy14	%age uptime	99.999 %	*99.999 %*	Yes	Yes
Dummy15	%age uptime	99.9 %	*99.9 %*	Yes	Yes

Implementation of the Information System of the Telecom Operators Using the ITIL V3 Methodology for the Service Design Phase

Anel Tanovic
Department for IT development of multimedia services
BH Telecom d.o.o. Sarajevo
Sarajevo, Bosnia and Herzegovina
anel.tanovic@bhtelecom.ba

Fahrudin Orucevic
Department of Computer Science and Informatics
University of Sarajevo, Faculty of Electrical Engineering
Sarajevo, Bosnia and Herzegovina
forucevic@etf.unsa.ba

Abstract - **In order for any business organization to perform on a successful level, it is necessary that it has a defined set of activities that manage IT processes and services. Defined set of activities constitutes the Service Management of that organization. There are many types of Service Management, and this paper is based on the type of which is called Information Technology Infrastracture Library V3 (ITIL V3). ITIL V3 has a total of five phases: Strategy Service, Service Design, Service Transition, Service Operation and Continual Service Improvement. This paper is focused on the description of its second phase, which is called Service Design. The aim of this paper is to describe information systems development of the Telecom Operator for Service Design phase of the ITIL v3 methodology in all seven of the processes that are contained in these phases. These are: Service Catalogue Management, Service Level Management, Capacity Management, Availability Management, IT Service Continuity Management, Information Security Management and Supplier Management. This paper attempts to show to the telecom operators that during the realization of their IT processes they do not necessarily have to use the eTOM standard, but may use other standards and concretely they can use ITIL V3 standard. This paper in its conclusion, with previously detailed analysis of the implementation of the Service Design processes according to ITIL V3 methodology in the information system of the telecom operator, needs to define which teams are required to implement the Service Design processes and which are the activities that each team has during the implementation of the information system of the telecom operators.**

Keywords-ITSM, ITIL V3, Service Design, Service Catalogue Management, Service Level Management, Capacity Management, Availability Management, IT Service Continuity Management, Information Security Management, Supplier Management

I. INTRODUCTION

Any organization that wants to increase the level of their business depends on the way of the IT Service Management. In order to make IT processes and services successfully conducted, it is necessary that the organization defines a set of specialized organizational capabilities that are provided to customers in the form of the service. This set of specialized skills consists the Service Management organization. There are many standards for Service Management practices such as ITIL, COBIT, CMMI, eTOM, PRINCE2 and ISO 20000 [6].

ITIL is an acronym for "Information Technology Infrastructure Library," and falls under ITSM, which is an acronym for "Information Technology Service Management". ITSM is defined as a discipline for managing IT systems, which is centered on the customer perspective of IT's contribution to the business. ITIL is the best practice environment for companies that provide IT services as its core business functions. ITIL takes a "life" approach for the development of IT services and produces the best practices and guidelines throughout each "life" stage to help IT companies to deliver the best possible services to their customers. ITIL V3 defines five main phases of life cycle [1], [2], [3], [4], [5]:

1. Service Strategy
2. Service Design
3. Service Transition
4. Service Operation
5. Continual Service Improvement

1. Service Strategy provides guidance on how to design, develop and implement Service Management, not only as an organizational capability but also as a strategic asset. These guidelines are based on the principles that underpin the practice of Service Management, which are useful for the development of Service Management policies, directives and processes through ITIL lifecycle development services [1]. Service Strategy includes the following five processes: Strategy Generation, Risk Management, Demand Management, Financial Management and Service Portfolio Management.

2. Service Design is the second phase of the development lifecycle of IT services for ITIL v3 methodology and an important element within the business processes aimed at

ISBN: 978-1-61208-105-2

change. The role of Service Design can be defined as developing appropriate and innovative IT services, including their architecture, processes, policies and documentation to meet current and future agreed business requirements [2]. Service Design has the following seven processes: Service Catalogue Management, Service Level Management, Capacity Management, Availability Management, IT Service Continuity Management, Information Security Management and Supplier Management.

3. Service Transition phase has a role to let the operational use of the services that have been designed in the Service Design phase. Service Transition performs the same thing in a manner that it receives the Service Design Package and transferred to the Service Operation phase of each element required for operations that are executed [3]. If the business assumptions, forecasts or request change from the Service Design phase, then they may need modification during the Service Transition stage in order to deliver necessary IT service. Service Transition has a total of six basic processes: Change Management, Service Asset and Configuration Management, Knowledge Management, Transition Planning and Support, Release and Deployment Management and Service Validation and Testion.

4. Service Operation phase has a role to ensure that the user is provided with the agreed level of quality service that is necessary for this phase to be able to manage applications, technologies and infrastructure that supports the implementation of services. The goal of Service Operation is the coordination and execution of activities and processes that are necessary for enablement and management of the services at agreed levels [4]. Service Operation has a total of five processes: Event Management, Incident Management, Problem Management, Request Fullfilment and Access Management.

5. Continual Service Improvement is the last fifth stage of the development lifecycle of the IT services for the ITIL V3 methodology, which is responsible for continuously changing business needs by improving business needs by improving the effectiveness and efficiency of the process [5]. Duration of the Continual Service Improvement depends upon the size of the organization and takes several weeks to several months. The result of Continual Service Improvement is reflected through improved IT services throughout all of the first four phases of the ITIL V3 lifecycle.

Section II begins with a description of the importance of building a unified Telecom Operator information system with the aim to improve its business, and continues with a concrete description of the implementation of all processes of the Service Design phase of the information system of the Telecom Operator. Conclusion of the work, which originated as a result of the analysis carried out in Section II, proposes teams for implementation of all Service Design phase processes, the time period for action of each team and the guidance in which direction should further research in this field be aimed.

II. IMPLEMENTATION OF SERVICE DESIGN PROCESSES IN INFORMATION SYSTEM TELECOM OPERATOR

In condition of competitive environment and objective geopolitical circumstances, one of the basic tools for the successful positioning of telecom operators in area of some region is to construction of the system that will enable rapid implementation of the services and flexible tariff plan [10], [12]. The realization of such a system is aimed at providing conditions for the service users to be able to request in an identical manner and only in one step through multiple channels of access (Telecom center or web) each individual service, or any combination of available services in any part of the area in which operates Telecom operator and to receive personalized account for these services. Information System of the Telecom Operator, whose development for the Service Design phase of the ITIL V3 methodology is described in this document, should consolidate all of the databases for the Telecom operator's technologies: fixed telephony, mobile telephony and Internet services and their billing systems into a single database and a unique billing system. Development of the information system is based on development of the two key applications: application of the user data and sales applications that use Oracle database. Construction of the entire system involves the definition of business requirements with the creation of infrastructure design, logical and physical database design with the specification of the target business processes and functionality of the new applications, development and functional testing of the system with the development of applications and scripts for data migration and implementation of systems with migration of data into a new system and performing tests for checking the functionality of the new system.

General functionalities that need to be supported by the information system are:

- Administration of users, data entry and data management of all user-oriented activities.
- Support for complex hierarchical structure of the customer.
- Interaction with systems in the environment.
- Recording of the complaints and grievances.
- Generation of reports on service users.
- Sales Management with the complete supervision of the transactions execution.
- Dynamic generation of reports, documentation, and monitoring the sales process.
- Provision of the efficient distribution channels.

- Enablement of the review of the customer informations regarding subscription packages, included products, equipment and services.
- The execution of the integration with other applications.

A. Service Catalogue Management

The definition of a unique catalogue of IT services is affected by: products, product catalogue, product offer and price of the products whose value is affected by the first three mentioned parameters (Figure 1).

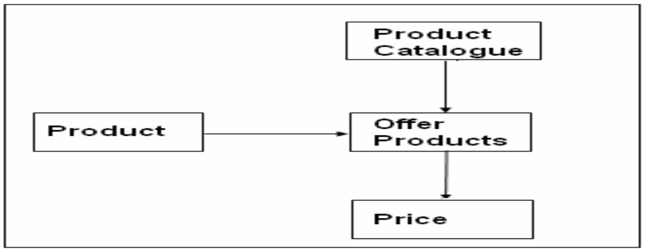


Figure 1. The parameters that influence the definition of IT service catalogue

During the construction of information system what should be taken into consideration is to build several different types of product catalogues, depending on the different categories of end users [8], [26]. It is necessary to define a total of seven product catalogues depending on the different categories of the users:

1. Product Catalogue for the business users
2. Product Catalogue for the residential customers
3. Product Catalogue for the disabled
4. Product Catalogue for the war veterans
5. Product Catalogue for the students
6. Product Catalogue for people with special needs
7. Joint product catalogue

Each of the displayed product catalogues should provide:

- Membership of the certain offer to a particular specification. Specification represents a group of a services with similar characteristics, which are sold and implemented in a similar manner.
- The associated products that define the way of payment and price.
- Link to a specific sales channel.
- Link to certain types of network elements.
- Connection to distribution channels.
- Connection to marketing promotions.
- Description of the data that must be entered for each of the products sold in conjunction with validating the rules.

External factors are affecting definition of the final structure of the product that is inserted into the product catalogue, and these factors are: a view of the end user, the marketing impact, the impact of the accounts by the end user, the financial impact and payment structure of the product catalogue [6], [26] (Figure 2).

Price list of the information system should primarily include: the price of connecting practice, a subscription, the cost of monthly fees and the basic price of the device. It may or may not contain the costs of the service which is charged by usage and prices that are charged by duration or quantity. Catalogue of the IT services should enable discounts upon several different categories:

- Discounts on the basis of action offer
- Discounts for the duration of the contract
- Discounts for the length of the subscription service
- Discounts for each user
- Discounts for the number of connections
- Discounts on special conditions

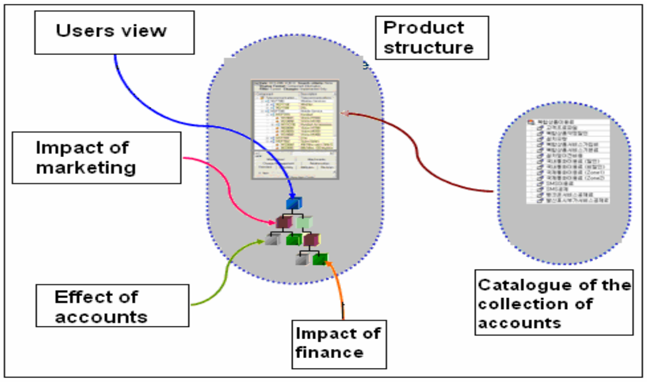


Figure 2. The parameters that influence on definition of the final structure of the product

B. Service Level Management

In the analysis of Service Level Management three types of agreements are taken into account: Service Level Agreements (agreements with end users), Operation Level Agreements (agreements within organizational units of the firm) and Contracts (agreements with the partner company that participates in the implementation of a project) [19]. All three agreements must ensure that service levels are aligned with business expectations.

Construction of agreements with customers (Service Level Agreements) involves the construction of a single document that represents a description of the mutual rights and obligations that need to be followed by the Telecom operators and their end-users when two sides agree to sign a contract on the use of a Telecom service operators by the end user. The document itself should contain seven items:

1. Defining the price of services that is defined by the Telecom operator through the catalogues of IT services and the payment method that requires the end user.

2. Defining the implementation of the services in which the Telecom operator needs to commit to a service realization time starting from the date on which end-user submitted a request to fulfill services.
3. Defining the terms on which Telecom operator can't implement the request for a service made by the end user because there is no technical possibility for the realization of the service.
4. Defining the termination of the contract between Telecom operators and end-user if the user does not pay the debt in cash within a defined time, is not available at the registered address or if there is reasonable suspicion that the user intends to misuse or abuse the service that is leased.
5. Defining the deadline for returning the terminal equipment by the end user if he withdraws from service or his service is turned off for legitimate reasons.
6. Defining the timeframe for abandonment of the implementation of services by the end user.
7. Defining the responsibilities of Telecom operator for any damage inflicted upon a user or third party due to misuse of user data from a person that performs before mentioned responsibility.

Required elements in the user data application for full implementation of the agreements with customers are:

- Provide monitoring of the data about all users of technology for Telecom operator to look at the details of user data from all billings and all of the technologies, as well as for creating a billing group according to user needs.
- Develop a clear hierarchy of users with the possibility to record the different roles that physical or business person may have, as well as defining the connections and relations between different users.
- Provide a flexible system of allocation of the discounts and affordable packages throughout segmentation of the users.
- Give the possibility to create different user profiles and assign user profiles defined for the purposes of reporting based upon user profiles.
- Support monitoring and recording of objections, claims and complaints of users, as well as creating and monitoring their implementation stratus.
- Define mechanisms for the integration of various technologies users into a single application user data.

Agreements between different organizational units (Operation Level Agreements) imply an agreement described in the document on cooperation in building a unified information system. The document must be a deal agreed between all organizational units of the Telecom operators should commit to mutually exchange data and information when building information systems. Telecom operator's organizational units that should be included in the building of a new information system are: Division of fixed telephony, mobile telephony department, department of private Internet users, department of business Internet users, call center operator Telecom, Telecom Center and Department VPN.

Agreements with foreign company - consultant (Contracts) are achieved only after a public process of choosing IT consultant who will help the telecom operator in the implementation of information systems [23]. The choice of IT consultants is done in two phases: phase prequalification stage and calls for submission of bids. Required steps for the candidates to pass the prequalification status are:

a) If there is no legal impediment to their participation in the competition for the selection of the best IT consultant.
b) To be entitled to pursue professional activities and is registered in appropriate professional registers.
c) To have economic and financial condition to realize a successful implementation of the contract.
d) That their technical and professional capacity guarantees the successful implementation of the contract.

As for the technical and professional skills, candidates should meet the following minimum requirements:

a) Employees at least 20 certified IT consultants.
b) At least one reference to the information system implementation project in the telecom industry.
c) Employeed at least 10 certified IT consultants who will be constantly involved in the project and who have participated in at least 3 projects of implementation of information systems in the industry.
d) Owning your own HW/SW infrastructure necessary to start the project (testing and development) and other necessary technical preconditions.

The second phase is the phase of invitations to tender where it is necessary to establish criteria for evaluating bids. There are three basic parameters by which the IT consultant is selected from all of the candidates: the lowest price (with percentage share of 60%), the quality of the offered solutions (with percentage share of 25%) and a time deadline for delivery of the offered solutions (with percentage share of 15%). The contract will be assigned to the IT consultant who submitted the top rated acceptable bid.

Telecom operators may, after completion of the first phase, and before the second phase, stop the process of selecting IT consultant for one of the following reasons:

a) No bids have been submitted within a certain deadline.
b) None of the bids received is not technically acceptable.
c) Prices of all eligible bids were significantly higher than the budget for the contract signing with the IT consultant.
d) Number of acceptable offers is less than 3 and does not ensure genuine competition.
e) Number of qualified candidates is less than 3 and does not ensure genuine competition for the actual contract.

C. *Capacity Management*

In order to verify system performance and architecture on which the system is based upon, during the design of information systems it is required to perform tests of system performance and stress tests [15], [20]. The aim of these tests is:

- Prove that the system is able to handle the expected load.
- Prove that the system is able to withstand the excessive load.
- Find errors in the architecture of the system that cause oversized stress.
- Propose a strategy to optimize the system.

For the purpose of measuring performance it is necessary to use the following three tools:

- Enterprise Service Bus (ESB) Console – used to measure the performance of business applications and communication between them.
- Business Process Execution Language (BPEL) Console - used for definition of web service compositions.
- Windows Task Manager – should be used on servers in order to verify the efficiency of the CPU.

For each technology, telecom operators should perform the following five tests:

1. Functional test - Only one request is sent to the subsystem of the technologies in order to prove the functionality and initialization of internal structures.
2. Scenario with an expected average load - simulating the expected load on the system in order to prove that the system is stable at this load.
3. Scenario with the expected excess load - Excessive load on the system is simulated in order to prove that the system is stable at this load.
4. Extreme scenario (shock) loads - Excessive workload increases by 10 times to prove that the system can withstand this pressure.
5. Scenario long test - This scenario has the same structure as the scenario with an expected average load, only taking into account that the time of the load is increased.

There are two general conclusions that should result from the above tests that are done before the system itself is released into production:

1. Information system is ready to withstand the expected load level with respect to capacities that are currently available.
2. Information system is ready to withstand the excessive load level if it faces in the future the expanded capacity which currently has system itself.

Systems analysts should be taking into account the results of the testing and with knowledge of the system architecture propose four recommendations that will enable the system stability when it comes down to it’s current and future capacity:

1. Use direct calls between Java Enterprise Service Bus (ESB) and Business Process Execution Language (BPEL).
2. Reduce the level of logging in a production environment.
3. Reduce Business Process Execution Language (BPEL) control level in a production environment.
4. Perform regular cleaning of the log files from the ESB and BPEL after measurements of system performance.

Giving the above recommendations requires the creation of the links of the Capacity Management with Demand Management for the formation of the necessary requirements for the implementation of the above proposals. So the link to the Demand Management influences on the formation of the budget.

D. *Availability Management*

The new information system of the Telecom operator should be available and functional 24 hours a day during 365 days year. The reason for this is the Billing system and web portal that always must be functional. This means that the only possible value of the availability of the system is the maximum one (100%). For this purpose a stable system architecture needs to be built, design accessibility plan and and form a group of IT professionals that will release the information system in production, constantly update the program status because of the eventual changes in the architecture of the system.

The plan includes access to all components of the architecture of information system that must always be functional [16], [24], [25]. Components of the architecture of the system are summarized in Figure 3. Note that the server side architecture are web services, Java services, Enterprise

ISBN: 978-1-61208-105-2

Resource Planning (ERP), Oracle databases, and SAP; the user updates the database. The client-side architecture are web, application user data and application sales. The connection between server and client side architecture is realized through the Business Process Execution Language (BPEL), which is primarily used for definition of web service compositions.

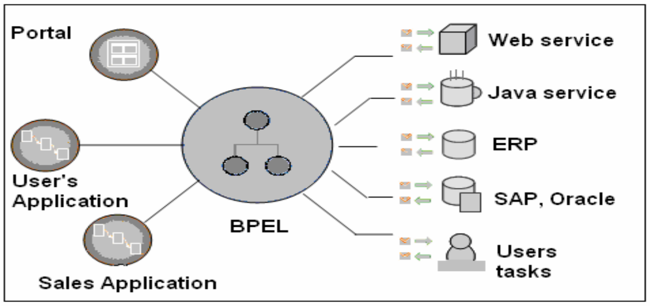


Figure 3. Components of the architecture of the Information system which are involved in the Availability plan

Availability plan consists of the following items:

- Defining the interoperability of systems - Define the integration of heterogeneous systems purchased from different suppliers based on different technologies into a single system.
- Defining the flexibility of the system - Define the integration of new applications into existing system structures. The whole logic of the integration of the information systems should be contained in one place and should not be placed in applications because it has to be defined in XML documents (BPEL processes). Business processes and information system should be flexible and tolerant to the changes in the integrated applications and systems. Integration platform should be such to provide a central point that the applications are associated to.
- Defining the independence of the data structure - Data within the system should be such that they are exchanged in form of XML documents that can be transformed so that the independence of the data structure is achieved.
- The definition of the integration based on events - BPEL processes should provide almost instant exchange of data and events between the integrated applications.
- Defining the performance and reliability - The platform should support a web server for fast execution of the program so that the BPEL Process Engine running on Oracle application server provides high availability and scalability of the application servers.
- Defining user tasks - Oracle BPEL Process Manager needs to support the creation of workflows and integrate business processes with the BPEL processes. Standard graphical user interface for defining user tasks that can be partially or completely changed should be provided that way.
- Definition of system maintenance - Plan should include the availability of an algorithm for solving the problem. Every problem that impairs the availability of services, within a certain period of time.

If a problem results in termination of the entire system, the problem must be solved 24 hours from the hour of the occurrence problem. If the problem represents a decline in the primary service, then this problem must be solved 48 hours from the hour of creation. If the problem represents decline in the secondary service, then this problem must be solved six days from the date of creation. If it is a minor problem that caused the fall of a service, then this problem should be settled within 7 days from the date of the problem occurence.

Each time delay regarding solution of these problems violates the above maximum systems availability. Every problem that has been successfully resolved in the timeframe described above maintains maximum availability of IT system. The Telecom partner and firm partner are responsible to solve the above-mentioned types of problems. Each one of the defined items of the accessibility plan must be updated regularly by a group of IT professionals.

E. IT Service Continuity Management

The basic step in the implementation of IT Service Continuity Management in Information System of the Telecom operator is to build the Business Continuity Plan [6]. Business Continuity Plan in information system contains the steps that need to be implemented when one of the services stops working in order to achieve faster construction of a new service that will continue to operate on the same level it operated before the old service failure [2]. It consists of four phases: initialization, definition of requirements and strategies, implementation and preparing for the production.

For the software engineering team to start the implementation of the Business Continuity Plan at all, it is necessary that reports reagarding termination of an old service's funcioning are delivered to them. To come to the team for a software engineering that is responsible for implementiation of the Business Continuity Plan, it is necessary to pass through the four lower levels of support. The first level of support is reporting a problem, another level of support is to locate the problem, the third level of support does support the site, which is responsible for normal technical requirements, and the last fourth level of support is a technical support. Of course, the implementation of the Business Continuity Plan will come if any of the first four levels of support fails to recover the old service. Table I. shows the Business Continuity Plan for the information system of Telecom operator.

TABLE I. BUSINESS CONTINUITY PLAN FOR INFORMATION SYSTEM OF TELECOM OPERATOR

Name of Business Continuity Plan phase	Name of the under phases of Business Continuity Plan
I. Initialisation	1. Document drafting job descriptions
	2. Detailed specifications for the functionality application development solutions
	3. Specification of target business processes and functionality of new application modules
	4. Detection and specification of services and orchestration integration flows
II.Definition requirements and strategy	1. Design data models and messages
	2. Design integration architecture
	3. Development of test strategy
	4. Preparation of development platform
	5. Implementation of database
	6. The development, implementation and Functional test of web service
	7. The formation of the final documentation
III.Implementation	1. Documentation of system
	2. Preparation of test platform
	3. Functional tests
	4. Integration tests
	5. End-2-End tests
	6. Creation production platform
	7. The formation of final documentation
IV. Preparation for production	1. Acceptance tests
	2. Training end users
	3. Moving into production

F. *Information Security Management*

Security in applications of information systems [25] of the Telecom operators are implemented on the basis of 6 levels where the most important role is played by a database administrator, application server administrators and end users:

- The level of database
- The level of web services
- The level of application server
- The level of application
- The level of URL
- The level of page

Each level of security must ensure the possibility of executing the action and review of the data only to a certain specified entities under a certain conditions. Figure 4 shows the relationship between the six levels of security to be realized when implementing this process.

1. The level of database

There should be a database user who will have access rights to all actions of the database and who will have access to all data in an information system. This user has all rights of access to all data that are needed for work. Its features, such as user name and password, knows only the chief administrator of the database. For the purposes of access to data from the database, it is necessary to access the application server which contains username and encrypted password.

2. The level of web services

Web services that are used in applications of the information systems can provide more tools to help out of which the Oracle Web Service Manager is often used, which is primarily used against Oracle databases (OWSM). Exposed Web services can not be accessed except through defined channels and defined access rights. All monitoring or unauthorized attempts to access web services is done through OWSM application on application server.

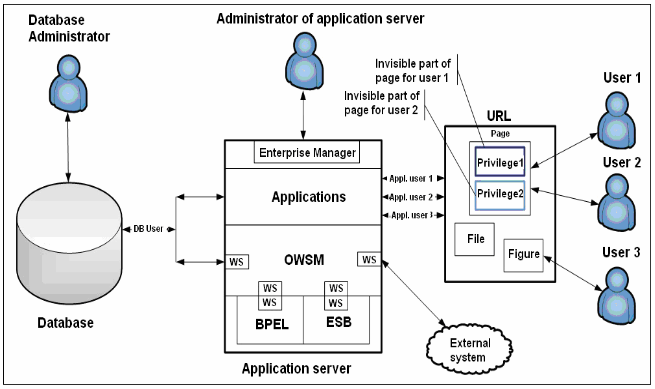


Figure 4. Information Security Management in Information system Telecom operator

3. The level of application server

Oracle Application server has a application for server management - Enterprise Manager Application Server Control. This application is executed on the server and it can be accessed by knowing the administrator username and password. The main application server administrator is responsible for creating, assigning roles and access rights to the application server. Roles and users of applications can be maintained in the application server.

4. The level of the application

Security-sensitive applications and information system components are divided into: URL (images, files, etc.), page (JSP, HTML, CSS, etc.) and part of the site. Any safety sensitive component represents a system of privilege. List of privileges that a user can have is constructed on that foundation. All privileges are grouped into application roles. Whenever security-sensitive components are accessed, it is checked whether the user belongs to a role that has the right of access privileges. If it does, the process goes smoothly. If user has no right of access, he gets notified that he is trying to access a protected resource. If it is the page that the user is accessing, then the forbidden part of the page is invisible or the access is denied. Most of the application options are in the menu of applications. Each user will have defined rights to access certain options of the menu. The real URL that is shown during the selection of menu options is presented as a privilege and it is indirectly assigned to a user through a specific role.

5. Application roles of access to applications

The goal of application roles is to group access rights in order to facilitate the granting, revocation or modification of the rights. All users belong to a single application of application roles. User roles activated in applications are shown in Table II.

6. The level of URL

URL level of security is tied to the resources that are accessed through the Application Server. URL can represent images, files, pages, etc. If it is a security sensitive component, then it will be assigned to the role whose members have access to these components. They will be located in specific-protected directories, which will be accessed only with prior checking of the access rights. If it is a safety sear components, then they can be located in public folders that can be accessed simply by specifying the URL of the web address in your browser.

TABLE II. DEFINED ROLES RIGHTS OF ACCESS TO INFORMATION SYSTEM TELECOM OPERATOR

The role of user applications	Description of the role of application users
Application user	User that monitors other users.
Application Administrator	User who performs user registration process and has access to codes
IT Administrator	Super user who manages the applications, maintains codes and review the logs.
Back Office Administrator	User who has a right of access to all options of applications in information system.
Front Office Administrator	Users that has greater subset of the access rights as opposed to the Administrator of the Back-Office that is required to work at the Telecom operators counter service.
Call Centre Administrator	Users that has greater subset of access rights as opposed to the Administrator of the Back-Office that is required to work at the Telecom operators call center.
VPN Administrator	User who has a right of access to VPN and other parts of applications required for managers of large accounts of Telecom operator

7. The level of page

If a review or modification of data in a table on the form: a user will be able to review only the information from the table, while others will be able to change data in the table. This is the hardest part of the implementation, so it is recommended that, whatever possible, it is implemented through one of the above ways, and then if there is no acceptable solution this method is to be approached.

G. Supplier Management

Telecom operators need to save money and human resources engage partner firm that will assist in implementation of a new information system. Modules that are implemented by the firm partner during the project, are obligatory to be maintained by firm partner after the system is available for production [8], [23]. Accordingly, it should define levels of support that the contract partner of the firm should implement in order to define hierarchies for problem solving (Table III).

With levels of maintenance contract should define levels of problems, definitions of failures, response time and problem removal that due to the contract should be addressed by the firm partner (Table IV).

After defining the level and types of problems, we can define a workflow to solve problems for which the firm partner should be in charge of according to the contract (Figure 5).

A signed agreement between telecom operator and companies partner needs to define the total time of the entire information system's construction and a time deadline to complete the activities for which the partner firm in charge of. The selected firm partner should be an IT company that has already implemented a number of systems, and it should take over much of the analysis of the new system andthe entire design and implementation of a whole new system. Figure 5 shows the workflow of the support in the five defined levels of the support in order to resolve the problem by the firm partners.

Telecom operators should take the following actions:

1. Planning, system analysis and definition of requirements specifications in accordance with all the necessary activities (final project and a term plan, defining frameworks and boundaries of the system, analysis existing processes and applications, defining business processes and specification of requirements that the new system should satisfy, as well as setting technical architecture, systems hardware platform-way necessary for project development).
2. Conduct testing of new systems implemented in the category of functional, integration, and tests end2end final checks.
3. To build a production database, and create a production platform on the basis of a detailed written plan for transition to production.
4. Integrate new system with the billing system, execute the migration of data from old system to the new system according to a pre-defined plans for transition to the new system.
5. Conduct user training that will work on a newly implemented information system.

The company partner should execute the following activities:

1. Execute the whole system design that includes architectural solution's design, modeling and database design (logical and physical data model) and the creation of the development of the database.
2. Set up and test system development platform.
3. Implement the security level and define the role of administrative control over the new system.
4. Design, build and develop applications including the preparation of interfaces to other applications, with special emphasis on the interpretation of the business logic and processes.
5. Define the integration of the old information system's customer base.
6. Define, encode, and make all the necessary mechanisms for the integration of information systems with other relevant information systems.
7. Document system as defined by the methodology of development of the information system within the telecom operator and the recommended methods of the partner companies in all stages of solution's development. Final papers should be such to have a form of the derived state with a detailed plan of the transition to producing.

TABLE III. LEVELS OF THE MAINTENANCE OF THE INFORMATION SYSTEM, WHICH DUE TO THE CONTRACT NEED TO BE MAINTAINED BY A SUPPLIER

The level of maintenance system	**The job of maintaining the level of the system level support in the maintenance of a new information system**
Level 1	Logging and tracking requests of Telecom Operator
Level 2	First line of support, location of problems and issues forward in the right department
Level 3	Support the site for normal technical requirements
Level 4	Technical support in case of escalation of problems that precedes the software engineering
Level 5	Software engineering, which includes work on the code and changes the design of the system

TABLE IV. CATEGORIES OF THE PROBLEMS WITH THE TIME FOR RESOLUTION THAT SUPPLIER MUST SOLVE DUE TO THE CONTRACT

Level of the problem	**Definition**	**Time for call off**	**Removal time**
Level I	A complete system crash	1 hour of receiving a verbal notification of the problem	24 hours after receiving oral notice of the problem
Level II	Fall of crucial part of the system	24 hours after receiving oral notice of the problem	48 hours after receiving oral notice of the problem
Level III	Fall of operating part of the system that is crucial for the whole system	2 business days of receiving notice of the problem	6 working days of receiving notice of the problem
Level IV	A minor problem that does not affect the operation of the system	3 working days of receiving notice of the problem	7 working days of receiving notice of the problem
Level V	A request of Telecom operator for additional functionality in the system	By agreement with a management of foreign firm	In case of agreement, the time of delivery and installation as well as the prices will be defined in cooperation with the telecom operator

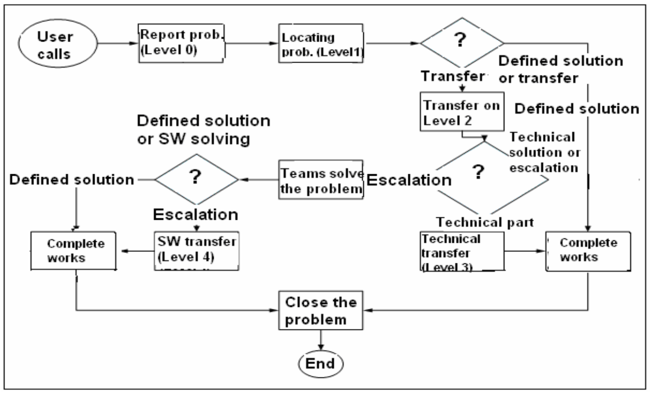


Figure 5. Workflow of the support for problem resolution for which the supplier is in charge of

III. CONCLUSION

For the realization of the Service Design processes in an information system of the telecom operator, it is necessary to establish more teams of IT professionals for each of the Service Design processes before starting the actual implementation of selected recommendations of ITIL V3 [2], [6]. Service Catalogue Management Code required a team that will define the seven product catalogues, which should be given to the end user. Service Level Management requires the formation of three teams of IT specialists to define three different types of agreements: Service Level Agreements, Operating Level Agreements and Contracts. With Capacity Management only one team that should define five functional tests that measure the current and future IT available capacities that are available to the system itself is required. Availability Management demands formation of the two teams of IT professionals: the first team to define system components that should go into the future plan of the availability and other team that should be responsible for implementation of all of the defined items of the accessibility plan. When IT Service Continuity Management is concerned four teams are required for the implementation of each of the four phases of the Business Continuity Plan. Information Security Management requires the formation of two teams of IT experts: the first team that is responsible for implementation of the six defined levels of security, and a second team to assign administrative roles for management of the system itself. For Supplier Management three teams of IT professionals are required: the first team that is responsible for monitoring activities to be implemented by the Telecom operator while implementing information system, the second team to monitor activities to be implemented by the firm partner when implementing IT Systems specialization, and the third team to monitor the speed of solving problems at levels of priority for which the company partner is responsible.

Each of the 16 teams in all seven Service Design processes, except for three teams of Supplier Management, is independent from one another and can be carried out continuously. Time limit for completion of team activities for the first six Service Design process is a maximum of 2 months. Regarding the activities of Supplier Management teams, they start with just three months after signing the agreement with the partner company and are valid depending on the category of agreements which can be six months, one year or two years.

Further research in this area is related to the development of similar studies for the remaining four phases of ITIL V3 standards, primarily for the Service Strategy phase because of its close links with the Service Design phase.

ACKNOWLEDGMENT

The autors would like to thank management and technical staff of the BH Telecom for suggestions regarding creation of this document.

REFERENCES

[1] S. Taylor, M. Iqbal, and M. Nieves, “ITIL Version 3 Service Strategy”, The Office of Government Commerce, May 2007.

[2] S. Taylor, V. Lloyd, and C. Rudd, “ITIL Version 3 Service Design”, The Office of Government Commerce, May 2007.

[3] S. Taylor, S. Lacy, and I. Macfarlane, “ITIL Version 3 Service Transition”, The Office of Government Commerce, May 2007.

[4] S. Taylor, D. Cannon, and D. Wheeldon, “ITIL Version 3 Service Operation”, The Office of Government Commerce, May 2007.

[5] S. Taylor, G.Case, and G.Spalding, “ITIL Version 3 Continual Service Improvement”, The Office of Government Commerce, May 2007.

[6] J. van Bon, A. de Jong, A. Kolthof, M.Pieper, R. Tjassing, A. van der Veen, and T. Verheijen, ”Foundations of IT Service Management Based on ITIL V3”, The Office of Government Commerce, September 2007.

[7] R. Steinberg, “Measuring ITIL: Measuring, Reporting and Modeling – the IT Service Management Metrics That Matter Most to IT Senior Executives”, The Office of Government Commerce, April 2008.

[8] J. van Bon, A. de Jong, A. Kolthof, M.Pieper, R. Tjassing, A. van der Veen, and T. Verheijen, ”Service Design based on ITIL V3”, The Office of Government Commerce, June 2008.

[9] P. Brooks, J. van Bon, and T. Verheijen, “Metrics for IT Service Management”, The Office of Government Commerce, April 2006.

[10] L. Shwartz, N. Ayachitula, M. Buco, G. Grabarnik, M.N.Surendra, C. Ward, and S. Weinberger, “IT Service Provider’s Multi-Customer and Multi-Tenant Environments”, 9th IEEE Int. Conference on E-Commerc Technology and 4th IEEE Int. Conference on Enterprise Computing, E-Commerce and E-Services, July 2007, pp. 559-566.

[11] A. Wegmann, G. Regev, G.A. Garret, and F. Marechal, ”Specifying Services for ITIL Service Management”, The International Workshop on Service-Oriented Computing Consequences for Engineering Requirements, 16th IEEE International Requirements Engineering Conference, September 2008, pp. 85-94.

[12] S. Kovacevic and F. Orucevic, “Using ITILv3 methodology for implementing new e-mail services in operator for producing and distributing electrical energy”, in Proceedings of the 32nd International Conference on Information Technology Interfaces, June 2010, pp. 147-154.

[13] M. Sharifi, M. Ayat, and S. Sahibudin, “Implemented ITIL-Based CMDB in the Organizations to Minimize or Remove Service Quality Gaps”, 2nd Asia International Conference on Modeling & Simulation (AMS), May 2008, pp. 734-737.

[14] M. Spremic, Z. Zmirak, and K. Kraljevic, „IT and Business Process Performance Management: Case Study of ITIL Implementation in Finance Service Industry”, 30th International Conference on Information Technology Interfaces, June 2008, pp. 243-250.

[15] E. Orta, M. Ruiz, and M. Toro, ”A System Dynamics Approach to Web Service Capacity Management”, 7th IEEE European Conference on Web Services, November 2009, pp. 109-117.

[16] J. Zeng, “A Case Study on Applying ITIL Availability Management Best Practise”, Contemporary Management Research, vol. 4, December 2008, pp. 321-332.

[17] X. Su, “An Overview of Economic Approaches to Information Security Management”, Centre for Telematics and Information Technology, University of Twente, August 2006.

[18] M. Brenner, “Classifying ITIL processes: A Taxonomy under Tool Support Aspects”, 34th Hawaii International Conference on System Sciences, July 2006, pp. 19-28.

[19] G. Blokdijk and I. Menken, “Service Level Management Best Practise Handbook: Building, Running and Managing Effective Service Level Management SLAs – Ready to use supporting documents bringing to ITIL Theory into Practise”, Emereo Pty Ltd, October 2008.

[20] G. Blokdijk and I. Menken, “Capacity Management Best Practise Handbook: Building, Running and Managing Effective Capacity Management – Ready to use supporting documents bringing to ITIL Theory into Practise”, Emereo Pty Ltd, October 2008.

[21] B. Johnson and J.Higgins, “ITIL and the Sotware Lifecycle: Practical Strategy and Design Principles”, Van Haren Publishing, August 2007.

[22] M. Wedemeyer, G. Blokdijk, and I. Menken, “The ITIL V3 Service Management Awareness Pocket Guide – The ITIL V3 Pocket Toolbook: A Quick Reference Guide to all the processes and activities for Improving Quality and Speed”, Emereo Pty Ltd, December 2008.

[23] G. Blokdijk and I. Menken, “Supplier Management Best Practise Handbook: Evaulating, Sourcing, Managing and Delivering Supplier Excellence In Relationships. Quality and Costs”, Emereo Pty Ltd, August 2008.

[24] G. Blokdijk and I. Menken, “Availability Management Best Practise Handbook: Building, Running and Managing a IT Security Management Governance, Risk and Compliance Process – Ready to use supporting documents bringing to ITIL and GRC Theory into Practise”, Emereo Pty Ltd, October 2008.

[25] G. Blokdijk and I. Menken, “IT Security Management Best Practise Handbook: Building, Running and Managing Effective Availability Management – Ready to use supporting documents bringing ITIL Theory into Practise”, Emereo Pty Ltd, October 2008.

[26] G. Blokdijk and I. Menken, “IT Services Portfolio Management Best Practise Handbook: Planning, Implementing, Maximizing Return on Investment of Strategic IT Portfolio Management– Ready to use bringing Theory into Action”, Emereo Pty Ltd, July 2009.

Online Service Similarities and Reputation-based Selection

Oana Dini[*], Pascal Lorenz[**], Abdelhafid Abouaissa[**], Hervé Guyennet[*]
[*)] Université de Besançon, France
[**)] Univeristé de Haute Alsace, France
oana.dini@univ-fcomte.fr, lorenz@ieee.org, a.abouaissa@uha.fr, guyennet@lifc.univ-fcomte.fr

Abstract – **Selection of the most appropriate service by correct invocation is a challenge. This is due to the difficulties of correctly exposing proper ways to invoke a service, to the variety of services, from on-line services, software pieces, to shopping, and to different invoker behavior. When considering invoker feedback, service ranking based on the user's perception, or based on the recommenders' statistics are relevant. The paper presents adapted approaches to select services based on distance and similarity, and introduces a similarity taxonomy to better tune various kinds of service invocation under specific constraints, such as relaxation, type of similarity, context, and service ranking. Selection is also based on the feedback from the user. The proposed model is used for building a selection algorithm that allows variations on service invocation.**

***Keywords** – service similarity; similarity class; temporal similarity; selection patterns*.

I. INTRODUCTION

The large spectrum of user behaviors (and, in general, the variety of needed services) leads to the need of similarity-based matching, when a given service is required. Traditionally, the notions QoS (Quality of Service), and QoE (Quality of Experience) deals with these aspects. However, the perfect matching and the approximate matching depend on a large number of factors. For example, if we consider Web Services dedicated to weather forecast, location, month/day/year, parameters (rain, wind, temperature, and pressure) can be appropriate parameters when inquiring. Definitively, there are several forecast services, and the experience of a particular user might differ from one forecast service to another. Some provide information that is more accurate than others (i.e., data is more frequently updated), history is better preserved by particular services, via backward search, e.g., Weather Underground, etc. A similar problem is observed when choosing and downloading a particular piece of software, when inquiring for a specialized on-line book shop, or when looking for a service providing the most updated world-wide information. Finally, some services offer a friendlier interface for searching, ordering, and getting delivered a particular need (i.e., personalized interface, myAccount, etc.).

There are meta-services, providing the service at choice. Such examples are those for buying flight tickets, where the cheapest, the quicker, or other selection criteria are used for service selection. Other meta-services are for selecting the most appropriate software to download, or for booking a hotel. In most of the cases mentioned above, one criterion is usually considered to select from an existing service pool.

Two phases involve service features, (i) service discovery (locating) and (ii) service selection (in the case of a set of services, relatively satisfying the needs with similar degrees of satisfaction). Both phases require special mechanisms to assess service similarity. Meta-services have a restrained number of known services, that are well localized and whose parameters are also limited. Because of this, the selection appears to be less complex. With a well known service and limited criteria (usually one search/ selection criterion), similarity is relatively easy to be determined.

The above considerations are no longer valid for a large spectrum of properties a service might expose in order to satisfy a given service request. To satisfy a request, service similarity plays an important role for timely identification and delivery, and for an optimal (maximal) customer (invoker) satisfaction. Customer satisfaction is expressed by QoE, on-line feedback, service ranking, and manifested by variations of QoS to keep service costs and satisfaction in synchrony.

The paper deals with service similarity and proposes a adaptive similarity taxonomy and mechanisms to handle service discovery and service selection considering service specification, end-user (requester) perception, and service reputation. In Section II, existing approaches for service similarities are presented. Section III introduces a context-based similarity model, including distance and similarity metrics, a similarity taxonomy, and other facilities to consider service ranking and feature relaxations. Section IV presents an algorithm to compute a minimum set of existing services satisfying a given query, following the newly introduced model. Section V concludes on the approach and presents further developments.

II. RELATED WORK

Finding similar services (approximate but satisfactory matching) is somehow similar to (i) text matching, (ii) schema matching, or (iii) software-component marching. For some text matching solutions (information retrieval) mechanisms based on term frequency are used [7][8]. In schema matching, special techniques are using semantics of the schemas to suggest schema matching [9]. Mainly, linguistic and structural analyses, as well as domain knowledge, are methods to handle schema matching. When expanding to software component matching [10] (considerably used in software reuse) component signature and program behavior (usually formally defined) are considered; in this case, data types and post-conditions should be considered for matching. However, these techniques are not suitable for Web Services [6], as data types and post-conditions are not available. Usually, such a service has a name and text description in UDDI **(**Universal Description, Discovery, and Integration) registry, operation decryptions, and input/output descriptions; the last two are usually specified in WSDL (Web Service Description Language).

Dong *et al.* [6] proposed criteria for associating similar terms. They introduced the cohesion/correlation score, as a measure of how tight two terms are. However, they do not consider particular characteristics of a term. They applied the score only to Web Services. We start from the idea that services similarity has a meaning only between services than can be context-oriented and belong to a cluster (e.g., invoking a service gives a list of similar operations with similar results). Other approaches consider both diversity and similarity at the same time, having the distance as a metric [11]. We adopt these metric (see Section III) and adapt them to the service similarity computation.

In fact, specific to each service, there are particular service parameters that are agreed upon between a provider and a subscriber, commonly settled by the SLA (Service Level Agreement). On the provider side, the SLA parameters are used for technical audit and litigations (leading to penalties or bonuses towards a given user or class of users). Specific on-line and off-line measuring mechanisms for SLA metrics and specialized audit techniques have been proposed. On the consumer side, the subscribers' satisfaction is gathered and mapped to the audit results to validate a given service, to detect flaws in delivering a service, and to ultimately build a view on service reputation. In general, a record is handled per service or per products, with respect to a given subscriber or a class of subscribers. Feedback can be used to enforce service similarity.

In this paper, we expand the cluster-based similarity to service similarity and introduce similarity taxonomy, where the service consumer has a weight in deciding service similarity. The idea is to establish service ranking (and reputation) inside a given cluster, and define similarity considering service-provider and service-user feedback.

III. A CONTEXT-BASED SIMILARITY MODEL

Then main idea of our approach is (i) having well defined service clusters, (ii) compute the distance between service feature, (iii) evaluate service similarity, based on service features, (iv) consider user-, service,- and producer-based similarity reflected by the appropriate reputations, and (v) evaluate how interchangeable two services are. When a service query is issued, the algorithm we propose selects the most appropriate service, considering both distance and similarity between services.

A. Identifying clusters of similar services

Expanding what was mentioned in [6], service cohesion of a service cluster must be strong (best potential to be similar), while correlation between two service clusters should be weak (service independence). We say that service s_1 is similar with s_2, and note $s_1 \sim s_2$, if the similarity confidence is greater than a given threshold δ. In a cluster S with $\|S\|$, where $\|x\|$ is the cardinality of x, we redefine cohesion and correlation as follows:

$$\mathrm{Cohe}_S = \{(s_i, s_j) \mid s_i \sim s_j \, (\sim_{\mathrm{thres}} > \delta)\} \,/\, (\|S\| \text{ x } (\|S\| - 1)) \quad (1)$$

and

$$\mathrm{Correl}_{S,S'} = (A\,(S, S') + A\,(S', S)) \,/\, 2 \text{ x } \|S\| \text{ x } \|S'\|, \quad (2)$$

where

$$A\,(S, S') = \| \{s_i, s_j \mid s_i \in S, s_j \in S' \; si \sim sj \mid \sim_{\mathrm{thres}} > \delta\} \| \quad (3)$$

with $\sim_{\mathrm{score}} = \mathrm{Cohe}_S \,/\, \mathrm{Correl}_{SS'}$ (4)

defining the similarity score.

We notice that $\sim_{\mathrm{score}}$ defines similarity classes based on the preexisting service clusters. To enhance the similarity score, clusters aggregation and clusters split operations are possible. Conditions and assessments for doing these are presented in [6].

B. Distance metrics for service similarity

Let us assume that a service *s* has *n* features (usually called data-points, as they are expressed by concrete values in an *n* dimensional space). The following distance methods are adapted for comparing services:

(a) Service Euclidian distance between two services in the *n* dimensional space

$d(s_1,s_2) = 1/n\ \Sigma\ (a_{1i} - b_{2i})^{**2}$, for all i = 1…n
(5)
where a_i, b_i are service features.

(b) Service city-block distance
$d(s_1,s_2) = 1/n\ \Sigma\ |a_{1i} - b_{2i}|$, for all i = 1…n
(6)

(c) Service Pearson correlation coefficient

$r(s_1,s_2) = 1/n\ \Sigma\ ((a_{1i} - \underline{a})/\sigma_a) \times ((b_{2i} - \underline{b})/\sigma_b)$,
(7)
where $\underline{a}$ and $\underline{b}$ are the sample mean of a_i and b_i respectively, and σ_a and σ_b are the sample standard deviation of a_i and b_i.

The service Pearson distance is defined as
$d(s_1,s_2) = 1 - r(s_1,s_2)$
(8)

(d) Service Cosine similarity
$d(s_1,s_2) = \cos(\theta) = (s_1 \bullet s_2) / (|s_1|\ |s_2|)$
(9)
where ● is the vector product of s_1 ans s_2.

By selecting a service distance metric, a clustering algorithm computes the distance matrix between two services. Mostly, (a) and (b) of the above are satisfying the triangle inequality, as true metrics.

C. Classes of similarities

In order to select the most appropriate service, we introduce producer-based similarity ($\sim_{prod}$), recommender-based similarity ($\sim_{recc}$), and user-based similarity ($\sim_{user}$). Producer similarity is based on the expectation, recommender's similarity is statistics-based, and user similarity is based on user feedback. In this taxonomy, $s_1 \sim_{prod} \{s_2, s_3, \ldots\}$ define a cluster of similar services, as defined by the producer.

To refine service similarity, we introduce the notions of *primary service features* and *secondary service features*, as shown in Figure 1,

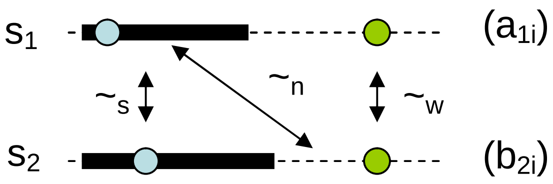


Figure 1. Similarity classes.

where the bold items represent primary service features (A_1 set), and the dashed items represent secondary service features (A_2 set) (similar for s_2)

We introduce strong, weak, and normal similarity, represented by $\sim_s$, $\sim_w$, and $\sim_n$, respectively.

Therefore, $(s_1 \sim s_2)$ =

$= \sim_s$, iff all $a_{1i} \epsilon A_1$ and $b_{2i} \epsilon B_1$

$= \sim_w$, iff all $a_{1i} \epsilon A_2$ and $b_{2i} \epsilon B_2$

$= \sim_n$, iff there are $a_{1i} \epsilon A_1$ and $b_{2i} \epsilon B_2$ or
there are $a_{1i} \epsilon A_2$ and $b_{2i} \epsilon B_1$ (10)

Similarity composition allows to capture all possible combinations, e.g,, $\sim_{prod/s}$ represents a strong similarity defined by the producer, based on the primary service features.

A refinement of feature-based similarity can be expressed when service features do not show a direct semantic matching, but feature composition might lead to such a match. Considering a subset of a service feature for a given service equivalent with a feature for a service for which the similarity is computed, we introduce feature composition-based similarity, as shown in Figure 2.

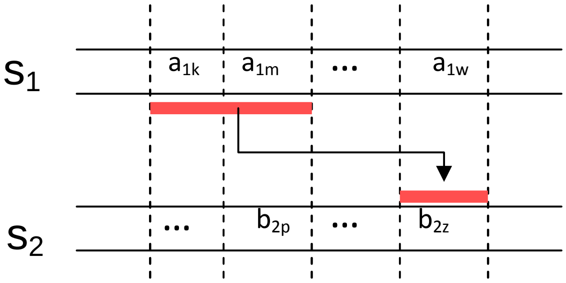


Figure 2. Feature composition-based similarity.

$s_1 \sim_{a1k,a1m\ /\ b2z} s_2$ (11)

with the semantic that the values of a_{1k} and a_{1m} composed are similar to the values of b_{2z}. Composition might be any arithmetic or Boolean operator, according to the nature of the features, e.g., if sets, then 'U' (union), if values, then '+' (addition), etc. If type, and a_{1k}:T1 and a_{1m}:T2, and b_{2z}:T3, then, then T3 is a subtype of either T1 or T2.

Combination between $\sim_s$, $\sim_w$, and $\sim_n$, and feature composition-based similarity can be applied following (10).

D. Updating similarity

When evaluating service similarities, perfect match of service features is desired, but rarely found, due to some continuous values of the features. For example, looking for a service offering the weather temperature with an accuracy of 0.1^oF is not feasible. A query on what month the temperature is 67.3F might have no match; but, for a given location, a query for what month shows [75-80] ^oF might be answered by April or May, if a Mediterranean area. We identify two possible relaxations when performing the matching.

D.1 Context-based feature migration

In time, and based on business models or customer feedback, some primary features become secondary, and vice-versa. Even more, at the same time, in different contexts, a feature can belong to either primary or secondary feature sets.

Let C = $\{c_i\}$ a set of contexts and

$$s1 ::= (A_1 U\ A_2)_{context\ =\ c1},\ with\ A_1 \cap A_2 = \phi$$

$$s1 ::= (A'_1 U\ A'_2)_{context\ =\ c2},\ with\ A'_1 \cap A'_2 = \phi \qquad (12)$$

then, the following is possible:

$$s_1 \sim_{context\ =\ c1} s_2$$

$$s_1 \sim_{context\ =\ c2} s_3 \qquad (13)$$

D.2 Feature relaxation-based similarity

Service features are not always perfectly matching (so goes for query matching, as well). Most of the time, the exact matching is not mandatory, e.g., if a service feature has a numeric value, a variation of a_{1i} (usually symmetric, but not necessarily) of +/- α_{1i} is allowed. As a result, the similarity metrics presented in II.B can be relaxed. The same relaxation can be applied for similarity on data type/subtype, for similarity concerning the set of interface operations, or similarity concerning variations of an algorithm implementation. For example, when a query (with explicit relaxation of +/- 2ms) targets a service with a response delay of 10ms, any service offering a delay within [8ms, 12ms] is a desired matching. With no explicit relaxation delay, 10ms is mandatory. In this case,

$$s1 \sim_{a1i\ +/-\ \alpha 1i}\ s2 <=> b_{2i} \epsilon [a_{1i} - \alpha_{1i}, a_{1i} + \alpha_{1i}] \qquad (14)$$

where a_{1i} and b_{2i} are the corresponding features of s_1 and s_2, respectively.

E. Recommender-based similarity

Recommender mechanisms rank [1] the products or services based on feedback received after a series of recommendations and successful transactions. The *ranking* is subject to incomplete, fictitious feedback, volume of transactions for a given product or provider, and confidence in feedback. Based on statistics, the recommender computes its own *ranking* per product, defining the reputation (r) of a service/product.

Considering a set of service clusters a recommender builds based on type of services/products,we define:

Cluster = $\{cluster_i\}$
with s_1 ϵ $cluster_i$ and s_2 ϵ $cluster_{i,}$ for a given service feature

$$s_1 \sim_{feature\ =\ ai} s_2 ::= |rank_{s1} – rank_{s2}| < \varepsilon_{ai} \qquad (15)$$

In general,

$$s_1 \sim_{Uai} s_2 ::= \max \{|rank_{s1} – rank_{s2}|\} < \min \{\varepsilon_{ai}\} \qquad (16)$$

F. Customer feedback reputation-based similarity

Based on customer individual metrics, context, and potential query with relaxation, a reputation is associated with a service/product. Heuristics for updating the reputation have been presented in [1][2]. In general, the following information is available:

s <r>: each service <s> has an associated reputation <r>
P_i<s,r_i>: each provider offers a service with its associated reputation
P_j<s,r_j>: another provider can offer the same service with a different associated reputation
u <e, c>: a user *u* has a credibility and confidence metrics associated with it
For simplicity, we consider that <e, c> are the same for any service.

For a given user, we define similarity in terms of r_s

$$s_1 \sim_{feedback} s_2 ::= |r_{s1} – r_{s2}| < \varepsilon_0 \text{ with } e> e_0 \text{ and } c>c_0 \qquad (17)$$

In the following, the newly introduced model is used by an algorithm to identify the most suitable service to satisfy a query for a service.

III. ALGORITHM FOR SERVICE RETRIEVAL USING SIMILARITY

We introduced a similarity model and classes of similarity that allow a user (invoker) to use a service in a given context, allowing or not precise relaxation for some service features, and under different types of similarity (strong, weak, normal). Distance metrics were also adopted for services, in order to cluster the most suitable services for a particular query, before computing the similarity.

Based on the model introduced in Section III and on the user model [2] and reputation [1], a query for a service *s* can be expressed as

Q (s, similarity type, context, with/without relaxation on $\{a_{1i}\}$)

The algorithm presented below illustrates the main steps to reach a service proposal that can be a set, a given service, or no service at all.

Algorithm for finding a requested service query **Q**, *based on similarity between potential satisfying services*

```
1:  begin
2:  identify the service cluster [see (4)]
3:  select a distance metric [see (5)-(9)]
4:  calculate distance between all $s_i$ in the cluster
5:  select a subset {$s_k$ with min {$d(s_i, s_j) < \varepsilon$}
6:  if Q with relaxation
7:    apply (10) and (11) for all mentioned features
8:  if not
9:      if Q with context
10:       apply (12) and (13)
11:     if not
12:   compute a subset {si} of the set found before
        step12
13:   select {sl} from the subset of step 12, with
          rank ($s_l$) > $\delta_1$ and $r_{feedbakc} > \delta_2$ [see (16) and (17)]
14: select a subset for the subset of step 13
15: return the subset of step 14
16: end
```

Note that the output of step 15 might be an empty set, or a set having many recommended services complying to the query conditions.

The complexity of the algorithm is given mostly by the number of service features that can be considered with relaxations.

A variation of the algorithm was experimented with relaxation conditions for a set of contexts. The number of features with relaxation, the number of contexts, and the number of services into a cluster determine the performance of the algorithm.

Different experiments on the on-line Barnes & Nobles system (on-line bookstore) show a reasonable improvement on the precision the algorithm returns after running various numbers of queries and varying different conditions.

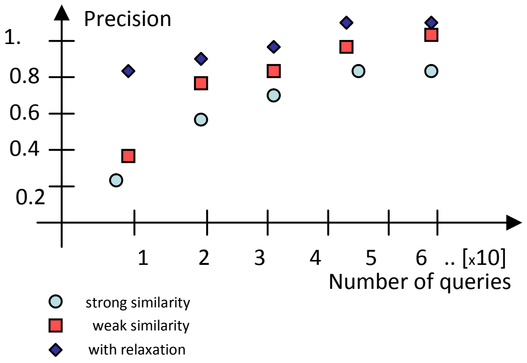


Figure 3. Precision of service returned to queries with different types of similarities

It is no surprise that a service satisfying a query with relaxation reaches faster and with a higher precision the query expectation.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented an approach for service invocation using similarity taxonomy with weak, strong, and normal similarity. Practically, services are clustered and service distance/similarity metrics were adopted from text-based domains. A reputation-based mechanism (introduced in [1][2]) is used in combination to context-based similarity and feature relaxation methods to identify a set of services that better serve a given query.

We also introduced the techniques of feature aggregation when similarity is evaluated, and the continuous update of feature classification, i.e., primary/secondary, according to the context. More work should be done on these two items, as semantic-based aggregation should be considered.

REFERENCES

1. O. Dini, P. Lorenz, and H. Guyennet; An Enhanced Architecture for Web Recommenders, SERVICE COMPUTATION 2009, IEEE Press, pp. 372 – 378, ISBN: 978-1-4244-5166-1, Athens, Greece
2. O. Dini, P. Lorenz, A. Abouaissa, and H. Guyennet, Dynamic Feedback for Service Reputation Updates, ICAS 2010, pp. 168-175 ISBN: 978-1-4244-5915-5, Cancun, Mexico
3. C. Wu and E. Chang, Searching Services ‘in the web’: A Public Web Services Discovery Approach, SITIS 2007, The Third IEEE Conference on SignalImage Technologies and Internet-based Systems, pp. 321-328.
4. M. Paolucci, B. Shishedjiev, Xh. Zenuni, and B. Raufi, GHSOM-based Web Service Discovery, 2010 European Computing Conference, ISSN: 1790-5117, 2010
5. M. Szomszor, C. Cattuto, H. Alani, K. O'Hara, A. Baldassarri, V. Loreto, and V. D. Servedio, “Folksonomies, the semantic web, and movie recommendation,” In 4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0, 2007.
6. X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, Similarity Search for Web Services, The 30th VLDB Conference, Toronto, 2004
7. S. Cost and S. Salzberg, A Weighted Nearest Neighbor Algorithm for Learning Symbolic Features. Machine Learning, No. 10, 1993, pp. 57-78
8. L.S. Larkey and W. Croft, Combining Classifiers in text Classifications Techniques, ACM SIGIR 1998.
9. H.-H. Do and E. Rahm, COMA – A System for flexible Combination of Schema Matching Approaches, VLDB 2002
10. A.M. Zaremski and J.M. Wing, Specification matching of software components. TOSEM, No. 6, pp. 333-369, 1997
11. C. Bouras and V. Tsogkas, Improving text summarization using noun retrieval techniques, LNCS, Knowledge-based Intelligent Information and Engineering Systems, vol. 5178/2008, pp. 593-600

Collaborative Digital Library Services in a Cloud

Kurt Maly[1], Harris Wu[2], Mohammad Zubair[1], Milena Mektesheva[1]
Department of Computer Science, Old Dominion University, Norfolk, VA, USA

maly@cs.odu.edu, hwu@odu.edu, zubair@cs.odu.edu
1) Department of Computer Science, Old Dominion University, Norfolk, VA, USA
e-mail: maly@cs.odu.edu; zubair@cs.odu.edu, mmekt001@odu.edu
2) Department Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA, USA
e-mail: hwu@odu.edu

Abstract— **We have developed a web-based system that allows users to collaboratively organize large online image collections according to an evolving faceted classification schema. One of the major issues identified in the early deployment and evaluation in a university setting is the scalability of the system on traditional server implementations. Traditional computing cannot support ever-increasing number of users, documents, schema objects, schema history, and automated classification processes without difficult, expensive and time consuming resource reconfiguration. To address this problem we are proposing to move our system on a cloud-based Microsoft Windows Azure platform exposing it to users as a collaborative cloud service. Cloud computing will enable the Facet System to scale virtually unlimited. In this paper, we describe the architectural design for deploying our facet-based system and early prototypical modules as well as the work in progress implementing it on the Windows Azure platform.**

Keywords-**faceted digital library; windows azure; cloud computing**

I. INTRODUCTION

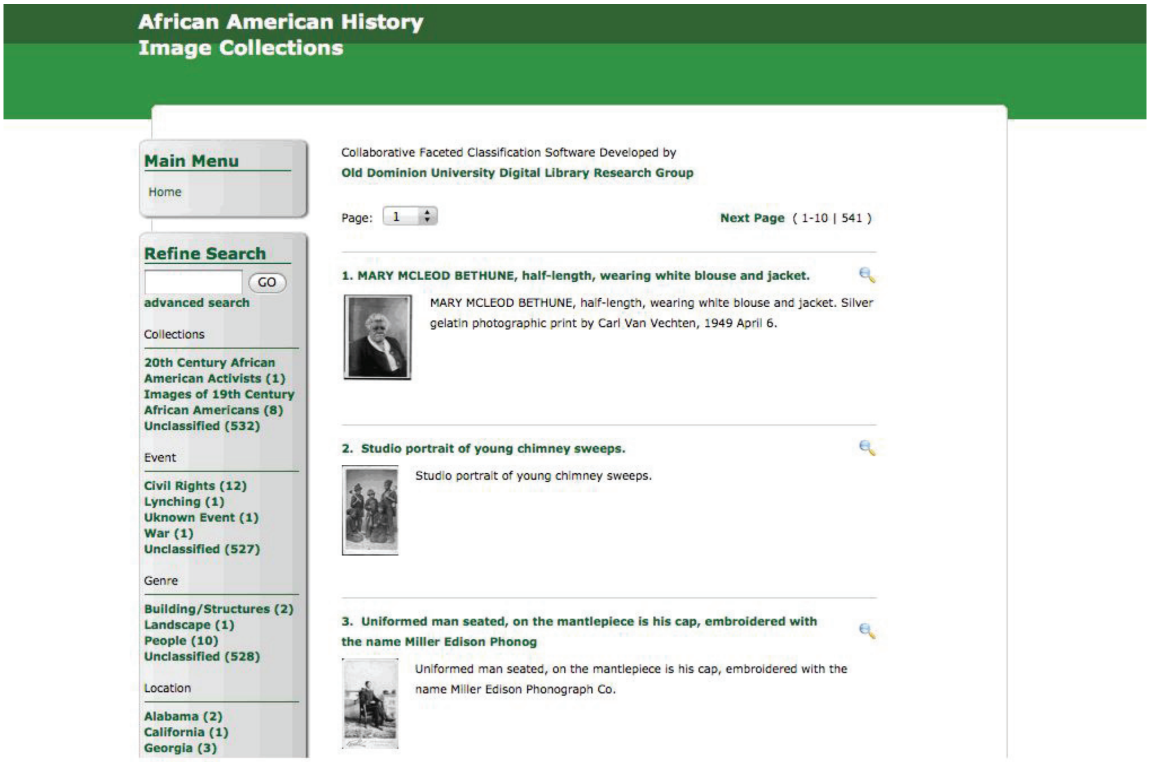
We have developed a web-based system that allows users to collaboratively organize large online multimedia collections into an evolving faceted classification [1; 2; 3]. The system includes backend algorithms that systematically enrich the classification and automatically classify documents [4; 5; 6]. Evaluation of the prototype system, hereby referred to as the Facet System, shows promise [7; 8]. One of the major issues identified is the scalability of the system on traditional server implementations. Traditional computing cannot support ever-increasing number of users, documents, schema objects, schema history, and automated classification processes without difficult, expensive and time consuming resource reconfiguration. To address this problem we are proposing to move our system on a cloud-based Microsoft Windows Azure platform [14], exposing it to users as a collaborative cloud service. Cloud computing will enable the Facet System to scale virtually unlimited. There are a number of cloud services available in the commercial market but, as an academic institution, we will propose to use the time being made available on Microsoft Azure by a joint RFP by the National Science Foundation and Microsoft. Evaluating a social classification service on Windows Azure will allow us answer research questions specific to large-scale deployment of social systems that harness and cultivate collective intelligence. In section 2, we provide a review of the existing facet based system with an emphasis on the compute intensive nature of some of the features.. In section 3, we describe more specifically the evaluations we have performed thqat demonstrates the existence of severe scalability issues. Section 4 gives our design for deploying the facet system onto an azure architecture. In the final section we lay out the future work tasks.

II. BACKGROUND

In this section, we briefly review the existing facet based system. Fig. 1 and Fig. 2 show the browsing and classification screens in the Facet System, through which users can browse and collaboratively evolve a multi-faceted classification. In addition to a global classification of a large collection, we are adding a personal schema feature to the Facet System. In Fig. 2, clicking on the “person” icon above the faceted classification (the 1st of the 3 icons) will display a personal schema. The personal schema allows user to have a personal, persistent, idiosyncratic view of the collection [9]. In addition, most likely a user is interested in only a subset of the collection, when the collection is very large. Furthermore, a category in a personal schema can be a dynamic link (like a “shortcut” in Windows File Explorer) to a category in the global collection, which is constantly evolved by the community.

Fig. 3 shows the classification screen with both global and personal schemas. The back-end algorithms utilize the metadata in personal schemas for enrichment (construction, pruning, etc.) of global schema and automated classifications. Except for shortcuts, maintenance of facets and categories in a personal hierarchy requires separate storage in the database. To reduce user efforts in classifying documents into her personal hierarchy, a user can instruct the system to automatically classify or recommend documents from the overall collection into the personal schema.

When automated classification is enabled for the personal hierarchy (in user preference settings), the backend algorithms take significant amount of computing resources for each additional user. Furthermore, our system supports schema history – which allows users to examine global or personal schema at any given point in time.



African American History
Image Collections

Main Menu
Home

Refine Search
GO
advanced search

Collections
20th Century African American Activists (1)
Images of 19th Century African Americans (8)
Unclassified (532)

Event
Civil Rights (12)
Lynching (1)
Uknown Event (1)
War (1)
Unclassified (527)

Genre
Building/Structures (2)
Landscape (1)
People (10)
Unclassified (528)

Location
Alabama (2)
California (1)
Georgia (3)

Collaborative Faceted Classification Software Developed by
Old Dominion University Digital Library Research Group

Page: 1 Next Page (1-10 | 541)

1. MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket.

MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket. Silver gelatin photographic print by Carl Van Vechten, 1949 April 6.

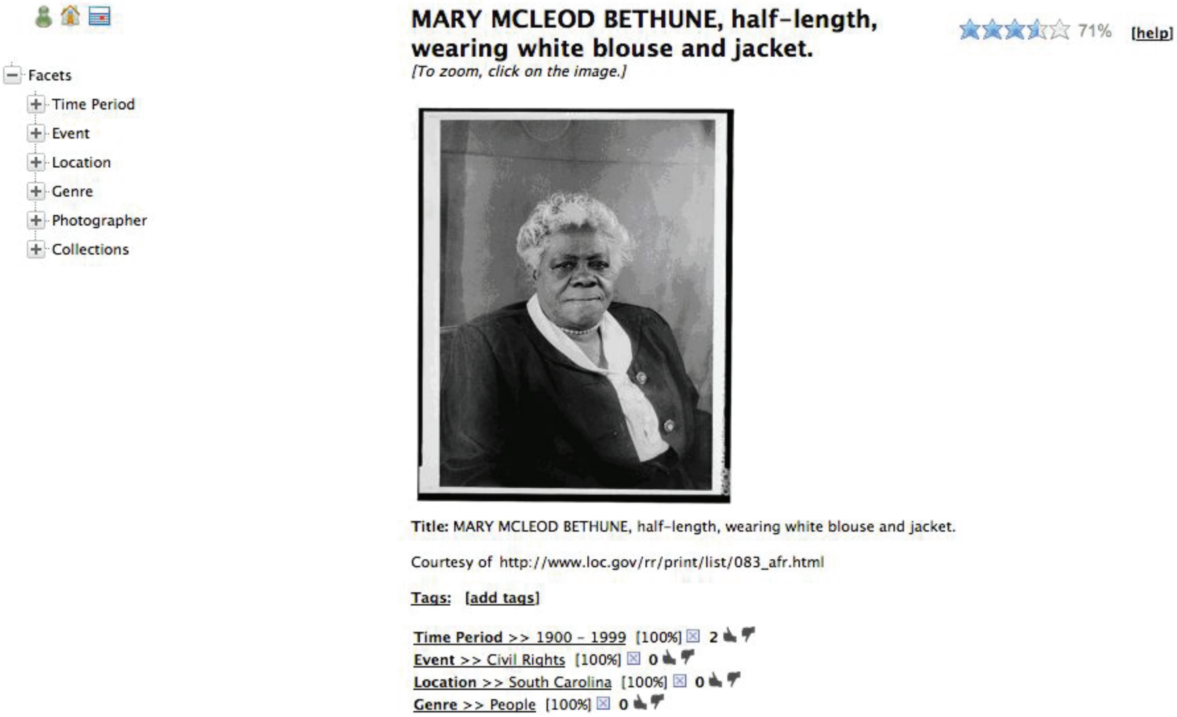
2. Studio portrait of young chimney sweeps.

Studio portrait of young chimney sweeps.

3. Uniformed man seated, on the mantlepiece is his cap, embroidered with the name Miller Edison Phonog

Uniformed man seated, on the mantlepiece is his cap, embroidered with the name Miller Edison Phonograph Co.

Figure 1. Browsing screen of the Facet System.



Facets
Time Period
Event
Location
Genre
Photographer
Collections

MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket.
[To zoom, click on the image.]

★★★☆☆ 71% [help]

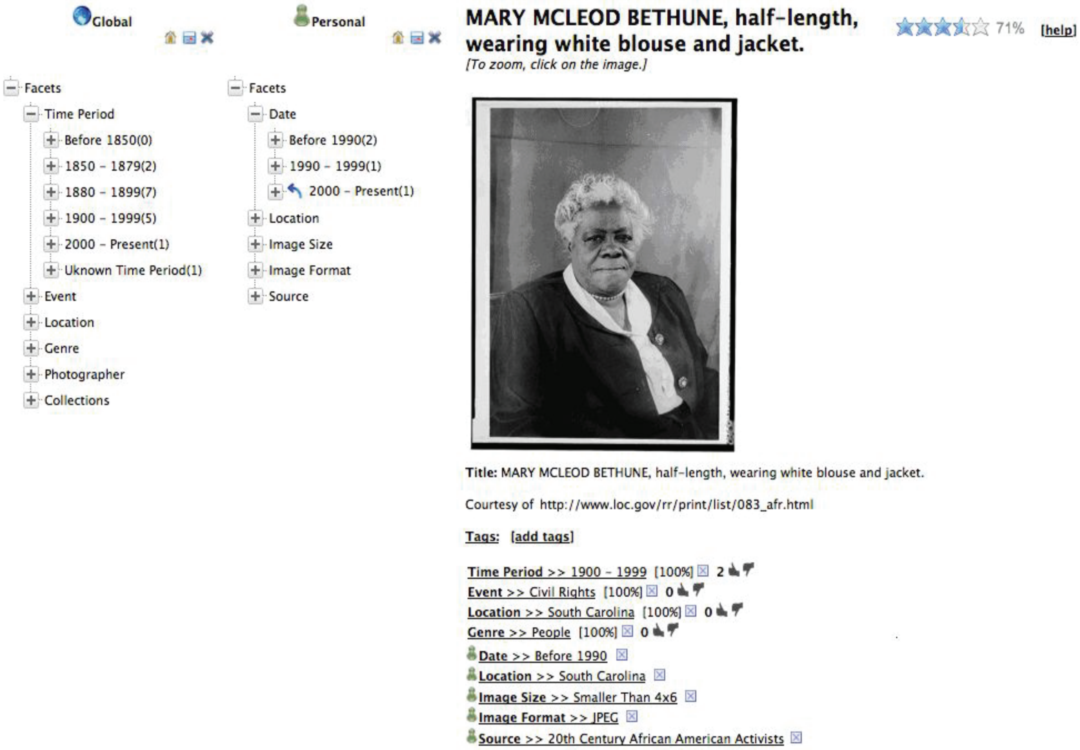
Title: MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket.

Courtesy of http://www.loc.gov/rr/print/list/083_afr.html

Tags: [add tags]

Time Period >> 1900 - 1999 [100%] ☒ 2
Event >> Civil Rights [100%] ☒ 0
Location >> South Carolina [100%] ☒ 0
Genre >> People [100%] ☒ 0

Figure 2. Classification screen of the Facet System.



MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket.

[To zoom, click on the image.]

Title: MARY MCLEOD BETHUNE, half-length, wearing white blouse and jacket.

Courtesy of http://www.loc.gov/rr/print/list/083_afr.html

Tags: [add tags]

Time Period >> 1900 - 1999 [100%] ☒ 2
Event >> Civil Rights [100%] ☒ 0
Location >> South Carolina [100%] ☒ 0
Genre >> People [100%] ☒ 0
Date >> Before 1990 ☒
Location >> South Carolina ☒
Image Size >> Smaller Than 4x6 ☒
Image Format >> JPEG ☒
Source >> 20th Century African American Activists ☒

Figure 3. Personal and Global schemas in the Facet System.

While one of the users' favorite features based on user evaluation, the History feature requires significant database storage. Just as data has a time dimension in a data warehouse, the metadata has a time dimension in the Facet database. Therefore we refer to the metadata storage as a Metadata warehouse. In Fig. 2 and Fig. 3, clicking on the Calendar icon will allow users to see the schema at a given point in time. The History feature shows users the evolution of the schema and the trends in the community. Without the History feature, users may have difficulty finding documents at the "place" where they used to be, as the schemas evolve over time [10].

III. Evaluation and Scaling Issue

We have evaluated the Facet System for over a year with over 300 students at the Old Dominion University and the University of Delaware, and will continue the evaluation in collaboration with additional universities. We have developed image harvesting programs that can ingest thousands of public domain images per day on a given topic [11]. We have tested the system by simulating a large number of users. The scaling issue proves to be a critical factor in expanding the evaluation and deploying our system for public use in a multimedia document repository.

Traditional computing cannot support ever-increasing number of users, associated personal schemas, schema history logging, schema enrichment, and automated classification processes. With traditional computing, resources are typically configured rigidly with respect to both hardware and software (including licenses) to handle expected usage for a fairly short time horizon. Scaling up the configuration is a non-trivial effort and in many case literally impossible within the infrastructure of the supporting organization due to interdependencies. Cloud computing, on the other hand, enables our system to scale to virtually unlimited number of users. Enabling a collaborative document organization system to support virtually unlimited users may lead to a breakthrough in the way that electronic documents are organized. Our long-term vision is that this cloud-based document-organization approach may go beyond organizing an online multimedia collection to organizing knowledge bases in a large enterprise or a global research community. Users can store and organize documents in a computing cloud instead of on their desktop computers or departmental file servers. Besides maintaining virtual, personal "file systems", users can collaboratively evolve community-wide document collections. The cloud not only eliminates the storage limitation of desktop computers and traditional file servers, but also reduces duplicate storage and allows for value-added services such as document version controls (as in Microsoft SharePoint).

We have explored Microsoft Windows Azure and the development fabric, and see great potential of deploying the Facet system on Windows Azure. As the primary purpose of our Facet system is to organize documents including multimedia, we feel that the Microsoft platform will facilitate adoption of the Facet system by research communities, industry enterprises and the society. While the cloud computing infrastructure has been developed for several years, social and organizational support of cloud computing paradigms is still lacking. Deploying our system on Microsoft Windows Azure will open the door of evaluating the system to large enterprise environments in addition to online communities. Using Microsoft Windows Azure explores the potential of integrating our system with the Azure cloud storage, an upcoming popular choice for enterprise file sharing, and Microsoft SharePoint service, a dominating player in today's document management market.

In the following section, we present the preliminary design of how our system is being deployed on the Microsoft Windows Azure cloud. We believe that migrating our system from an open-source stack to Microsoft Windows Azure will provide valuable lessons to other similar efforts in the research community.

IV. Cloud Deployment Architecture

Our system is currently implemented as an open source extension of Joomla [15], a popular content management system built on LAMP stack: Linux [16], Apache[17], MySQL[18], and PHP[19]. We are now deploying the system along with PHP and MySQL on Windows Azure. To achieve scalability, we are moving metadata storage from MySQL to SQL Azure, and document storage from the Web server's file system to Windows Azure Storage. To minimize code changes and demonstrate the modularity of our system, we are continuing to use MySQL to support the core user interface features provided by Joomla!, such as authentication and menu management. Our system has a modular design to facilitate integration with other content management systems. If time permits, we will explore the potential of integrating our system with Microsoft SharePoint services on the Azure platform. Such integration would eliminate the need for MySQL.

The virtual machines in Windows Azure take either Web roles or Worker roles. The Web Role instances run our user-facing Facet System, which is programmed in PHP. The Worker Role instances run the MySQL database and back-end schema enrichment and classification programs in Java. Both Web Role and Worker Role instances connect to SQL Azure to access the metadata warehouse, and utilize Windows Azure Storage for multimedia file storage. Fig. 4 provides a general overview of the virtual machines (VMs) in the Azure cloud, adapted from a Microsoft white paper [12]. Windows Azure Storage and SQL Azure are part of the cloud utilized by the Web Role and Worker Role instances.

In our deployment there are different Web Roles and Work Roles. There are two Web Roles: FacetUI and FacetAdmin. The FacetUI instances serve the end-user interface to the Facet system. The FacetAdmin role contains administrative tools (such as PHPMyAdmin) that administer the database and caches. There are three Work Roles: MySQL, MemCached, and FacetBackend. The MySQL instances host the MySQL database that supports user authentication, menu management and other core-Joomla features. Master-slave configuration of MySQL databases

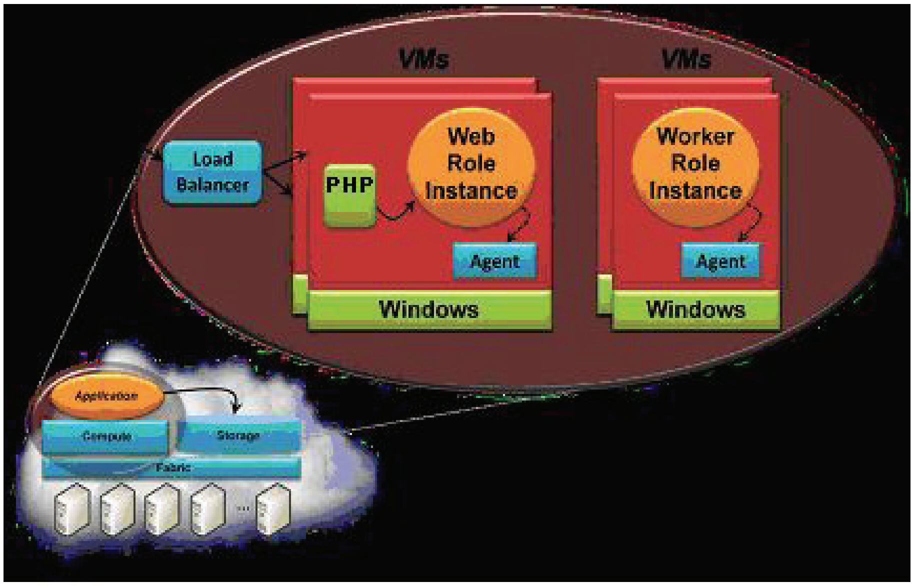


Figure 4. Overview of the Azure Cloud where the Facet System will be deployed.

ISBN: 978-1-61208-105-2

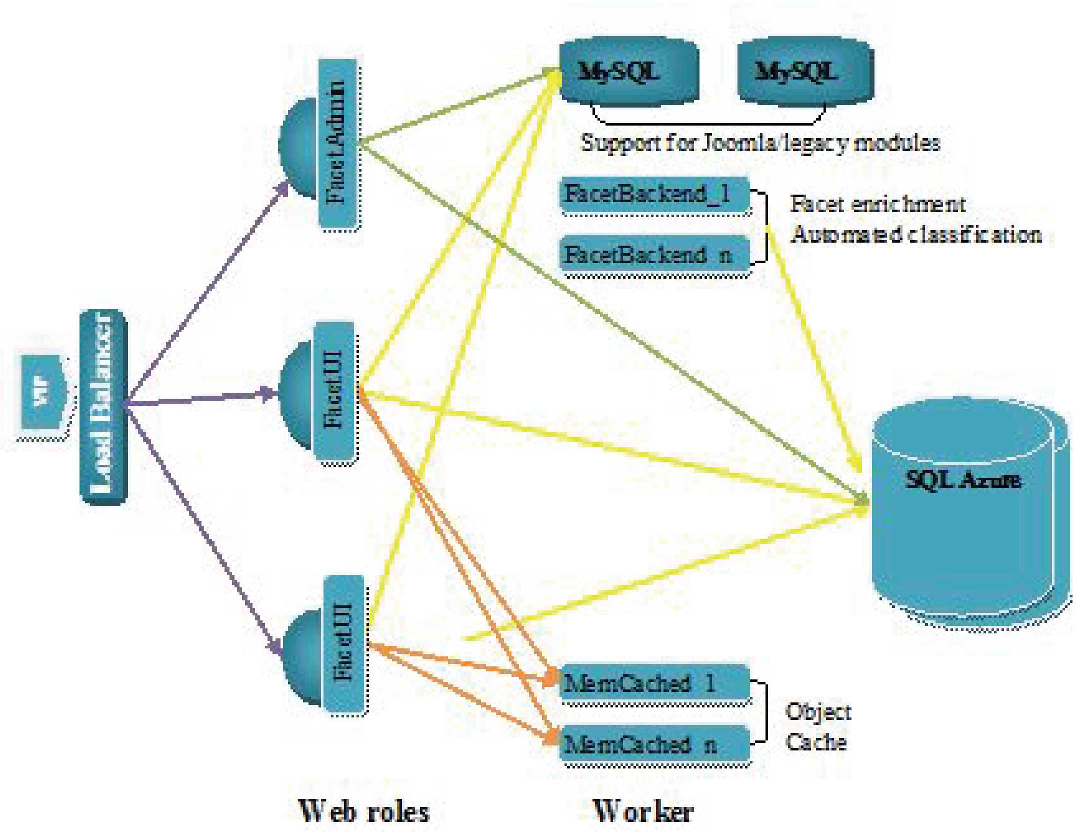


Figure 5. Architecture of proposed deployment on Windows Azure.

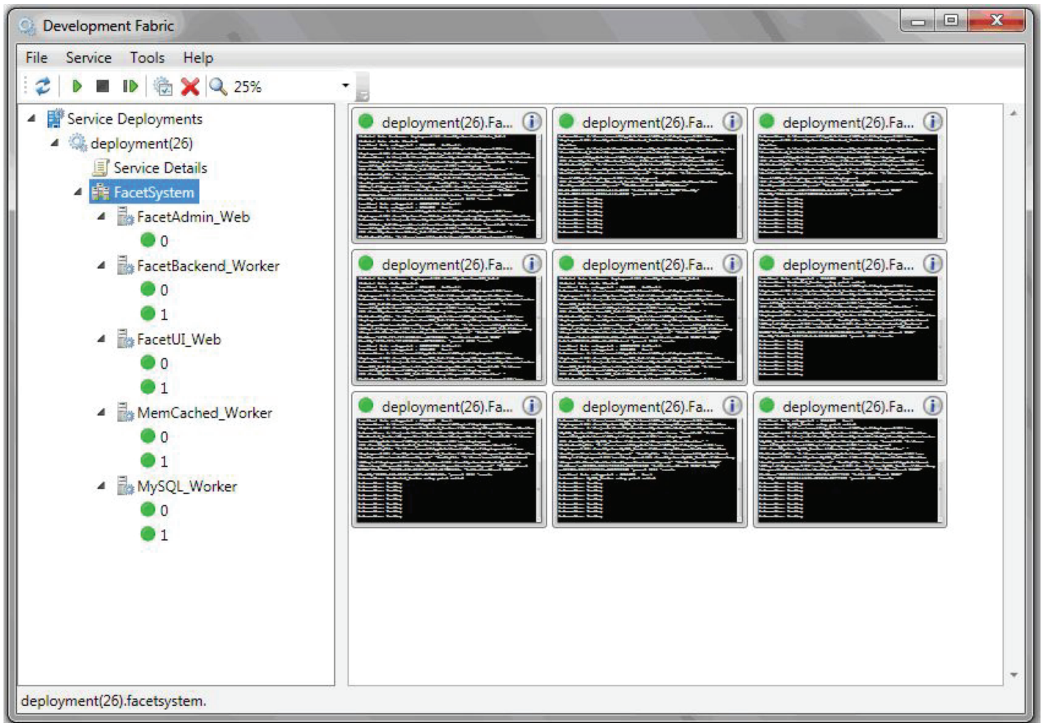


Figure 6. Deployment of Facet System on Azure Development

provides added reliability. The MemCached instances host Memcached, a popular distributed object cache system. A FacetUI instance attempts to read data from the object cache first. If the object does not exist in the cache, the FacetUI instance reads from the database, and load the object into the cache. Many common user requests, such as request to common menu items and the global metadata, are being served by Memcached to lower the database load and improve the performance. The FacetBackend role contains systematic schema enrichment and automated classification algorithms, which operate with data stored in SQL Azure. Deployment of these Web and Worker Roles will utilize Microsoft Windows Azure Solution Accelerators for PHP, MySQL, and Memcached [13]. Fig. 5 shows the architecture for the cloud deployment.

We spent limited concept-proving efforts in exploring the Azure development fabric. Fig. 6 shows the deployment fabric configured with different Web and Worker instances.

V. FUTURE WORK

On the user-oriented side of the system, we will address issues that come with the large scale, such as how to manage a large number of personal schemas and historic views of metadata. Among other adjustments, the user interface needs to be made stateless for load balancing and cloud deployment. On the back-end, we will address scalability issues of schema enrichment (mainly clustering and association mining) and automated classification (current implemented using Support Vector Machine) algorithms. We will evaluate various aspects of system functionality, including both user interface and backend algorithms. In parallel with code changes, we will develop a large test bed (an expansion of the current collection of African-American history images) that allows us to test the scalability of the system. As we gain experience with the large-scale deployment on Microsoft Windows Azure, we will explore design alternatives and make improvements to the system architecture, user interface and backend algorithms.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0713290.

REFERENCES

[1] G. Arnaout, K. Maly, M. Mektesheva, H. Wu, and M. Zubair, “Exploring Historical Image Collections with Collaborative Faceted Classification,” Proceedings of the Digital Humanities 2008, pp. 44-47, Oulu, Finland, June 2008.

[2] H. Wu, K. Maly, and M. Zubair, “Collaborative Faceted Classification System,” the 17th Annual Workshop on Information Technologies and Systems (WITS’07), Montreal, Canada, December 2007, pp. 237-248.

[3] H. Wu, M. Zubair, and K. Maly, “Collaborative Classification of Growing Collections with Evolving Facets,” Proceedings of the ACM 17th Conference on Hypertext and Hypermedia (Hyptertext’07), pp.167-170, Manchester, UK, September 2007.

[4] K. Maly, H. Wu, M. Zubair, and V. Antonov, “Automated Support for a Collaborative System to Organize a Collection using Facets,“ 13th International Conference on Electronic Publishing, Milan, Italy, June 2009, pp. 187-203.

[5] H. Wu, M. Zubair, and K. Maly, “Harvesting Social Knowledge from Folksonomies,” Proceedings of the ACM 17th Conference on Hypertext and Hypermedia, Odense, Demark, 2006, pp. 110-115.

[6] H. Wu, K. Maly, and M. Zubair, “Maintaining and Evolving a Taxonomy with Social Tagging,” INFORMS, Washington, D.C., 2008.

[7] H. Wu, K. Maly, and M. Zubair, “Supporting Multi-Criteria Decision Making through Collaborative Faceted Classification,” 20th International Conference on Multi Criteria Decision Making, Chengdu, China, 2009, CD-ROM.

[8] K. Maly, M. Zubair, and H. Wu, “A Collaborative Faceted Categorization System – User Interactions,” 14th International Conference on Electronic Publishing, Helsinki, Finland, 2010, CD-ROM.

[9] H. Wu and M.D. Gordon, “From Social Tagging to Social Hierarchies: Sharing Deeper Structural Knowledge in Web 2.0,” Communications of the Association for Information Systems Vol. 24, Article 45, 2009.

[10] H. Wu and M. Gordon, “Collaborative Structuring: Organizing Document Repositories Effectively and Efficiently,” Communications of the ACM 50, July 2007, pp. 86-91.

[11] L. Fu, K. Maly, M. Zubair, and H. Wu, “Building Dynamic Image Collections from Internet,” Digital Humanities 2010, 2010, CD-ROM.

[12] D. Chappell, “Introducing the Windows Azure Platform,” http://www.microsoft.com/windowsazure/whitepapers/, 2009.

[13] M. Srivastava, and T. Shanbhag, “Developing PHP and MySQL Applications with Windows Azure,” Microsoft Professional Developers Conference, 2009.

[14] Microsoft Windows Azure platform. http://www.microsoft.com/windowsazure (Last accessed 2010)

[15] Joomla. www.joomla.org (Last accessed 08/16/2010)

[16] Linux. http://www.linux.org/ (Last accessed 08/16/2010)

[17] Apache. http://www.apache.org/ (Last accessed 08/16/2010)

[18] MySQL. http://www.mysql.com/ (Last accessed 08/16/2010)

[19] PHP. http://www.php.net/ (Last accessed 08/16/2010)

Security Service for the Rollout of Security Credentials in Ubiquitous Industrial Automation Environments

Rainer Falk, Steffen Fries
Siemens AG
Corporate Technology
Germany
{rainer.falk; steffen.fries}@siemens.com

Abstract— **Industrial control networks, e.g., for factory, process or energy automation and smart metering, are increasingly based on IT communication technologies like Ethernet, IP, and Web-Services. Security measures as authentication or cryptographic VPNs are used to protect communication links to supervising control stations and for remote service. While standard communication technologies have been used at the supervision level for some time, they will increasingly be used down to the field level comprising a huge number of field level control devices, sensors and actuators. These may be installed in largely distributed, uncontrolled areas. IT security measures are needed to protect the exchange of control commands and monitoring information between these field level devices and towards control stations. The efficient and secure distribution of required security credentials is challenging This paper describes a service for managing security credentials for ubiquitous field level devices (sensors, actuators) in an industrial automation environment.**

Keywords – Ubiquitous Security, Industrial Communication, Energy Automation, Sensor Actuator Network

I. INTRODUCTION

Standard communication technologies as Ethernet, the IP protocol, and Web-Services are increasingly used in industrial environments such as automation systems for energy distribution, building, factory and process automation, or for smart metering. This trend will extend down the automation pyramid to field level devices including even individual sensors and actuators. These numerous field level devices being widely distributed form an ubiquitous automation environment. Integrated security mechanisms have to be supported by a huge number of pervasive devices. Extremely easy commissioning and integrated security functionality are required to make the technology suitable for industrial applications. Automated Plug&Work mechanisms especially supporting security configuration are needed also to support agile automation concepts in which the production environment is flexibly adapted to changing needs. Moreover, security configuration has to take into account that automation environments may be geographically far-flung.

This paper describes challenges, side conditions and approaches for a security service enabling the efficient rollout of security credentials in ubiquitous industrial automation environments. This service comprises technical as well as organizational means. It allows field level components to be configured with the required set of security parameters to protect the device itself and its communication. In particular the pre-configuration of security credentials during the manufacturing process is considered as one way of supporting a secure configuration as part of device installation.

The remainder of this paper is structured as follows. Section II provides a motivation for security configuration processes based on existing security applications in automation networks. Section III describes a security service for the rollout of security credentials covering the whole security parameter lifecycle, which is the discussed in the context of the product life cycle of ubiquitous industrial field devices. Section IV afterwards describes different supported approaches for key distribution, applicable to industrial environments, while Section V describes an exemplary setup of the security service, where one service instantiation is used during device manufacturing, and a second one during device installation and operation. Section VI summarizes the findings and gives an outlook to future work.

II. AUTOMATION SYSTEMS SECURITY

Typical automation systems are built in a hierarchical way as shown in Figure 1. It shows typical layers of an automation pyramid. On the lowest level there are sensors and actors that are connected to field devices. Specialized field buses are expected to be increasingly replaced by standard communication technology as Ethernet and IP. These field devices are actuated by controllers, e.g., a programmable logic controller PLC, which may be interconnected using industrial real-time

Ethernet protocols as, e.g., ProfiNet (cf. [10]). On the top are interconnections to supervisory systems and enterprise resource planning systems.

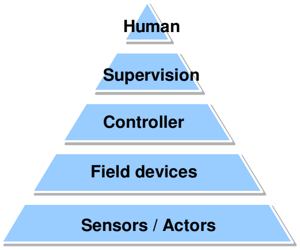


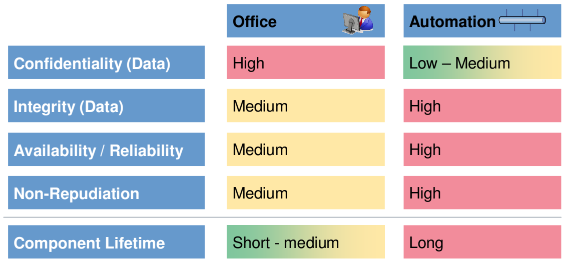
Figure 1: Automation Pyramid

Today, security in industrial automation is commonly applied by isolating cells on controller level using security gateways. Internally, the automation network is assumed to be closed and communication is not further cryptographically secured. In the future it is expected that devices down the automation pyramid including sensors and actors will feature integrated security functionality. This has already been discussed in the context of funded projects like the European funded project "Virtual Automation Networks – VAN" (cf. [1]) enabling secured communication between automation cells or devices in automation cells of different production lines. In contrast, energy automation already uses IP connectivity down to the field level.

Security mechanisms to be supported in automation communication comprise well-known security services:

- **Authentication**: The property that the claimed identity of an entity is correct.
- **Authorization**: The process of giving someone permission to do or have something.
- **Integrity**: The property that information has not been altered in an unauthorized manner.
- **Confidentiality**: The property that information is not made available or disclosed to unauthorized individuals, entities or processes.
- **Availability**: The degree to which a component is operable (non-cryptographically service).

In contrast to office networks, automation networks have different requirements to security services as shown in the Figure 2.



	Office	Automation
Confidentiality (Data)	High	Low – Medium
Integrity (Data)	Medium	High
Availability / Reliability	Medium	High
Non-Repudiation	Medium	High
Component Lifetime	Short - medium	Long

Figure 2: Comparison Office/Industrial Security Requirements

The determination of security needs reveals the high importance of integrity and availability within automation networks. Also non-repudiation is often important so that e.g. reliable information about production is available or to provide billing-relevant information that can be relied upon. These security needs are quite different to typical priorities in office networks, see Figure 2.. A particular design consideration is the long component life time (several decades, depending on industry). All of the stated security services, independent of the application area, have one in common. They all need some type of security credential (which may be symmetrical or asymmetrical), where they can build upon. Thus, the process to efficiently install required security credentials on a huge number of devices will provide a big challenge. Its solution is a prerequisite to the successful adoption of integrated security mechanisms.

To better motivate the need for security credentials or more generally security parameters, the following subsections outline concrete examples for security in automation communication.

A. *Example 1: Energy Automation*

IEC 61850 provides a standard for communication in the domain of energy automation. It addresses the data exchange on process level, field level, and station level. Today, IEC 61850 is mainly used for reporting status and sampled value information from Intelligent Electronic Devices (IED) to a substation automation controller as well as for command transport from a substation automation controller to IEDs. It also covers the communication between IEDs instead of dedicated wires.

The following Figure 3 gives an example for a typical IEC 61850 communication scenario.

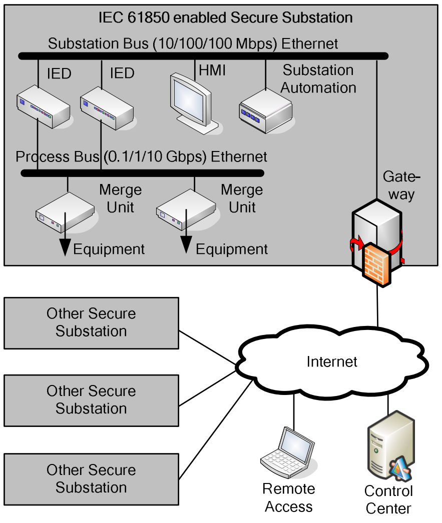


Figure 3: Typical Energy Automation Scenario

This scenario obviously requires security services to protect the communicated control data. For energy automation the necessary security services are defined in IEC 62351. It defines explicit measures for communication using TCP and also serial protocols which are used directly in substation automation as well as in adjacent communication supporting energy automation, e.g., inter-control center communication. IEC 62351 addresses the general security requirements stated above. Currently the standard comprises eight parts that are in different state of completion.

While part 1 and 2 are more general and provide information about considered threat scenarios and the definition of terms, part 3 to 6 are directly related to energy automation protocols like IEC 61850 (IEC 62351 Part 6) and IEC 60870-5-x (IEC 62351 Part 5) and their mappings to protocols like TCP/IP (IEC 62351 Part 3) and MMS (IEC 62351 Part 4). These parts utilize symmetric as well as asymmetric cryptographic functions to secure the payload and the communication link. Moreover, the existing security protocol Transport Layer Security (TLS), which has been successfully used in other technical areas and industrial applications, is directly applied. Here, IEC 62351 specifies cipher suites (the allowed combination of authentication, integrity protection and encryption algorithms) and also states requirements to the certificates to be used with TLS.

Besides TCP/IP, IEC 62351 Part 5 relates to the specialties of serial communication. Here security measures are defined to especially protect the integrity of the connections based on pre-shared keys. This part also specifies the key management necessary for the security measures.

IEC 62351 Part 7 describes security related data objects for end-to-end network and system management and also security problem detection. These data objects support the secure control of dedicated parts of the energy automation network.

Part 8 of the standard is currently in definition and addresses the integration of role-based access control mechanisms into the whole domain of power systems based on ID-certificates, attribute certificates, or software tokens. This is necessary as in protection systems and in control centers authorization as well as stringent traceability is required. One usage example is the verification of the authorization and accomplishment of a dedicated switching action.

As it can be seen from the description above, IEC 62351 utilizes security credentials, e.g., in the context of the transport layer (using TLS or serial communication) but also on application layer for role-based access control. Crucial to the application of security credentials is the general credential handling comprising generation, provisioning, revocation, and especially the initial distribution to all participating entities. This is currently underspecified, but has been acknowledged by standardization as important. As the standard is extensible, it is expected that there will be a new part, describing credential handling in the context of IEC 62351 services.

B. *Example: Wireless Sensor Networks*

Wireless sensor networks consist of sensors (and actors) that communicate using short range wireless communication based on 802.15.4, Bluetooth, ZigBee or wireless HART. Important industrial use cases are machine and plant monitoring, asset tracking, and metering [8]. As the wireless communication can easily be intercepted and manipulated, a cryptographic protection is a must. Therefore, the sensors/actuator nodes have to be configured with a join key that allows to securely join a wireless network and to set-up required security associations. The join key is typically a secret key that is used to authenticate towards a security manager. The security manager authenticates the nodes and provides required session keys. The join keys are configured when the sensor network is installed, but it would also be possible to provide sensor nodes that have been pre-configured during manufacturing.

C. *Example: Product Authentication*

To identify products, in particular replacement parts, and to verify the claimed identity, electronic authentication mechanisms can be integrated directly into the components. This allows an automation system to automatically identify installed components and verify whether they are genuine (anti counterfeiting). Further information can be stored along the product life cycle [9]. An electronic authentication module being part of the product provides a cryptographic authentication function.

III. SECURITY SERVICE FOR THE CREDENTIAL ROLLOUT

The rollout of security credentials describes the process of the initial setup of security credentials (e.g., keys, certificates) and related configuration information (permissions, policies). The result is a trust anchor enabling the further deployment of configuration information, services and communication.

The main functionalities of the security service for the credential roll-out for ubiquitous industrial field level devices are:

- Credential generation, certification, and archival,
- Credential distribution to field level devices,
- Credential life cycle management.

The security service can be adapted to different application-scenario-specific requirements through configurable policies. This ensures that the credential management is compliant with relevant requirements of the automation operator. The security service is exposed

ISBN: 978-1-61208-105-2

towards a user as for example a worker installing a field level or an employee in the field level device manufacture only in a way that ensures that the security service is used easily while ensuring compliance with defined security policies.

Security credentials are, like other type of data or equipment, part of a lifecycle. They are created, applied, and destroyed and need to satisfy a certain security policy. The typical life cycle of security credentials is depicted in Figure 4.

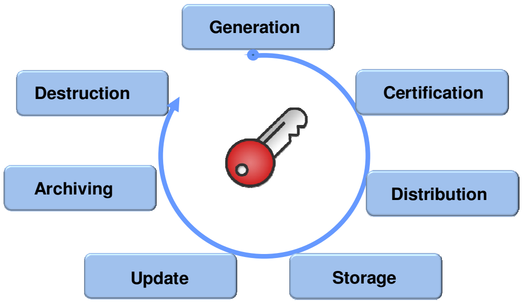


Figure 4: Security Parameter Life Cycle

The following list explains the single stages in the life cycle that are realized by the credential rollout service:

– **Key Generation**: Device keys can be created on the device itself. For example, in case of asymmetric key pairs, the device may generate the key material and a Certificate Signing Request (CSR), which is sent to a Certification Authority. Alternatively, keys may be created externally (e.g., a trust center, an engineering station, or an administrators laptop) and installed on the target device (off-device key generation).

– **Key Certification**: Typically done for asymmetric keys through a certificate authority. Depending on the key generation, this can be part of the key generation in a trust center or may be done on information sent in a CSR.

– **Key Distribution**: In case of off-device key generation, the device key has to be installed on the target device. This can be performed offline, e.g. the key is installed to the target entity during a manufacturing step, or online requiring communication with a security server (out-of band using a separate communication channel or in-band as part of a service communication).

– **Key Storage**: The private/secret device key can be stored in secured memory (e.g., flash) or in a separate hardware module (e.g., smart card or a trusted platform module).

– **Key Update**: Session key update does not belong to the describe process of security parameter rollout as it is typically performed by the security protocol used, based on a given security policy. Cryptographic keys have a dedicated lifetime, e.g., user certificates typically have a lifetime of 2 years, while server certificates are limited to 1 year.

– **Key Archiving**: Typically long term (secure or private) keys are archived to enable access to encrypted data. A use case is given by an employee leaving a company. While encryption keys are archived, signature/authentication keys need not to be archived as it is sufficient to archive corresponding public key / certificate.

– **Key Destruction**: Session keys are destructed (deleted) as soon as the session has ended. Long term keys are deleted, after keys have been renewed. This can be the case after the lifetime of the key has ended regularly, or if the key has been compromised.

The security parameter lifecycle has to be aligned with the product lifecycle, whereas the product may be a single component or a complete automation system.

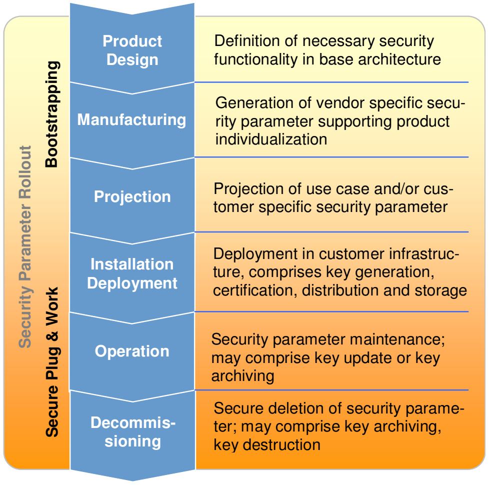

Security Parameter Rollout
Bootstrapping
Secure Plug & Work
Product Design
Definition of necessary security functionality in base architecture
Manufacturing
Generation of vendor specific security parameter supporting product individualization
Projection
Projection of use case and/or customer specific security parameter
Installation Deployment
Deployment in customer infrastructure, comprises key generation, certification, distribution and storage
Operation
Security parameter maintenance; may comprise key update or key archiving
Decommissioning
Secure deletion of security parameter; may comprise key archiving, key destruction

Figure 5: Security Parameter Rollout in the Product Life Cycle

Figure 5 shows main phases along the product life cycle:

– **Manufacturing**: This phase concerns the production of the hardware, possibly including a dedicated hardware security module, and the programming of the flash memory by the manufacturer. The product is individualized during manufacturing by programming a fixed device identifier (e.g. serial number).

– **Project Planning (Projection)**: During the project planning phase, a certain industrial installation is planned. The used components, their (security) configuration and interconnection are defined.

- **Installation**: During the installation phase, the equipment is physically installed, configured and tested according to the project plan.
- **Operation**: During the operation phase, the devices are in regular operation mode. It is interrupted by maintenance and repair phases, which may comprise security parameter updates due to the normal key management lifecycle or complete device exchange.
- **Decommissioning**: Finally, the devices are decommissioned, i.e., they are put out of operation. Installed security parameter need to be deleted before leaving the customer premises.

As shown, security parameters are used in different phases of the product lifetime and are applied as:

- short term or session parameter (e.g., for integrity or confidentiality protection of an administrative action)
- long term or permanent parameter (e.g., for authentication)

Besides the pure key material, an efficient and secure solution requires well defined organizational processes for the life-time management of security credentials, standardized (and preferably certified) software and hardware components, a protected environment, etc.

It is useful to distinguish different phases of the security rollout:

- **Bootstrapping** concerns device-specific credentials installed as part of manufacturing. These are not bound to a specific usage environment of the product but may be used as a trust anchor for the next phase.
- **Secure Plug&Work** describes the process of installing a device in its intended usage environment. The installed credentials are specific to this usage as defined by the project plans.

These different phases pose quite different requirements on the handling of security credentials:

- During the bootstrapping phase, a manufacturer creates and installs credentials for a huge number of devices in a uniform way. The challenge is to define processes that allow handling the huge number of security credentials cost-efficiently, in a uniform way. This comprises the in-factory handling and also the distribution of the device connected parameter to the end customer.
- During Plug&Work the installation personal has to be supported so that they can install and commission devices very easily according to the project planning documentation. Here an individual device has to be brought to operation. As a huge number of devices have to be installed in a typical industrial plant, it is important to limit the effort to install a single device while configuring it according to its role in the project plans.

IV. Key Generation and Distribution

The security service for the rollout of security credentials is an important functionality to create, distribute and manage credentials for ubiquitous industrial field level devices. The differences of different application field require some flexibility concerning the deployment and operation of the security service. This Section describes supported options to distribute cryptographic keys to target field level devices..

Cryptographic keys may be generated by the security service itself, e.g. within a trusted hardware security module including a physical random generator. The created keys are then installed on the target device. This has to happen in a secure environment, e.g. a manufacturing plant. Alternatively, the secret/private keys are created on the field level device itself and the security service certifies the public key by a digital certificate.

The rollout of security parameter may be distinguished based on the credential distribution methods into:

- Offline parameter distribution
- In-band parameter distribution
- Out-of-band parameter distribution

None of the stated methods does necessarily require a cryptographic key already in place to support the bootstrapping. Obviously, there are technical variations for each of the categories. The following list provides a short characterization of the method and also provides some examples for each category:

- **Offline parameter distribution:** Performed using dedicated engineering tools directly connected to the device or via a separate network before the device is brought to operation (see Figure 6).

  This requires a (mobile or fixed) engineering station in the offline network having all parameter sets for the devices to be bootstrapped available. Besides the example given in Figure 6 another approach is the application of a token to transport the cryptographic parameter to the target device. This approach is supported for instance in the setup of common WLAN routers using Wi-Fi simple configuration. A further example is given by application of SIM (Subscriber Identity Modules) cards in mobile devices, were the SIM card, carrying all necessary security parameter, can be distributed independently of the actual mobile device.

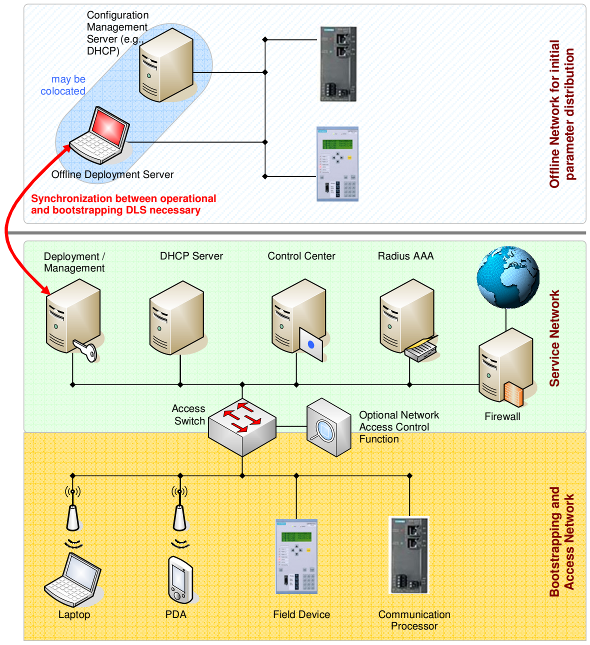

Configuration Management Server (e.g. DHCP)
may be colocated
Offline Deployment Server
Synchronization between operational and bootstrapping DLS necessary
Offline Network for initial parameter distribution
Deployment / Management
DHCP Server
Control Center
Radius AAA
Access Switch
Optional Network Access Control Function
Firewall
Service Network
Laptop
PDA
Field Device
Communication Processor
Bootstrapping and Access Network

Figure 6: Offline Key Distribution

– **Out-of-band parameter distribution:** A separate logical or physical communication channel used to configure security parameter (see Figure 7). It basically resembles the offline distribution approach using an online connection instead of a separate physical network. As stated before, devices may already possess a cryptographic credential, which can be provided by the device manufacturer.

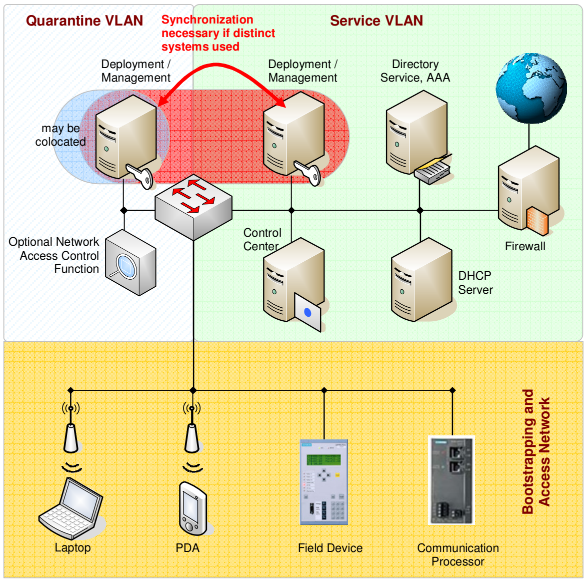


Figure 7: Out-of-Band Key Distribution

Figure 7 shows the application of a quarantine VLAN for the distribution of security credentials. This can be compared with today's methods of Network Access Control (NAC) by putting the connecting device into a dedicated logical environment to check it's compatibility to a local security policy before providing access to the Intranet. The security parameter bootstrapping may even be combined with this functionality.

– **In-band parameter distribution:** Distribution using the same communication channels as used during regular operation (see Figure 8). This may be based on a pre-configured device identifier (like the MAC address), manufacturer installed security credentials or even a liaison device.

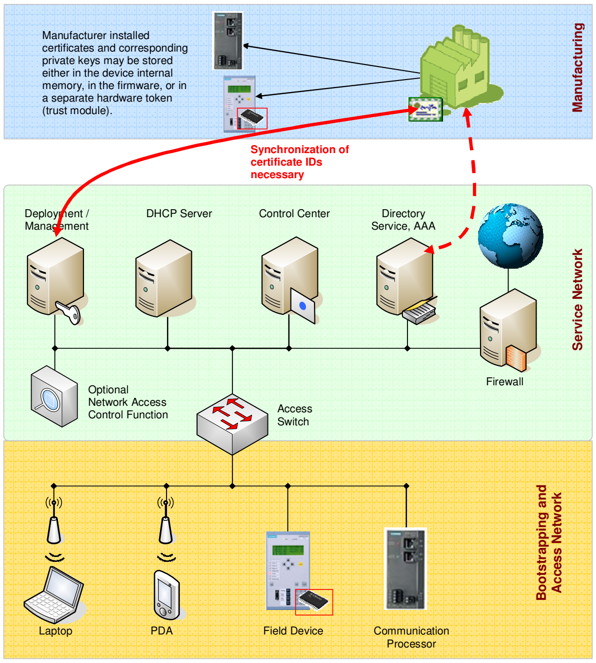

Manufacturer installed certificates and corresponding private keys may be stored either in the device internal memory, in the firmware, or in a separate hardware token (trust module).
Manufacturing
Synchronization of certificate IDs necessary
Deployment / Management
DHCP Server
Control Center
Directory Service, AAA
Service Network
Optional Network Access Control Function
Access Switch
Firewall
Laptop
PDA
Field Device
Communication Processor
Bootstrapping and Access Network

Figure 8: In-Band Key Distribution

Figure 8 shows an example using manufacturer installed security credentials to protect the distribution of customer specific key material.

A further variant of in-band parameter distribution is supported by the application of a liaison device, which is already in possession of a service technician. Here, the security credentials on the liaison device can be "borrowed" for the bootstrapping of the target security parameter by using near-field communication, an approach that is currently being standardized by 3GPP.

V. Use Case Example

The designed security service for the rollout of security credentials to ubiquitous industrial field level devices provides the flexibility to be adapted to different requirements. This Section describes a preferred variant based on device authentication credentials preinstalled during manufacturing. These allow the device to be identified and authenticated in its respective target environment. This secure device authentication is the trust basis for an automated bootstrapping of credentials within the target installation environment.

The following requirements respectively. side conditions of a typical industrial environment are respected:

– Before field level devices are installed, a detailed projection plan is defined. The projection plan defines the configuration for each component of the automation system. This information is useful as information in the expected devices and their interconnection needs to be available before the actual installation is performed.

– It should be possible that the installation is performed by personal not having an IT or even a security background.

– It must be possible that the correct installation according to the projection documents is proven to support a security audit trail.

The designed security service for the credential management for ubiquitous industrial field level devices achieves these objectives by the following design: When a field level device establishes network connectivity within the target environment for the first time, it authenticates towards a bootstrapping service using pre-installed device authentication credentials. The bootstrapping service checks whether the device is authorized and provides the target device configuration based on automation plant project planning data.

Devices are pre-configured by the manufacturer with a unique device key. This key is certified by a digital certificate. It allows installation personal to work only with device types and serial numbers, while not being exposed to cryptographic keys or certificates. The device manufacturer uses a corresponding security service that issues and manages device authentication credentials during manufacture that are valid for the product life time, independently on where the device is installed. The private device keys are created in a batch process to be installed within the manufacturing environment. The corresponding device certificate including the public device key is archived.

During installation, the device can be unambiguously and securely identified using the pre-configured device key. Using this initial device key, an installation-specific (customer-specific) device key is deployed in-band. The device and its configuration are registered in a configuration database. The automation system owner uses a second instantiation of the security service that issues and manages device credentials valid within the respective automation environment. The credentials are valid within the specific installation environment. The device keys can be, depending on respective policy, created on the field level device itself, or they are created during projection phase and installed on the respective target field level device during the setup. Optionally, after installation has been completed and the automation system is turned to operation, an automatic key update can be performed, so that the keys used during operation are not known by the installation personal. This re-keying is supported by the second security service. Similarly, such a key update can be performed as part of service, so that service personal does not get access to keys used during operation of the automation system.

VI. Conclusion and Outlook

While the problem of key distribution is as old as IT security, the increasing introduction of security mechanisms in industrial environments requires solutions that are adapted to the specific application field. This paper presented a security service for the rollout of security credentials to ubiquitous industrial field level devices. Both technical and organizational requirements have been described.

Currently, work is ongoing in the context of the European funded research project IoT@Work (Internet of Things at Work), service is worked out in more detail and is validated by prototypes.

References

[1] Homepage EU Project Virtual Automation Networks, http://www.van-eu.eu/

[2] RFC 5246: The Transport Layer Security (TLS) Protocol, Version 1.2, T. Dierks, E Rescorla, August 2008

[3] ISO-IEC 61850, Part 1: Introduction and Overview, May 2003

[4] ISO-IEC 61850, Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3, May 2004

[5] ISO-IEC 62351, Part 4: Communication Network and System Security – Profiles Including MMS, October 2006

[6] ISO-IEC 62351, Part 5: Security for IEC 60870 and Derivatives, February 2007

[7] ISO-IEC 62351, Part 6: Security for IEC 61850, October 2006

[8] Rainer Falk, Hans-Joachim Hof, Ulrike Meyer, Christoph Niedermeier, Rudolf Sollacher, and Norbert Vicari: "From Academia to the Field: Wireless Sensor Networks for Industrial Use", 7th GI/ITG KuVS Fachgespräch „Drahtlose Sensornetze", Berlin, 25-26 Sep. 2008.

[9] Rainer Falk, Andreas Koepf, Hermann Seuschek, and Ming-Yuh Huang, Mingyan Li: Simulating a Multi-Domain RFID System for Replacement Part Tracking, Third International Conference on Emerging Security Information, Systems and Technologies SECURWARE 2009, Athens/Glyfada, Greece, 18-23 June 2009.

[10] Profinet, http://www.profibus.com/technology/profinet/

Integrated e-Services in Public Sector

Seppo J. Sirkemaa
28120 Pori, Finland
seppo.johan@gmail.com

Abstract— **The use of Internet is increasing in all areas. It is still noteworthy that development of electronic services in public sector has not been as rapid as in the area of e-business. From citizens point of view it would be important to allow access to services and public decision making through the Internet. Electronic services have significant potential, but they may also transform structures in public organizations. Here, we look at the development of electronic services in public sector organizations. In the public sector these services are called e-Services when they are made accessible to the users through Internet. The development of e-Services face a number of challenges – it is not an easy to realize the potential of technology. Could experiences from e-business development be used in public sector? The potential benefits are almost identical, however in public sector there are administrative and departmental barriers that need to be crossed in order to create value to the user of electronic services.**

Keywords- e-Services; Internet; development; public sector; e-business

I. INTRODUCTION

Internet makes it possible to provide services in a new way, making it possible to create added value to the user. At the same time organizations may re-organize and streamline their processes. The idea in services that are built on top of Internet is anywhere, anytime – but is this the case in services provided by public sector?

In private sector information technology, Internet-based applications and technologies are used widely in e-business applications. In the public sector electronic services are referred to as e-Services, which relate to services that public organizations provide. The term e-Services is further defined as interactive, content-centered services are accessed through the Internet [19], [20].

It has been noted that public sector needs to move from paper to electronic correspondence, and from this toward a self-service model where citizens can get the answers and make transactions through the Internet [1]. However, in public sector the goal is not only to move forms and services provided by different departments to Internet, it is more a question of developing one-stop government solutions [13], [8].

A. Goals and structure of the paper

In this paper, we look at development of electronic services to users of public information systems. The focus is on the challenges facing development of electronic services in public sector organizations face. It is an environment which calls for cooperation of various departments and functions, and interaction between service providers, experts and other stakeholders. The question of interest is what makes providing electronic services in public sector so different from development of e-business applications in private companies. Another goal is to better understand why some development projects in the area of electronic services are successful and others fail.

II. ELECTRONIC SERVICES IN PRIVATE SECTOR

The development of electronic services in public sector organizations has been relatively slow [9], [16]. This is interesting, because it seems clear that also public sector would benefit from electronic access to services [27]. It is not surprising that there is pressure and an increasing demand for development of e-Services in public sector. Let us look closer at development of e-business applications provided by companies in order to better understand the existing potential.

Internet is changing the way products and services can be marketed, delivered and purchased. With e-business small- and medium sized companies can compete globally. From the business owners perspective there are several targets when moving activities to the Internet, serving customers on a 24 / 7 –basis and global reach are the most significant issues that prompt the development of e-business applications [20].

The most significant benefits of e-business are connected to transactions and communication [6]. Internet lowers transaction-related costs for both buyers and sellers. Companies can change prices on-line when raw-material costs change, for example. At the same time buyers have access to up-to-date prices directly from their terminals - most online shoppers use comparison-shopping engines [17]. Through Internet we may as consumers gather background information on competing products and services, compare and purchase things without the need to leave home – safe and easy.

In Internet, the concept of service is inextricably linked to e-business applications and the types of services there are in e-business environment [23]. Here, self service is typical; users have learned to help themselves in finding information and buying products. In e-business it is common that customers who make the reservations over the Internet, for example, receive a rebate or discount coupons that they can use when shopping again in the future. The goal is to develop lock-in, and push the customers to using services that are available on the Internet. The customers can do e-shopping whenever it best suites them, they don’t need to wait on phone, for example. At the same time self-service frees staff from answering customer calls to more productive work. The question here is that why could not public sector organizations use similar techniques in order to promote e-Services and ”locking” citizens.

One of the key drivers in e-business is that Internet makes it possible to increase company’s efficiency and effectiveness [20]. Internet allows restructuring of processes which results better profitability - these are important issues for all companies, and are motivators for development in the public sector as well. Even though goals, ethics and values may slightly differ business-like performance measurement and orientation has been evolving in public sector organizations as well [18], [24].

III. ELECTRONIC SERVICES IN PUBLIC SECTOR

There is an almost infinite potential in development of electronic services in the public sector. Typically, services that are provided through the Internet are connected to sharing information. Public sector services are mostly connected to information – and Internet is a very efficient way to gather and share it. We argue here that developers of public sector e-Services should do more than they have done so far. Let us look at the situation in Finland.

A. Case Study: Finland

There is a clear need to develop electronic services in public sector, in municipalities and cities. This is the case in Finnish cities and municipalities which were studied in December 2009 [26]. In this study, altogether 304 key persons from 191 cities and municipalities answered the questionnaire (about 3/4 of all cities and municipalities in Finland). Geographically the answers cover 62 % of the total inhabitants of Finland, and so the results reveal the national situation in the area of electronic services.

In Finland, it is expected that electronic services will bring added value in the public organization in management and increase efficiency in internal processes, and in specific in core processes [26]. Persons who answered believe that there will be more multi-channeled solutions, and integrated services that cross internal organizational borders. Furthermore, it is expected that the burden of administrative work will decrease because of streamlined processes and electronic services. From the citizens perspective electronic services are expected to allow better access to public services. It is likely that the number of available services will increase as the demand is rising. On the other hand, citizens are expected to use access and use electronic services so that investments made in the development make sense.

The current situation in the area of electronic services could be better [26]. It has been noted that there are relatively few e-Services; they are not integrated across administrative offices or even between departments within one single office. There is no common interface, citizen’s login or similar standards at the moment in Finland. Clearly, there is work to be done here.

B. Better integration and services needed

Public administration is full of administrative jargon and official pseudonyms, which are likely to be transferred to web when they have been digitized. The whole structure of the web-sites is based on different agencies, departments and units (stovepipe-structure) rather than integrated portals [1]. In addition, the sites are often relatively unfriendly and there are no comprehensive search-engines that would make it easier to find information from the site, for example. It is often noticed that people send lot of email to public administration. This is because it is easier to ask than try to figure out what agency to contact and how to proceed. In this way poorly designed Internet site can easily increase the burden of the staff in public sector organizations when the amount of incoming email queries go up. This should indicate that existing service through the Internet needs to be developed.

Electronic services should be integrated, enhance self-service and trust so that users see the added value of electronic services. Technologically, users should be able to complete most of their transactions online. Here easy-to-use, robust and trustworthy services are needed so that more users start using e-Services in public sector.

Electronic services need to be developed around user needs. Transferring existing papers, files and information from different agencies into web, and placing some hyperlinks between them is not enough. As Löfstedt [25] noted “..it is about reinventing the way in which governments interact with citizens, governmental agencies, businesses, employees, and other stakeholders.”

IV. CHALLENGES IN DEVELOPMENT

Usually, the development involves cooperation of several people. Especially in development of electronic services like one-stop government services there is a need to combine resources and expertise from different sources. This means that people from various functions, units and locations are brought together, and also outside expertise is needed. Hence, development can be seen as a partnership.

The definition of partnership ranges from working relationships to active transactions and collaboration between organizations. Different types of joint ventures, subcontracting, alliances and acquisitions are included in partnership arrangements. In a partnership actors learn to know each other in the long run. Often relationships are relatively intensive and even personal. In business relationships competence and goodwill are needed for trust to develop [4]. The important issue here is that partnerships are based on commitment to cooperation. In this context the term partnerships includes inter-organizational cooperation – which is needed in development of e-Services [13].

In the public sector, initiatives where services cross departmental boundaries organizational cooperation are a challenge. In most projects there are often external organizations, IT expertise and special skills that are needed.

Cooperation of several partners, units and stakeholders can become a barrier for projects that involve several organizations. Development is often faced with the fact that stakeholders act too independently, because projects tend to be poorly coordinated [12].

The lack of alignment between organizational goals is put forward as a major factor in the set of organizational and managerial challenges. Furthermore, the size of project and the diversity of users and different organizations involved make the development work more demanding. Dawes and Pardo [5] also address the existence of multiple and partially conflicting goals in public sector projects involving several stakeholders. In inter-organizational projects there is a built-in delay as a result of inadequate organizational cooperation [13]. They (ibid.) identify six areas which cause failures and delays in development of electronic services. The first key area is lack of organizational cooperation. The second key area is missing legal regulations and the third is that necessary pre-conditions in regard to technology are not met. The fourth key area is human factors, skills and resources. The last obstacles (or first) in development are result of inadequate funding and political support.

Lack of organizational cooperation
Missing legal regulations
Technological incompatibilies
Staff resources and skills inadequate
Funding inadequate
No political support

Figure 1. Barriers in development of electronic services

Gil-García and Pardo [7] found that challenges to various e-government initiatives are cross disciplinary and may be grouped into five categories: (1) information and data, (2) IT, (3) organizational and managerial, (4) legal and regulatory, and (5) institutional and environmental. Information and data (first category) covers the capturing, management, use, dissemination, and sharing of information. In this category the developers also need to address data quality and data accuracy as well as dynamic, changing information needs. Information technology (second category) refers to issues like technological incompatibility and complexity, security, usability, technical skills and experience, and technological newness which all present challenges for development and use of services. Organizational and managerial issues (third category) are the main challenges to information systems development [7]. It is clear that laws and regulations must be taken into account when developing electronic services (fourth category). The institutional and environmental challenges (last category) are result of the institutional framework in which public organizations. The framework also includes the existing policy environment.

V. FOCUSING ON SERVICES

We have looked at development of electronic services and the challenges that this involves. The focus has been on services in the public sector, and they have been mapped against commercial e-business services. In this context interplay of several units, functions and organizations is needed – especially if the provided services are sophisticated, and providing users one-stop government e-Services [1], [2], [25].

The development of electronic services - or information technology in general - requires connecting technologies and applications in order to provide solutions for users. There is a variety of underlying information infrastructures, applications and services that may be owned, maintained or developed by organizations from private or public sector [2], [22]. Similarly, development of e-Services is a combination of expertise and effort from people in the organization and from external environment.

The types of electronic services vary greatly in public sector. It is natural to expect that services are integrated into processes and information systems of the organization that provides them. However, in public sector organizations it is common that departments and units provide services to citizens rather independently. Departments have different processes and information systems, which are not connected. In many cases information is stored in separate databases. This may be enough when services are oriented to information delivery between the public administration and the citizens. For example, providing downloadable documents and forms is simply offering documents in electronic format and making them accessible through the Internet.

Over time more services are developed, more features are added to existing services, and more enhanced, transactional services are developed [23], [1], [3]. This is challenging as when services become more sophisticated the overall complexity increases. It has been noted that moving to services that are transactional is a big step [10]. Transactional services require connectivity, information in other systems and data-bases needs to be accessed, combined and updated from users' interface through the web. This is challenging from the information systems viewpoint as the situation calls for connecting originally separate systems which may be based on different software and database structures.

Cooperation is a challenge for management of the development of electronic services. It is not uncommon that managers find themselves making decisions about technology for which they are unprepared or even ill-equipped [7]. Successful development of electronic services calls also for top management commitment, linkage to business, technical alignment, knowledgeable personnel and involvement of users [28].

The success of electronic services depends on whether users find them valuable and start using them. In e-business solutions it has been found that sites need to be both easy to use and add value to the user, these are key attributes that increase the use of services [11], [14], [15]. The added value lies in properties as “time-saving”, “range of options” and “ease of use” [23]. The web sites should also provide enjoyable experiences; these kinds of sites will probably be visited also in the future [21].

VI. Conclusion

There is an almost infinite potential in development of electronic services in the public sector. Typically, services that are provided through the Internet are connected to sharing information. Public sector services are mostly connected to information – and Internet is a very efficient way to gather and share it. Technologically, users should be able to complete most of their transactions online. Here easy-to-use, robust and trustworthy services are needed so that more users start using e-Services in public sector. As long as there are citizens that do not use electronic services organizations must to provide services electronically and as a traditional service – the result is increased costs instead of cost savings [22].

The developers of e-Services need to better understand users of public services. Clearly, the citizens should not have to surf the Internet and try to find different services that are spread all over. Better integrated, portal-type sites would make it possible to find relevant information effortlessly. This involves integration of services that are generated in separate offices, departments and units [1], [2]. For example, too often agencies provide information only from their “own” services and activities. Instead, information should be widely available so that users would not have to guess or know what other related information and services there are so that users could better have their problems solved. Services should also include information, advice and links that are not provided by the agency itself. There is a need “…to approach the Web with a philosophy of helping users solve problems, not merely delivering their same old services through new medium” [1].

It is very important to look at services from user’s perspective – whether they are connected to e-business or public services. When compared to e-business applications there are several shortcomings in e-Services [27]. The most obvious is that services should be presented in an integrated way so that they are easy to find, understand and use. Sites should also be trustworthy which increases the use of services. Services could enhance self-service which reduces the workload in the public administration, and may be seen positive from the user perspective. Citizens may have their application filled anytime, anywhere; for example, it is an added value of electronic services. If services are based on existing departments, administrative procedures and processes it may not be able to provide added value to the user. There is a need to do things differently, cross boundaries and redesign processes when designing e-Services. The work of developing and rebuilding government for the digital age is just beginning [1].

References

[1] R. D. Atkinson and A. Leigh, “Customer-oriented E-Government: Can We Ever Get There?” Journal of Political Marketing, 2 (¾), 2003, pp. 159-181.

[2] A. Ancarani, “Towards quality e-service in the public sector: The evolution of web sites in the local public service sector.” Managing Service Quality, Vol. 15, No. 1, 2005, pp. 6-23.

[3] M. Asgarkhani, “The Effectiveness of e-Service in Local Government: A Case Study”, The Electronic Journal of e-Government, Vol.3, No 4, 2005, pp. 157-166.

[4] K. Blomqvist, Partnering in the Dynamic Environment: The Role of Trust in Asymmetric Technology Partnership Formation. Acta Universitatis Lappeenrantaensis 122, 2002.

[5] S. S., Dawes and T. A. Pardo, “Building collaborative digital government systems”, In: McIver, W.J. & Elmagarmid, A.K. (Eds.) Advances in digital government. Technology, human factors, and policy. Kluwer Academic Publishers, Norwell, MA, 2002.

[6] A. Dutta and R. Roy, “Anticipating Internet diffusion.” Communications of the ACM, (2), 2003, pp. 66-71.

[7] J. R. Gil-García and T. A. Pardo, “E-government success factors: Mapping practical tools to theoretical foundations.” Government Information Quarterly (22), 2005, pp. 187-216.

[8] D. Gouscos, M. Lambrou, G. Mentzas, and P. Georgiadis, “A Methodological Approach for Defining One-Stop e-Government Service Offerings.” Proceedings of Electronic Government, Second International Conference, 2003, pp. 173-176.

[9] H. Hasan and H. R. Tibbits, “Strategic management of electronic commerce: an adaptation of the balanced scorecard,” Internet research: Electronic Networking Applications and Policy, Vol. 10, No. 5, 2000, pp. 439-450.

[10] M. Howard, “e-Government across the Globe: how Will ”e” Change Government?” Government Finance Review, August, 2001.

[11] M. Igbaria, J. Iivari, and H. Maragahh, “Why do individuals use computer technology?” Information and Management, 29(5), 1995, pp. 227-238.

[12] Z. Irani, P. E. D. Love, and A. Montazemi, “E-government: past, present and future.” European Journal of Information Systems. 16, 2007, pp. 103-105.

[13] H. Kubicek and M. Hagen, “One Stop Government in Europe: An Overview.” In Hagen, M., Kubicek, H. (Eds. 2000). One Stop Government in Europe. Results from 11 National Surveys. University of Bremen, Bremen 2000, pp. 1- 36.

[14] M. K. Lee and E. Turban, E. “A Trust model of consumer Internet shopping.” International Journal of Electronic Commerce, 6(1), 2001, pp. 75-91.

[15] K-S. Lim, J-S. Lim, and J. H. Heinrichs, “Testing an Integrated Model of E-Shopping Web Site Usage.” Journal of Internet Commerce, Vol 7(3), 2008, pp. 291-312.

[16] R. McIvor, M. Mchugh, and C. Cadden, “Internet technologies supporting transparency in the public sector.” The International Journal of Public Sector Management, Vol. 15, No. 3, 2002, pp. 170-187.

[17] S. Mulpuru, "Topic overview: US online retail. Forrester Research." 2007. Available from http://www.forrester.com/go?docid=41752 Accessed 26th August 2010.

[18] A. M. Parhizgari and G. R. Gilbert, "Measures of organizational effectiveness: private and public sector performance." Omega, 32(3), 2004.

[19] R. T. Rust, P. K. Kannan, and N. Peng, "The Customer Economics of Internet Privacy." Journal of the Academy of Marketing Science, Vol 30, No 4, 2002, pp. 455-464.

[20] R. T. Rust and P. K. Kannan, "E-service: A New Paradigm for Business in the Electronic Environment," Communications of the ACM, Vol 43, No. 6, 2003, pp. 37-42.

[21] R. A. Shang, Y. C. Chen, and L. Shen, "Extrinsic versus intrinsic motivations for consumers to shop on-line." Information and Management, 42(3), 2005, pp. 401-413.

[22] B. Sundgren, "What is a Public Information System?" International Journal of Public Information Systems, Vol 2005:1, 2005, pp. 81-99.

[23] K. de Ruyter, M. Wetzels, and M. Kleijnen, "Customer adoption of e-service: an experimental study." International Journal of Service Industry Management, Vol 12, No 2, 2001, pp. 184-207.

[24] Z. Van Der Wal, L. Huberts, H. Van Den Heuvel, and E. Kolthoff, "Central Values of Government and Business: Differences, Similarities and Conflicts." Public Administration Quarterly, 30(3), 2006.

[25] U. Löfstedt, "E-government – Assessment of currentr research and some proposals for Future Directions." International Journal of Public Information Systems, vol 2005:1, 2005, pp. 39-52.

[26] KuntaIT, "National Survey of Electronic Services in Finland." Ministry of the Interior, Finland, 2009 (available from http://www.kuntait.fi/Kuvat/KuntaIT_kysely_joulukuu__2009.pdf). Accessed 26th August 2010.

[27] M. A. Ward, "Information Systems Technologies: A Public-Private Sector Comparison." Journal of Computer Information Systems. Spring, 2006, pp. 50-56.

[28] J. Ho and T. A. Pardo, "Toward the Success of eGovernment Initiatives: Mapping Known Success Factors to the Design of Practical Tools." Proceedings of the 37th Hawaii International Conference on System Science, 2004. IEEE.

Information and Knowledge Sharing: Involving Customers in Developing Services

Seppo J. Sirkemaa
28120 Pori, Finland
seppo.johan@gmail.com

Abstract— **Understanding customer needs is a critical success factor in development of products and services. In most enterprises there are several people interfacing with customers, throughout the organization in different units and functions. The result is that information on customer needs and requirements exists, but it remains scattered and unorganized. Another issue is that those who are interacting with customers are not necessarily involved with development of products and services. Here the challenge is to integrate information and knowledge from customer needs to the R&D process. Theoretically, there are mechanisms for sharing information and knowledge. Using different knowledge sharing mechanisms becomes especially important in large organizations which operate in multiple locations. These call for infrastructures, rules and procedures so that sharing of information would be possible. In this research, we study mechanisms that empower the sharing of knowledge and information on customer requirements, so that it could be effectively used in the R&D process**

Keywords- knowledge sharing, R&D, customer needs, customer requirements, absorptive capacity, development

I. INTRODUCTION

Information and knowledge sharing is important in all types of organizations. This means that there is a need to access and capture information, and shared it between units, teams and individuals throughout the organization. Zahra and George [17] use in this context the term absorptive capacity. It is defined as organizational processes and routines that are used in capturing, integrating and using information in developing dynamic capabilities in the organization. Absorptive capacity is here connected to the challenge of understanding customer, which involves sharing information and knowledge on customer needs.

Information and knowledge sharing is facing challenges which depend on the type of information that is intended to be shared. In operative activities knowing what, why and when is needed in order to have things done. This type of information is often numeric and manageable, in a way that it can be captured, stored or generated from processes. In contrast, information that is needed in business processes and strategic management of the enterprise calls for information which is not as straightforward to express in written form, and thus managing and sharing of this type of information is more challenging.

In this research we look at challenges in understanding customer information, in capturing and sharing knowledge on customer needs. The research is based on frameworks of absorptive capacity and classification of different knowledge-sharing mechanisms which will be used in studying different knowledge sharing practices found in case organizations. The empirical part of the research relies on interviews with selected persons in R&D and marketing departments. We also gave key customers a questionnaire so that the information sharing interfaces and mechanisms could be mapped. We approached altogether 9 large case organizations which all share an interest in developing services based on customer needs. Most of the case organizations operate in the industry, have manufacturing sites and units in different countries and customers worldwide.

The structure of the paper is following: first we look at different types of information and knowledge. Depending on the type of information the mechanisms and methods of sharing it vary; this is an issue that is being discussed in next chapter. Involving technology and information systems is a delicate issue, in some cases personal approach is more appropriate. In the conclusion we argue that there are several methods in listening, engaging and empowering people who are involved in sharing information on customer needs.

II. MANAGING INFORMATION AND KNOWLEDGE

Information and knowledge is a multifaceted concept. Polanyi [14] differentiates between tacit and explicit types of knowledge. The taxonomy is based on the ease of articulating and communicating knowledge to others. Another viewpoint is whether information and knowledge is individual or collective, or the extent to which knowledge is being held by one individual or embedded in the interaction of a group of people [3], [12].

In this research information and knowledge are studied in business context. Shortly, successful companies understand customer needs. This involves focusing on customers and working together with the clients in order to integrate customer-related information in to the development process. Customers have experience-based knowledge of products and services, therefore it would be important to better connect customers to development process [16], [4]. Managing customer related information includes capturing and sharing external customer information available from the customer himself or herself, for example ideas from a meeting with the customer. It also involves internal information; capturing and managing information on customer purchases, profitability etc. which are result of operations and obtainable from within the enterprise that is dealing with the customer.

Information management is here connected to the concept of absorptive capacity. It is a combination of processes and routines that are used in the organization in capturing, integrating and using information in developing dynamic capabilities. These capabilities give the organization potential to develop competitive advantage in the market [10]. Methods, processes and practices vary from organization to another, they may not be easily copied from one setting to another. The role of contextual elements – fit of people, information sharing methods and platforms to a certain situation – is what makes managing customer related information effective.

Managing information and knowledge is here defined as management of information, and sharing this information with others so that it adds value. This requires understanding what information is about, contextual factors are important so that information can be used wisely. Knowing what to do with information, how it has been generated etc. are examples of context-related issues. It is understanding the contextual factors that separates knowledge from information [1].

In this paper we look at information and knowledge that is needed in developing business further. Understanding the needs of the customer is cornerstone of business operations, therefore we study customer-related information. As noted by Hou [10] besides customers are other sources of external information as well which should be managed – like information from competitors and what they are doing, information on technological advancements. In addition, political and economical changes need to be taken into account.

III. INFORMATION AND KNOWLEDGE SHARING

The term knowledge-sharing mechanism refers to formal and informal mechanisms which are used in organizations for sharing information and knowledge. The focus is on sharing information and knowledge that is embedded in individuals or groups so that it can be used in work-related processes and activities [2], [3], [12].

Information and knowledge that should be shared may not always be easily shareable. Information sharing involves mapping the persons who are sharing information, defining the information itself and passing it on in an understandable format. When the information is complex this is not easy, sharing it with others may require prior knowledge of the subject area, and call for further clarification. If there is person-to-person interaction between humans the interaction itself is one component in the process. Here for example how information is presented and how reliable the source of information is considered affect the knowledge sharing process.

Aggregation of knowledge is a key concept in understanding information and knowledge sharing [3], [13]. It refers to the degree of aggregation which varies from personal and individual to collective knowledge. Personal knowledge is by definition individual, collected and stored for private use whereas collective knowledge is – ideally – generated, gathered and stored in a way that it is shareable and available for colleagues. Collective information and knowledge may be seen as integrated, embedded and institutionalized part of structures and routines of the organization. It must be noted that in organizations there is also “personal” information which is intended for personal use. Often this type of information is stored in the memory aids that people create for themselves. For example, notes that people write with their portable devices in a meeting. However, this knowledge and information might also be shareable and accessible to others, and it might be valuable to colleagues when shared. This highlights the fact that people are in the core of all knowledge sharing mechanisms – sharing information and knowledge is not technology alone even though there are all kinds of technical solutions available.

The degree of articulation is another key concept in this context. Degree of articulation is classified into explicit and tacit knowledge [14]. Typically, some information is relatively straightforward, easy to convert into oral or written format and thus can be made known – explicit - to others. Tacit knowledge cannot be shared as easily: tacit knowledge is often based on expertise and experience, and may not be explained or expressed in words [12].

Typically, in organizations there is plenty of data which is generated through organizational processes and directly stored into digital format. This type of information can easily be accessed, combined and transferred from one location to another. As an example, data on production and sales volumes is typically stored to information systems which are used in these functions. The related question could be

“How many units of product A have been manufactured in line X on last Tuesday?”

This kind of information is result of the manufacturing process, and data can be retrieved from the manufacturing information system so that the question can be answered. The answer is a number, straightforward to understand and can be easily shared with others. However, all questions may not be answered by retrieving the correct answer from the database. Many answers contain knowledge which is more difficult to articulate, often because it deals with issues which are not clear. Consider following question

“Why have the manufacturing volumes of line X been dropping?”

These manufacturing-related questions indicate the challenges of articulation. The first question can be answered by simply checking the production numbers. The prerequisite is that the company has a system for storing, browsing and retrieving manufacturing information. Usually this is the case so there is no problem here. The second question– why have manufacturing volumes been dropping - can be connected to several “issues” which have their roots either inside the company or in its environment. The volumes may have gone down because of problems with raw-material delivery, or there is not enough demand on products, for example. The explanation may require combining different facts and issues. Answering the question calls for understanding the overall setting, taking into account several factors that affect manufacturing is needed in order to answer the question thoroughly. This kind of expertise is referred to as tacit knowledge [12].

A. Mechanisms in information and knowledge sharing

Information and knowledge sharing is based on two main mechanisms (Table 1): knowledge may be shared from person-to-person which is referred to as personalization, or using technology and information systems to store, manage and share information that is codified to a format that allows this. When knowledge and information can be expressed in words and numbers it can be codified and shared, transmitted and stored in electronic format which allows browsing, retrieving and combining information. On the other hand, mechanisms for sharing experience-based, unclear, intuition-based or non-verbal information with others - tacit knowledge - cannot take advantage of technology and computers, not at least as directly as in the case where information is originally alphabetical, created and managed with information systems.

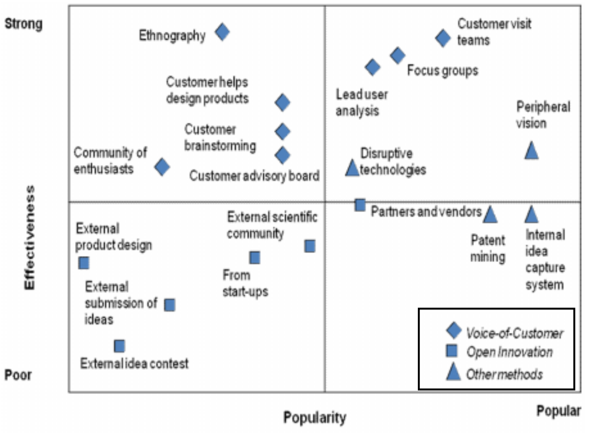
TABLE 1. MECHANISMS FOR SHARING KNOWLEDGE

	Individual	Collective
Tacit - Personalization	Social Networking	Facilititating person-to-person knowledge sharing
Explicit - Codification	Sharing personal memos, working papers etc.	Knowledge management applications

The mechanisms for sharing knowledge (Table 1) combine the two dimensions of information and knowledge. As a result, sharing knowledge and information is classified into individual-explicit, individual tacit, collective-explicit and collective-tacit classes. The degree of articulation is shown on the vertical axis explicit – tacit, and the level of aggregation - individualization and collectivism - is on the horizontal level. Depending on the degree of individualization and collectivism and the type of knowledge are the knowledge sharing mechanisms more person-to-person oriented or rely on sharing codified information, often using information technology. Classifying information and knowledge into one of the four segments helps in understanding mechanisms and methods in collecting, storing and sharing information, and the roles of information technology vs. person-to-person interaction. For example, when information is explicit, and/or collective by nature storing, retrieving and transferring it is relatively easy by using information technology (down-right corner in Table 1). [3], [7].

Information technology has an increasingly important role in knowledge sharing, but it is not the only thing that is needed here. Information that is individual and tacit is considerably more challenging to share (the opposite corner in Table 1). Boh [2] argues that when the complexity of information increases more personal interaction is needed in understanding it. Often sharing information calls for discussion and clarification so that there will become mutual understanding of phenomena. This highlights the importance of personal interaction, especially when the information is tacit.

In real life there are several practices for sharing information and knowledge in organizations. Factors like organizational size, geographical distances and industry explain the variety of different knowledge sharing methods. Typically, when the organizational size increases the challenges in managing dispersed locations, units and teams require more integration than in a smaller company. The challenging issue of interest is finding the best methods in sharing information and knowledge throughout the organization. Practices in one organization may not fit other, and even between units in one enterprise there are differences in the way information sharing among people is done. Typically, there is a built-in process of adjusting and developing process to address the needs of the people involved [2].



Source: Cooper & Edgett 2008

Figure 1. Methods in sharing information with customers

In this context understanding customer needs and sharing customer-originated requirements is of special interest. The taxonomy of methods in sharing information with customers (Figure 1) is a summary of results from a research which included over 160 companies. In this research Cooper and Edgett [6] studied the methods of capturing and sharing customer information on customer needs, in connection with idea generation and R&D development. Here the objective is to determine how extensively each method is used (popularity), and to gauge management's perception of the value of the method (effectiveness on the vertical axis) in generating high-value new product ideas.

The findings of Cooper and Edgett [6] indicate that most effective results can be achieved by working together with customer (methods above the horizontal middle line). Examples of these methods are customer visit teams, focus groups and lead user analysis. Also describing the relation from an ethnographic viewpoint and involving the customer in helping to design products, brainstorming, customer advisory board and building a community of enthusiasts were considered effective.

The most popular methods include internal idea capturing system, which usually involves staff soliciting new product ideas (often using internal Web pages), and then screening and managing these ideas with a structured process. Peripheral vision and patent mining were also very popular, and like internal idea capturing system these are methods that are based on organizations internal ideas, and do not directly involve customers in idea generation. Peripheral vision refers to assessing the external world to identify trends and threats and, through this process define potential new products. Patent mining involves mapping or mining others' patents and seeking technical and competitive ideas. However, this method does not directly generate new product ideas as such. The common element of these methods is that they do not involve customer in idea generation.

Discussions and changing ideas with the customer is a key issue. Relying on ideas submitted to competitions is not very effective (lower left-corner). It is still possible to take advantage of technology also in this context. Making notes with computers and PDA's, sharing memos with email, using CRM-systems to record ideas are examples on how technology can support capturing and sharing information and knowledge – once it has been codified to written format.

How is information being shared with the above methods? The mechanisms for sharing information and knowledge are mostly person-to-person, there are few exceptions like external submission of ideas that rely directly on technology in capturing ideas. The information in product and service development is mostly complex, unclear and tacit. When new innovations are being developed sharing ideas, discussing and brainstorming together rely heavily on personal interaction with others. In most cases ideas, comments and advice from the customer come through discussions, meetings, workshops and alike.

B. *Personal interaction or information technology*

Let us look at knowledge codification. It refers to a knowledge-sharing mechanism which aims to capture knowledge into systems that are accessible to others in the company. Usually knowledge-sharing mechanisms that rely on codification take advantage of information technology, computer networks and knowledge management applications. The mechanism should capture knowledge that is individual or collective in nature and make it wider property of the organization [8]. Hence, there should be an underlying technical infrastructure for sharing codified knowledge which has to be implemented, in operation and continuously developed in order to create a robust platform for sharing codified knowledge.

Information that is tacit cannot take advantage of information technology as directly: experience-based knowledge may need to be interpreted in order to be understood. Also in this case organizations have to develop platforms for sharing knowledge and information, there is a need for facilitating person-to-person knowledge-sharing – personalization - which is the primary mechanism for sharing tacit information. Here social networking is important [9].

It is argued that knowledge sharing mechanisms that are based on personalization are rather 'ad hoc' and informal because they are result of interaction between humans. On the other hand, the advantage of personal interaction is flexibility and the possibility to transmit and share tacit knowledge. In contrast, sharing codified information is assumed to be more formal and incorporate electronic databases in knowledge sharing [5].

Information technology allows connecting to information anywhere and at any time. However, this is not the case with person-to-person knowledge sharing. It has relatively poor geographical reach and also the number of people who can share information in this way is limited [18]. Whereas codified information can be coded, stored, browsed and retrieved with computer technology, requires personalization a more 'personal' approach. The knowledge seeker has to get into touch with the potential knowledge provider, and the knowledge provider should be willing to share information and knowledge with the knowledge seeker. Knowledge sharing via personalization means that the person who seeks knowledge is aware of what information others have, and knows where to find memos, working papers etc. Asking from colleagues may also lead to other concerns; seeking information from others in the company may be considered as a weakness or ignorance on a given issue [11].

The most significant benefit of personalization is flexibility in sharing information as personal interaction allows clarifications, argumentation and discussion, thus making it possible to share ideas and get feedback, build "consensus" in a way that it becomes clear that both the sender and receiver of information understand the issues and phenomena that are being shared. Furthermore, discussions and sharing different viewpoints can also generate new knowledge [15]. Promoting personal interaction and personalization should therefore be emphasized throughout the organization.

IV. Discussion

It is important to notice that some knowledge and information is created as result of cooperative action while other is result of personal reflection, intuition and understanding. In both cases the importance of human element in sharing knowledge is significant. Dealing with individual and/or tacit information is more challenging than sharing straightforward data [2], [7], [12]. Furthermore, the greater the complexity of the information the more is personal interaction needed in understanding it [2]. Often both technological solutions and person-to-person interaction mechanisms are used as these tend to complement each others in knowledge sharing [2]. This notion gives us an additional dimension to the framework for sharing knowledge.

The use of different knowledge sharing mechanisms in product and service development was studied empirically in a group of case organizations. The goal was to identify and evaluate methods in capturing customer requirements in product and service development. Here we study the mechanisms that have been developed and are being used in gathering, storing and sharing information on customer needs, and then integrating this information into product and service development process. As a result, the question of interest was how customers are connected into the development of products and services. We also studied how information technology is being used here, and what is the role of person-to-person information sharing.

The case companies were relative product-oriented, but they share a need for developing integrated services to the customers. This involves integrating existing products and services together, and moving from time-based pricing towards new pricing methods. Here issues like smooth operations, maintaining quality and avoiding interruptions are key issues, not the price of service-staff making a visit.

In developing products and services, a key issue is understanding customer needs. This research shows that there are several possible methods in listening to users, engaging and empowering the developers involved. Sharing information on customer needs, and making this more widely known among the co-workers is usually challenging. In the case organizations this issue was acknowledged and there were practices for sharing information on customer needs. These range from methods like working together with customers in development to internal infrastructures (such as workgroup and team-meetings, unit and company-level meetings gathering periodically or when needed). Clearly, the range of methods is an organization-dependent issue. These methods may also be extended beyond sharing customer information to a variety of other information sharing needs within the organization.

It seems that the more customer is involved in the development the better the solutions meet customer needs. It is still noteworthy that high customer involvement is only part of the formula; it is equally important to develop platforms and infrastructures for teams, units and functions that so that sharing customer-related information throughout the organization would become possible. Ultimately, knowledge sharing depends on human motivation and dedication - these should be empowered in this context.

REFERENCES

[1] Blair, D.C. “Knowledge management: Hype, hope, or help?” Journal of the American Society for Information Science and Technology, 53(12), 2002, pp. 1019-1028.

[2] Boh, W. F., “Reuse of knowledge assets from repositories: A mixed methods study.”. Information & Management, 45, 2008, pp. 365-375.

[3] Cabrera, A. and Cabrera, E. F. “Knowledge-sharing dilemmas.” Organization Studies, 23(5), 2002, pp. 687–710.

[4] Chesbrough, H. “Open Innovation. The New imperative for Creating and Profiting from Technology.” Boston, Massachusetts. Harvard Business School Press, 2003.

[5] Choi, B. and Lee, H., “An empirical investigation of KM styles and their effect on corporate performance.” Information & Management, 40(5), 2003, pp. 403–417.

[6] Cooper, R. and Edgett, S. “Ideation for product innovation: What are the best methods?” PDMA Visions Magazine, March 2008.

[7] Crossan, M. M., Lane, H. W., and White, R. E., “An organizational learning framework: From intuition to institution.” Academy of Management Review, 24(3), 1999, pp. 522–537.

[8] Earl, M., “Knowledge management strategies: toward a taxonomy.” Journal of Management Information Systems, 18(1), 2001, pp. 215–233.

[9] Hansen, M. T., “The search-transfer problem: the role of weak ties in sharing knowledge across organization subunits.2 Administrative Science Quarterly, 44(1), 1999, pp. 82–111.

[10] Hou, Jia-Jeng. “Toward a Research Model of Market Orientation and Dynamic Capabilities.” Social Behavior and Personality, Vol. 36, No: 9, 2008, pp.1251-1268.

[11] Menon, T. and Pfeffer, J., “Valuing internal vs. external knowledge: explaining the preference for outsiders.” Management Science, 49(4), 2003, pp. 497–513.

[12] Nonaka, I., “A dynamic theory of organizational knowledge creation.” Organization Science, 5(1), 1994, pp. 14–37.

[13] Olivera, F., “Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access.” Journal of Management Studies, 37(6), 2000, pp. 811–832.

[14] Polanyi, M., “Personal knowledge: Toward a post-critical philosophy.” New York: Harper Torchbooks, 1962.

[15] Prencipe, A. and Tell, F., “Inter-project learning: processes and outcomes of knowledge codification in project-based firms.” Research Policy, 30(9), 2001, pp. 1373–1394.

[16] von Hippel, E. “Democratizing Innovation.” MIT Press. Cambridge, MA. 2005 (available http://web.mit.edu) Accessed 26th August 2010.

[17] Zahra, S. and George, G. “Absorptive Capacity: A Review, Reconceptualization, and Extension.” Academy of Management Review, 27:2, 2002, pp.185-203.

[18] Evans, P. and Wurster, T. “Strategy and the new economics of information.” Harvard Business Review, 75(5), 1997, pp. 71–82.

Data Mining Governance for Service Oriented Architecture

Ali Beklen
Software Group
IBM Turkey
Istanbul, TURKEY
alibek@tr.ibm.com

Turgay Tugay Bilgin
Dept. of Computer Engineering
Maltepe University
Istanbul, TURKEY
ttbilgin@maltepe.edu.tr

Abstract— **The aim of this study is to propose a platform called Data Mining Registry, Repository and Statistics (DMRRS). The concept of this platform is to govern the data mining algorithm which needs to be integrated to service oriented architecture and to be used in the cloud analytics environment. The focus is on the notion of a reference architecture for DMRRS, XML schema-based algorithm definition data models and data mining algorithm life cycles.**

Keywords -service oriented architecture; soa governance; data mining.

I. INTRODUCTION

In recent years, data mining has attracted a great deal of attention in the information industry, as well as in society as a whole. This is due to the wide availability of huge amounts of data and the imminent need for converting such data into useful information and knowledge [1]. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration [1]. Data mining can be viewed as a result of the natural evolution of information technology [1].

Present day data can no longer be labelled as ‘simple’ [2]. As data in various domains becomes more heterogeneous, complex and peculiar, more intelligent techniques are required to mine it and to extract useful knowledge from it [2].

Data mining is a compilation of techniques, methods and algorithms utilized in order to extract knowledge hidden amongst huge amounts of data. It is, therefore, much more than a list of statistical formulas applied to a collection of data [2].

According to current trends, cloud computing and service oriented architecture are emerging as complex applications of the implementation and presentation type. Regarding data mining, there are many topics that need to be researched to adopt data mining algorithms with cloud computing and service oriented architecture.

In order to implement the data mining algorithm as a service, many interface options are available, for example web service. In order to administer the services, service registry and repository are necessary to define and manage the interfaces. Although the services could be organized in a service registry and repository, this does not allow developers or architects to manage and define the algorithm itself.

Management of the data mining algorithm necessitates a governance lifecycle. This lifecycle must allow information architects to test the maturity level of the algorithm in terms of performance and resource consumption. On the other hand, long running algorithms need to be monitored to predict the duration of further mining requests.

The aim of this study is to build a reference approach to adopt data mining algorithms to service oriented architecture (SOA) in a governed way to address the above issues. In order to implement the reference approach, data mining governance life cycle, the architecture of Data Mining Registry, Repository and Statistics (DMRRS) implementation and a sample algorithm definition schema have been developed.

II. RELATED WORK

There have been many studies published with the aim of adopting data mining to service oriented architecture. Chen et al. proposed service rating for data mining to improve information sharing [3]. Xu et al. proposed a service based architecture for data mining applications, including configuration service, service engine, monitor service, analysis service, visualization service, computing service and data and algorithm provision service [4]. Tsai et al. proposed a Dynamic Data Mining Process (DDMP) system based on Service-Oriented Architecture (SOA), which enables each activity in the data mining process to be promotable as a web service operated on the Internet, providing data preprocessing, data mining algorithm, and visualization analysis functions [5].

The Data Mining Registry, Repository and Statistics concept proposed in this paper however, is about data mining governance which is very important in terms of algorithm definition, life cycle management, runtime analysis reporting, and managing long and short running processes related to governed algorithms.

III. GOVERNANCE

Governance is the process of making correct and appropriate decisions on behalf of the stakeholders of those decisions or choices. In its corporate application, governance is the process of ensuring that the best interests of a company's or organization's stakeholders are met through all corporate decisions, from strategy through to execution. In an IT framework, governance focuses on appropriate oversight and stakeholder representation for IT spending and overall IT management [6].

SOA governance is the creation, communication, enforcement, and adaptation of policies used to direct and control the creation and implementation of the life cycle of services. It is a run-time and design-time administrative capability that no organization should be without [7].

In this research, we are proposing data mining governance as a new governance type to define the life cycle of data mining algorithms, which requires management and monitoring in a service oriented accessible repository.

IV. DATA MINING GOVERNANCE LIFECYCLE

In order to manage the life cycle of data mining algorithms and to integrate with service oriented architecture, defined policies and life cycle stages are necessary. In this study, a candidate reference life cycle is proposed. This life cycle consists of the following stages:

1. Define the purpose and development of algorithm.
2. Define the algorithm metadata and communication interface to make it discoverable
3. Test the maturity level with a pre-defined training data set
4. Deploy the algorithm
5. Monitor the algorithm runtime environment
6. Collect the feedback from the runtime environment

Figure 1 shows the life cycle.

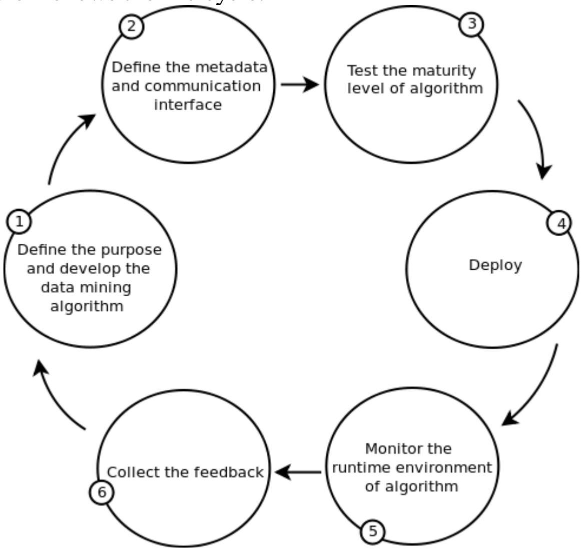

2
Define the metadata and communication interface
3
Test the maturity level of algorithm
1
Define the purpose and develop the data mining algorithm
4
Deploy
Collect the feedback
6
Monitor the runtime environment of algorithm
5

Figure 1. Data Mining Governance Life Cycle

The SOA governance addresses administration of the service life cycle. However, if the project needs to expose the data mining algorithms as a service, several problems may occur, such as:

- Atomic web services are not suitable for long running transactions, and data mining algorithms need to be managed and monitored as long running.
- SOA governance defines service and its life cycle but data mining algorithm does not fall within this scope.

The most important stage of the proposed life cycle is testing the maturity level of data mining algorithm. This stage guarantees the algorithm quality, in terms of algorithm duration, hardware resource consumption and mining result. For example, if the data mining algorithm owner would like to publish the algorithm, this step forces it to compare the algorithm with the best performing similar ones by using the same training data set. If it performs at least the mature ones this lifecycle stage allows him/her to move forward in the lifecycle.

On the other hand, provision of the data mining capabilities of the project as a service necessitates that it be discoverable by another system in the SOA. In order to mitigate with data mining governance requirements and to operate the life cycle, we propose the data mining registry, repository and statistics concept.

V. DATA MINING REGISTRY, REPOSITORY AND STATISTICS

The SOA service registry and repository is the service repository used for storing, accessing, and managing metadata used in the reuse, selection, invocation, management, and governance of SOA services in SOA architecture. It helps to define a central point for accessing predefined and custom service description artefacts acquired from a number of sources, including service and service application deployments, development tools, and other service metadata repositories. Interfaces are provided for finding, retrieving, and advertising services using classifications and properties during application and service design and development, service change and release processes, service invocation and execution, and operational management of services.

DMRRS is a concept which defines data mining algorithms and their metadata, manages their life cycle and monitors their performance. With the help of this concept, it becomes possible to govern native data mining algorithms, expose their interfaces in a transparent way to the SOA, help other services to discover algorithms, and link the Predictive Model Markup Language (PMML) to algorithms.

One of the critical requirement about proposing the data mining algorithm as a service is building and managing an association between PMML and algorithms. The DMRRS concept provides setting up an association between PMML and algorithms and allows the mining requestor to select and associate multiple algorithms to one mining model.

Another important factor in data mining is the runtime duration of the long running algorithms. This research proposes the statistics manager component. The proposed component in the tool's architecture is responsible for measuring the duration of running algorithm instances, and for providing this information to assist in prediction of the transaction duration for the next request to the other consumer systems. This also helps the architect to understand the behaviour of the algorithm, and to serve it as synchronous or asynchronous.

The component architecture diagram of the DMRRS represented in Figure 2 consists of six major components:

- Authentication: This component is responsible for authentication of users who have a definition in the user repository.
- Authorization: This component is responsible for authorization of users and generates user interfaces based on use rights. There are three types of users :
  - Mining Algorithm Developer: This role can create a candidate data mining algorithm definition and initiate a lifecycle. This role is also responsible for defining the metadata and communication interface and testing the maturity level of the algorithm.
  - SOA Architect: This role is responsible for auditing the compatibility of definitions with the SOA governance and approves the verified algorithm for deployment.
  - Data Architect: This role is responsible for linking the different algorithms and connecting them to Predictive Model Markup Language (PMML) documents. This approach proposes the building of a composite algorithm which consists of a unique flow of different types of algorithms, and links the algorithm runtime interface to the PMML definition.
- Document Manager: This component is responsible for managing the PMML documents and Data Mining Algorithm Definition Language (DMADL) documents. It proposes management of basic operations concerning these types of documents, and generation of DMADL type documents.
- Document life cycle modelling: This component is responsible for managing the data mining governance life cycle stages. It proposes assignation of responsibility roles to the stages.
- Algorithms runtime statistics: This component is responsible for providing the algorithms' runtime related statistical data as a web service. The last algorithm request uses this statistical output if it is available to predict the duration of last mining request. It does this by comparing the requested training data set size and algorithm type with the similar historical requests.
- Algorithm runtime manager: This component monitors the instances of running algorithms and feeds the runtime statistics database with data about transaction duration, central processing unit (CPU), random access memory (RAM) and hard disk drive input/output (HDD I/O) utilization.

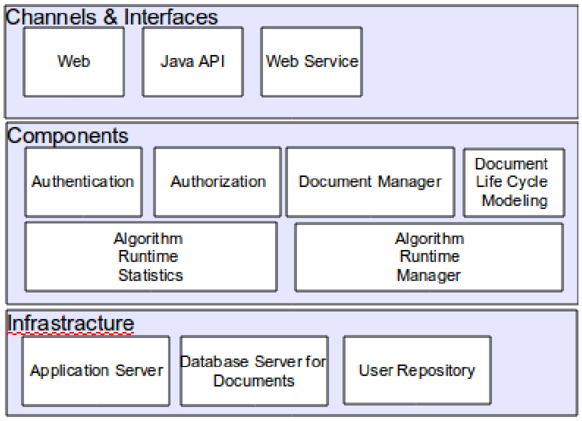


Figure 2. DMRRS Component Architecture Diagram

In many successful SOA implementations, all of the services base their communication on enterprise semantics. These semantics usually include a common vocabulary, a semantic information model, and common schemas. What is helpful is that such an approach does not require data transformation throughout the enterprise. Rather it is the responsibility of service consumers and providers to implement the abstractions from their internal data models to enterprise semantics [8].

Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information [10]. DMRRS tool allows data mining algorithm owner to define its metadata to make it discoverable. Otherwise, algorithm requestor does not able to search and discover the required algorithm.

In order to govern the data mining algorithm, define its metadata and to adopt them to SOA governance, this research proposes the data mining algorithm definition data model (DMALD) which helps to define common data mining metadata fields and implementation details. Because of data mining algorithm characteristics, every data mining algorithm requires different types of metadata. In this research we developed a sample algorithm metadata model called association rule learner algorithm (ARMD). In order to define the DMALD and ARMD, Extensible Markup Language (XML) schema is used.

The XML schemas enable you to declare the type of textual data allowed within attributes and elements, using simple type declarations. For example, by utilizing these types you could specify that an element may contain only date values, only positive numbers, or numbers within a certain range. Many commonly used simple types are built into XML Schemas. This enables you to easily create documents that are intended to represent databases, programming languages, and objects within programming languages [9].

In Figure 3, the relationship between the ARMD and DMALD XML schemas is clearly shown at a high level. The boxes titled Governance and Implementation are inherited from DMALD as a complex type and help to avoid duplication of common fields among the mining algorithm definitions. These schemas are also used in the DMRRS tool to govern and integrate the algorithm to the SOA environment in a standard way.

A part of the schema designed for association rule mining definition can be seen as follows:

```
<xs:include schemaLocation="DMALD.xsd"></xs:include>
  <xs:annotation>
      <xs:documentation>
         Assocation rule mining definition schema.
      </xs:documentation>
  </xs:annotation>
  <xs:complexType name="ARMD">
      <xs:sequence>
        <xs:element name="GovernanceInfo" type="Governance"/>
        <xs:element name="Implementation" type="Implemantation">
         </xs:element>
        <xs:element  name="InputParams"  type="InputParameters"/>
        <xs:element name="OutputParams"
                    type="OutputParameters">
      </xs:sequence>
  </xs:complexType>
  <xs:complexType name="InputParameters">
      <xs:sequence>
        <xs:element name="MaxItemsetCount" type="xs:long"/>
        <xs:element name="MaxItemsetSize" type="xs:long"/>
        <xs:element name="MaxSupport"
                    type="xs:nonNegativeInteger"/>
        <xs:element     name="MinItemSetSize"     type="xs:long"/>
        <xs:element name="MinProbability"
                    type="xs:nonNegativeInteger"/>
        <xs:element name="MinSupport"
                    type="xs:nonNegativeInteger"/>
      </xs:sequence>
  </xs:complexType>
```

VI. CONCLUSION

In this study, the importance of adopting data mining governance to SOA governance has been examined in detail. The key requirements of data mining governance in enterprise level applications have been mentioned.

In order to implement the data mining governance concept in a SOA project, DMRRS tool, governance life cycle and data mining algorithm definition data model have been designed and proposed as a reference.

Implementation design has been kicked off and our subsequent studies will focus on making it live and increasing the number of implemented data mining algorithm definition data model types.

On the other hand, cooperating with a cloud analytics project will be a challenging continuation in terms of integrating the proposed approach and tools to a cloud environment.

REFERENCES

[1] J. Han and M. Kamber, "Introduction". Data Mining: Concepts and Techniques, Second Edition. Morgan Kaufmann Publishers, 2006, pp. 1-4.

[2] D. Taniar, "Preface". Strategic Advancements in Utilizing Data Mining and Warehousing Technologies: New Concepts and Developments. IGI Global, 2010.

[3] Y. Chen, Brad Cohen and B. A. Hamilton, Data Mining and Service Rating in Service-Oriented Architectures to Improve Information Sharing. Aerospace Conference, 2005, pp. 1-10.

[4] L. Xu, Y. Wang, G. Geng, X. Zhao and Nan Du, SDMA: A Service-based Architecture for Data Mining Applications. IEEE International Conference on Services Computing, 2008, pp. 1-2.

[5] C. Tsai and M. Tsai, A Dynamic Web Service based Data Mining Process System, Computer and Information Technology, 2005, pp. 1-7.

[6] E. A. Marks, "Chapter 1 - The SOA Governance Imperative". Service-Oriented Architecture Governance for the Services Driven Enterprise. John Wiley & Sons. 2008.

[7] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, "Chapter 12 - SOA Governance". Applied SOA: Service-Oriented Architecture and Design Strategies. John Wiley & Sons. 2008.

[8] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, "Chapter 5 - Service Context and Common Semantics". Applied SOA: Service-Oriented Architecture and Design Strategies. John Wiley & Sons. 2008.

[9] D. Hunter, "Chapter 5 - XML Schemas". Beginning XML, 4th Edition. Wrox Press. 2007.

[10] National Information Standards Organization, “What is metadata”. Understanding metadata, NISO Press. 2010.

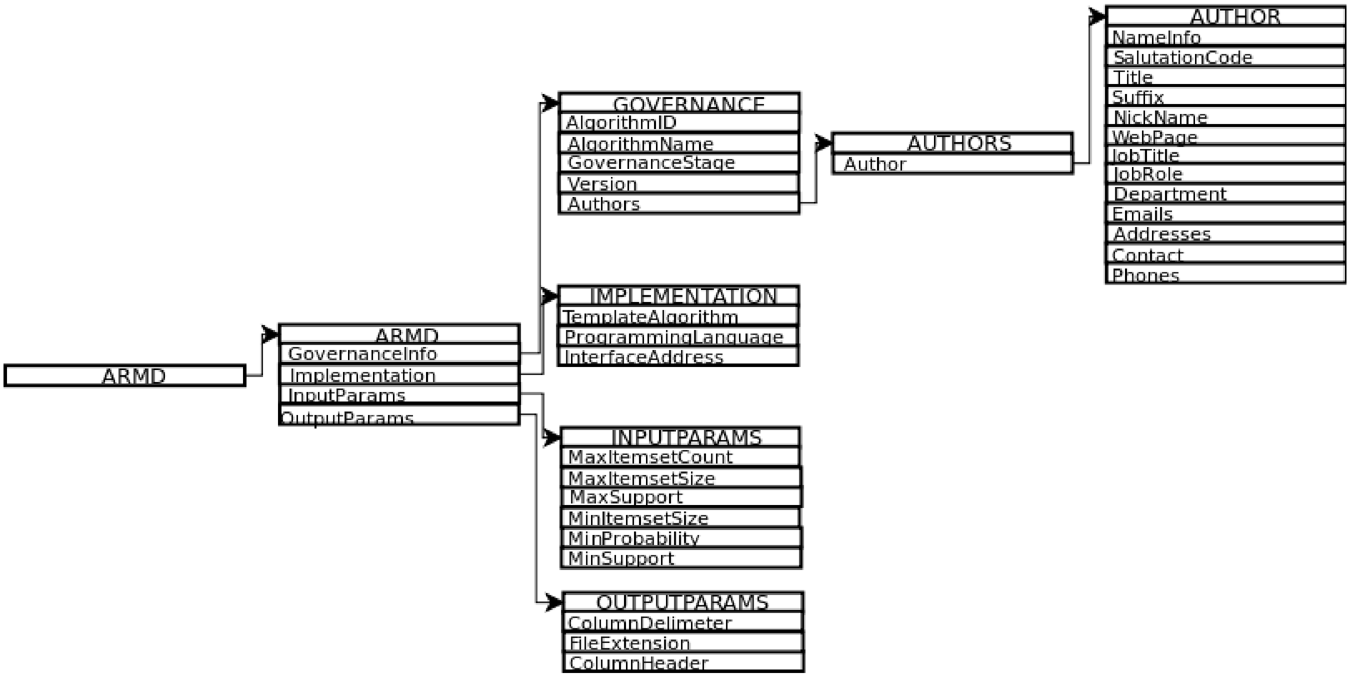


Figure 3. ARMD Schema Architecture

Generic Function Schema for Operations on Multiple Network QoS Parameters

Mark Yampolskiy[1,2,3], Wolfgang Hommel[2,3], David Schmitz[1,2,3], Matthias K. Hamm[3]
[1]*German Research Network (DFN),* [2]*Leibniz Supercomputing Centre (LRZ),* [3]*Munich Network Management (MNM) Team*
myy@dfn.de, hommel@lrz.de, schmitz@lrz.de, hamm@mnm-team.org

***Abstract*—Graphs are often used to model interconnected topological objects with different connection properties. Path finding in a weighted graph belongs to the classical problems of graph theory. Whereas the addition of the edges' weights as an aggregation and the interpretation of a smaller resulting sum as the preferable path works very well in applications like path computations, e.g., for road maps, it is not always applicable to those connections in computer networks that need to fulfill multiple independent Quality of Service (QoS) criteria in parallel. Until now, usually a special – and often manual – solution has been implemented for each new service with different QoS parameters. As the development of novel customer-faced network services often relies on different connection properties and their combinations, a generic treatment of QoS parameters becomes a critical factor for rapid development and network service rollout. In this paper, we present our proposal for treating multiple independent QoS parameters in a similarly fashioned way. Our work is aimed to foster routing algorithms that are considering multiple connection properties and corresponding constraints at the same time.**

***Keywords*-graph theory, multi-weighted graphs, QoS, QoS aggregation, QoS comparison.**

I. Introduction

Obviously, network connections are meanwhile broadly used as a basis or an integral part of the services realized upon them. Examples can be found in areas like internet-telephony, video-conferencing and video-on-demand, connectivity for GRID cooperation, etc. Common to all these examples is that the overall service quality directly depends on the combination of multiple Quality of Service (QoS) parameters of the underlying network connections. For example, regarding telephony those QoS parameters are bandwidth, low delay, and low jitter; video on demand – depending on the actual service model – might be more jitter-tolerant but instead requires much higher bandwidth. Network connections dedicated to the distribution of experimental data in the Large Hadron Collider (LHC) project [1] should provide dedicated bandwidth, high availability, and low maintenance time. In order to cope with a larger variety of customer and service requirements, miscellaneous QoS parameters as well as their combination should be considered during path computation (routing) for the connection setup.

Graphs are often used to model interconnected topological objects with different connection properties. Path finding in a weighted graph belongs to the classical problems of graph theory. Whereas the addition of edge-weights as an aggregation functions and the treatment of smaller resulting sum as the best path works very well in applications like path computations, e.g., for road maps, it is not applicable to connections in computer networks with multiple independent QoS parameters in general. Already the two QoS parameters considered most often, i.e., bandwidth and delay, show significant differences. Whereas the usual parameter treatment is applicable to delay, for bandwidth different functions are needed: the aggregation function is minimum and larger values are preferred to smaller results. The adequate QoS aggregation functions are significantly more complex than sum-of or minimum-of if other QoS parameters, e.g., reliability and availability, or aspects, which are relevant for service instance management, like maintenance windows for multi-domain connections, have to be considered.

Until now, in practice usually a special solution has been implemented for a service that required new or different QoS parameters. Nowadays the time for the development of a new service with customer-specific QoS parameters is becoming a crucial success factor. Therefore, a general treatment of QoS parameters is absolutely critical in order to ensure sufficiently fast adaptability and extensibility of already existing and new services. In the first place, an efficient way to distinguish between different QoS parameters is needed. Furthermore, a standardized general treatment for the aggregation of and the comparison between values of a particular QoS parameter is indispensible. As different customer-faced services might depend on different subsets of those QoS parameters, the efficient support of customer-relevant combinations of arbitrary QoS parameters is needed.

The remainder of this paper is structured as follows: In Section II the related work, which has influenced our solution is presented. The proposal for a generic treatment of different QoS parameters is described in Section III. It includes the distinction between the various QoS parameters and the definition of functions necessary during the routing process. Furthermore, we generalize the way to handle multiple QoS parameters simultaneously. In Section IV we extend our proposal in order to support value ranges as edge weights. In Section V we present how our proposal can be applied to the definition of optimized routing algorithms. A short outlook to our future work concludes this paper.

II. State of the Art and Road Map

Typically, graphs are considered that have single fixed values associated with their edges as weights. This repre-

ISBN: 978-1-61208-105-2

sentation is usually used for finding a path or a shortest path between two endpoints in a graph. However, such graphs only conditionally reflect all specifics of computer networks (see Figure 1). For instance, due to the support of different quality classes of the used network infrastructure, the property value supported by a single connection can vary in a broad range. In order to process value ranges, which are supported, e.g., by the information model described in [2], a transformation to so called *multigraphs* is possible. In the case of multigraphs, nodes may be directly connected by one or more edges. Even in a simple case with weights for a single property, such transformations can significantly increase the graph complexity. If multiple connection properties with value ranges have to be considered at the same time, the complexity increases start to be even more drastically.

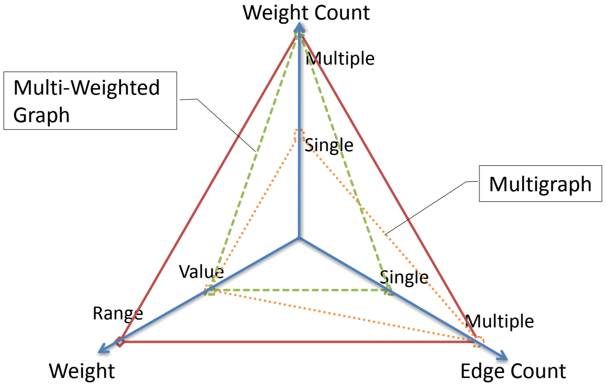


Figure 1. Graph properties, classification

Graphs that support multiple properties at the same time are known as *multi-weighted graphs*. Such graphs are hardly investigated yet. In [7], a very good overview about the state of the art is given, and various problems and solution ways are investigated. In summary, path finding in multi-weighted graphs is in general a $NP-complete$ problem. As for path finding in multi-weighted graphs the Bellman's optimality principle [8] is not fulfilled, broadly used routing algorithms that require this principle, e.g., Dijkstra's algorithm, cannot be used. Among other aspects, also the handling of multiple properties at the same time has to be solved. Currently, the common understanding is to describe multiple properties as value vectors. This allows the use of vector-addition as property aggregation operation. For a comparison of weight vectors, the concept of *non-dominance* has been established, as it is described and used in [4]. A vector A is non-dominant to vector B only if all of its weight elements, i.e., property values, are smaller or equal to the corresponding elements of vector B.

Considering operations for both single- and multi-weighted graphs, addition is used as an aggregation function and the smaller value is treated as the better one. Even if limitations of these operations w. r. t. the application to computer networks are long known to the research community, until now only workarounds have been proposed. For instance, in [9] an addition of $log(weight)$ is proposed, if the true aggregation function for $weights$ is multiplicative.

As a summary of the above discussed missing aspects, in order to enable operations on graphs describing computer networks with arbitrary supported properties, the following extensions have to be implemented:

- Support for arbitrary functions for aggregation and comparison of weights of a single connection property.
- Operations on bundles of properties, which could be used in multi-weighted graphs.
- Improved handling of value ranges.

Solutions for the first two aspects will be described in Section III. A proposal for handling of value ranges during path finding will be presented in Section IV-A.

Besides these purely technical aspects, also organizational specifics have to be considered. The so called policy-based routing, which is used for inter-domain routing, in the first place takes into account not only technical aspects, but rather provider-specific interests. Along with very restrictive information and management policies, which are out of the scope of this paper, SPs are generally interested in the reduction of resources used for a service delivery. A corresponding proposal will be given in Section IV-B.

III. Operations on Connection Properties and their Generalization

In this section we present our solutions for function generalization regarding both single properties and property bundles. The important extension for the treatment of value ranges is described in Section IV.

A. Functions for operations on a single property

During the path finding, the properties of the edges have to be aggregated. Typically, simple arithmetical addition is used as an aggregation function. As discussed in Section I, this is not necessarily the case for any QoS parameter. Furthermore, as discussed in [2], in the case of the inter-domain connections each Service Provider (SP) might have access only to its own infrastructure, which might not be sufficient to determine all the relevant connection properties. In this case also the aggregation of the partial views of involved SPs at the same inter-domain connection is needed. The calculation of QoS properties of the inter-domain link from two partial views is not necessarily identical to the aggregation of two physical connections of the same type and length. For instance, when describing a connection with the property ”delay”, not only the delay caused by the network cable should be considered, but also the delay caused by the active and passive network components used by each single SP; obviously, it typically varies between SPs.

If customer-specific end-to-end quality-of-service constraints shall be met, the value of the already found (partial) route has to be compared to these constraints during the path finding process. For path optimization it is also necessary to compare the values of two alternatives in order to choose the better one. In opposite to the case classically treated in graph theory, the meaning of what is ”better” might vary between different QoS parameters. Regarding the examples mentioned above, for bandwidth a bigger value can be considered as a better one, however for delay a smaller value is the more preferred one.

Consequently, with each supported connection property operations for value aggregation and comparison have to be associated.

B. Associating operations with properties

In IT industry, new technologies and services are evolving very fast. Therefore prior the association of operations with properties, a distinction between existing and upcoming properties is needed. We propose to assign a globally unique ID to each supported property. In order to ensure the global uniqueness of IDs, we propose to use a registration tree. Additionally to the distinction between properties, using a registration tree has another very important advantage. As multiple functions have to be associated with each supported property, it can be realized by the definition of the functions together with the registration of their property-ID (see Figure 2). Additionally this will ensure the identity of functions used among multiple SPs.

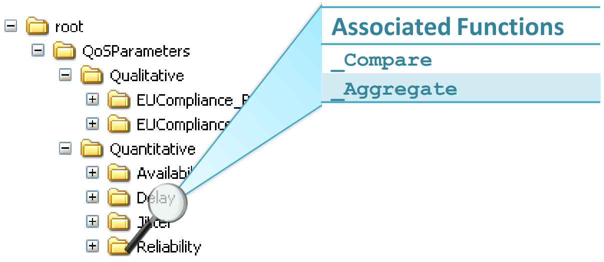


Figure 2. Registration tree example

C. Comparison and aggregation of multiple properties

Based on the previous definition, we introduce an approach for the handling of m different properties with the global unique IDs $\mathrm{ID}_1, \ldots, \mathrm{ID}_m$. In graph theory, it is a common practice to use vectors in order to describe multiple weights associated with a single edge or a path in general. For any path in a graph with m properties, the weight can be specified as $\overrightarrow{U} ::= (u_1, \ldots, u_m) \in \mathbb{R}^m$. In this definition, u_j is the weight of the j$^{\underline{th}}$ property with ID_j. The order of properties in the weight vector can be arbitrary, as long as the placement of the properties is identical among all weight vectors. Further, for the edges of a path being enumerated from 1 to n, the weight of an edge with index i will be referred to as follows $\overrightarrow{W}^i ::= (w_1^i, \ldots, w_m^i) \in \mathbb{R}^m$.

In order to calculate the weight vector $\overrightarrow{P}$ of the path consisting of n edges with weights $\overrightarrow{W}^1, \ldots, \overrightarrow{W}^n$, we first introduce an aggregation function for two weight vectors as follows:

$$\overrightarrow{\mathrm{Aggr}}(\overrightarrow{U}, \overrightarrow{V}) ::= (\mathrm{Aggr}_1(u_1, v_1), \ldots, \mathrm{Aggr}_m(u_m, v_m))$$

This definition is based on m aggregation functions for each property. The aggregation functions Aggr_i $(i = 1, \ldots, m)$ are functions associated with the property ID in the registration tree. We assume that all properties are independent of each other, i.e., they can vary without influencing the values of other properties. Furthermore, we assume that the binary operations defined by aggregation functions fulfill associative and commutative laws. Then we inductively define the computation of the whole path weight from weights of involved segments as follows:

$$\overrightarrow{\mathrm{Aggr}}\big(\overrightarrow{W}^1, ..., \overrightarrow{W}^n\big) ::= \overrightarrow{\mathrm{Aggr}}\big(\overrightarrow{\mathrm{Aggr}}\big(\overrightarrow{W}^1, \overrightarrow{W}^2\big), ..., \overrightarrow{W}^n\big)$$

Similar to the aggregation, we define the comparison of property vectors based on the comparison between identical properties. Corresponding to the *non-dominance* concept as it is described in [4], we define that vector $\overrightarrow{U}$ is better than $\overrightarrow{V}$ if and only if all properties in the first vector are better than the corresponding properties of the second vector. In order to depict that property u_i of vector $\overrightarrow{U}$ is better than the corresponding property v_i of vector $\overrightarrow{V}$, we use the symbol ”$\prec$”. In contrast to the comparison of single values, it is possible that some properties of the first vector are better and some others are worse than of the second vector. This situation should be treated as indefinite. We depict this with symbol ”$\neq$”. The comparison of two property sets can thus be defined as follows:

$$\overrightarrow{Compare}(\overrightarrow{U}, \overrightarrow{V}) ::= \begin{cases} =, \text{ if } & \forall 1 \leq i \leq m : u_i = v_i \\ \prec, \text{ if } & \forall 1 \leq i \leq m : (u_i \prec v_i \\ & \quad \vee\ u_i = v_i) \wedge \\ & \exists 1 \leq j \leq m : u_j \prec v_j \\ \succ, \text{ if } & \forall 1 \leq i \leq m : (u_i \succ v_i \\ & \quad \vee\ u_i = v_i) \wedge \\ & \exists 1 \leq j \leq m : u_j \succ v_j \\ \neq, \text{ if } & \exists 1 \leq i \leq m : u_i \prec v_i \wedge \\ & \exists 1 \leq j \leq m : u_j \succ v_j \end{cases}$$

IV. Treatment of Value Ranges

Some important aspects that are typical for computer networks are not directly addressed by classical graph theory. In this section we propose the treatment of value ranges, which can be associated with connection segments (graph edges) instead of multigraphs with multiple alternative connection segments with different fixed values.

A. Path finding with value ranges

Physical network connections usually cannot be realized with a single property set, because properties like bandwidth might vary in a wide range. In the case that an abstracted network description is considered, further connection properties can vary in a wide range. A good example is the variation of achievable delays for a single logical connection, as it can be realized by different physical connections. Consequently, also the property of the whole End-to-End (E2E) path between two endpoints might vary in a wide range. We will refer to the value range of a particular path $path$ as

$$\overset{\rightleftarrows}{W}^{path} = \left(\overrightarrow{W}^{path}_{min}, \overrightarrow{W}^{path}_{max}\right) \in \mathbb{R}^m \times \mathbb{R}^m,$$

i.e., the supported value range for the given path can vary from $\overrightarrow{W}^{path}_{\text{min}}$ to $\overrightarrow{W}^{path}_{\text{max}}$.

It is obvious that the path found between two endpoints can only be feasible if the best possible value fulfills the E2E constraints specified by customer (see Figure 3). Therefore we propose to operate with the best values of the available connection segments during the path finding process.

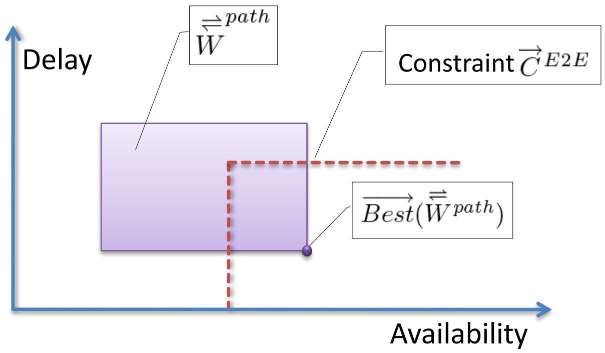

$\overset{\rightleftarrows}{W}^{path}$
Constraint $\overrightarrow{C}^{E2E}$
Delay
$\overrightarrow{Best}(\overset{\rightleftarrows}{W}^{path})$
Availability

Figure 3. Fulfillment of end-to-end constraints

We assume that all simultaneously considered path properties can vary independent of each other. Under this assumption, we define the selection function $Best$ for the best possible value of a path as follows:

$$\overrightarrow{Best}(\overset{\rightleftarrows}{W}^{path}) = \overrightarrow{Best}(\overrightarrow{W}^{path}_{min}, \overrightarrow{W}^{path}_{max})$$

$$\overrightarrow{Best}(\overrightarrow{U}, \overrightarrow{V}) ::= (Best_1(u_1, v_1), \ldots, Best_m(u_m, v_m))$$

$$Best_i(u_i, v_i) ::= \begin{cases} u_i, & \text{if } u_i \prec v_i \\ v_i, & \text{otherwise} \end{cases} \quad \text{for } 1 \leq i \leq m$$

Please note that this definition is applicable not only to a path as a whole, but also to any path segments.

B. Considering service provider interests: Optimization of resource usage

In contrast to customers, the service providers are usually interested in a reduction of resources used for service realization. This means that the requested service quality should not be the best possible one, but rather the one closest to the customer constraints. For paths complying with the E2E constraints, i.e., $\overrightarrow{Best}(\overset{\rightleftarrows}{W}^{path}) \prec \overrightarrow{C}^{E2E}$, we distinguish between three cases as depicted in Figure 4, given the weights of alternative paths A, B and C:

- All worst possible properties of the considered path are worse than the constraints (see ”Path A”)
- All worst possible properties of the path are better than the constraints (see ”Path B”)
- The worst possible properties of the path are for some properties worse and for other properties better than constraints (see ”Path C”)

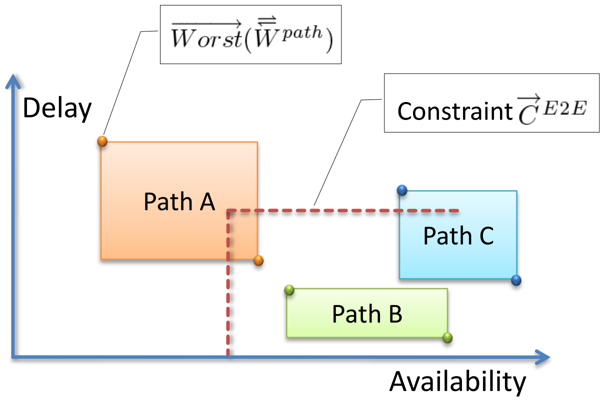

$\overrightarrow{Worst}(\overset{\rightleftarrows}{W}^{path})$
Delay
Constraint $\overrightarrow{C}^{E2E}$
Path A
Path C
Path B
Availability

Figure 4. Pathweights of paths complying to constraints

In order to distinguish between these alternatives, the function $Worst$ for the selection of the worst possible value of the found path can be defined as an opposite to $Best$.

In the case equivalent to ”Path B”, the worst possible value can be requested during the link ordering process. In the two remaining cases, an approximation to the constraint value should be performed. As the properties are independent of each other, such an approximation can be done separately (or even in parallel) for each affected property.

The whole E2E path weight is the sum of the weights of the involved parts. A possible gradation between the maximum and minimum values of connection parts is depicted in Figure 5. The E2E approximation of the path weight for a single property can be done in different ways. It can be seen as a knapsack-like problem with an intention to find a fit most close to the E2E constraint. We argue against this approach, as it may prevent the on-demand adaptation of requested service parts parameters. Instead we favor a ”fair split” among all connection parts. For each property i, we propose to use a divide-and-conquer strategy as follows:

ISBN: 978-1-61208-105-2

1) For each connection part j with a value range between w$^{j}_{i,min}$ and w$^{j}_{i,max}$ we compute values w$^{j}_{i,best}$=Best$_i$ (w$^{j}_{i,min}$, w$^{j}_{i,max}$) and w$^{j}_{i,worst}$=Worst$_i$(w$^{j}_{i,min}$, w$^{j}_{i,max}$).
2) For each connection part j we compute the realizable value $\left\lfloor \frac{w^j_{i,best}+w^j_{i,worst}}{2} \right\rfloor$.
3) If the computed path value $\sum_{j=1}^{k} \left\lfloor \frac{w^j_{i,best}+w^j_{i,worst}}{2} \right\rfloor$ is equivalent to the E2E constraint for the selected property, the selected values can be used as a result of this optimization.
4) If the computed path value is better than the E2E constraint, the computed values for connection parts should be used in the next step as w$^{j}_{i,best}$, otherwise as w$^{j}_{i,worst}$.
5) We propose to limit the number of optimization steps. If the number of maximal optimization steps is reached, the latest w$^{j}_{i,best}$ for each connection part should be used as an approximation value. If the amount of the maximum optimization steps is not reached yet, this procedure shall be repeated beginning with step (2).

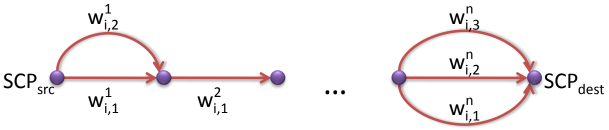


Figure 5. Possible gradation of values for different path segments for property i

Please note that in order to reflect the ”better/worthier” comparison instead of ”smaller/bigger” one, we define the unary operator ”$\lfloor\ \rfloor$” as follows: the result should be the worst realizable value, which is equal or better than the value enclosed in the brackets.

V. Application of Search Problems

In Figure 6, we present a path finding algorithm, which illustrates the usage of our operators. In the pseudo-code, a deep first search strategy is used for finding a path complying with multiple QoS constraints $\overrightarrow{C}^{E2E}$.

The presented algorithm solves the so-called *multi constrained path finding* (MCP) problem. The function requires four parameters. The first two parameters ($nodeCur$ and $nodeDest$) specify nodes in the graph, between which a path has to be found. As the MCP function is called recursively, the $nodeCur$ specifies the end of the intermediately considered path. The weight of the intermediate path is given in the third parameter $\overrightarrow{W}^{path2cur}$. Finally, $\overrightarrow{C}^{E2E}$ are always the E2E-constraints between two endpoints.

In the function, at first it is checked whether the destination node is reached yet. If it is the case, the $BacktracePath$ function is called in order to memorize the node in the path between two endpoints. Then the value $TRUE$ is returned, which signals that a path with acceptable properties has been found.

```
MCP (nodeCur, nodeDest, $\overrightarrow{W}^{path2cur}$, $\overrightarrow{C}^{E2E}$)

  if (nodeCur == nodeDest)
    BacktracePath (nodeCur);
    return TRUE;
  end if

  MarkNode (nodeCur);

  for each neighbor nodeNbr of nodeCur
    if (not Marked (nodeNbr))
      $\overrightarrow{W}^{path2nbr} = \overrightarrow{Aggr}\ (\overrightarrow{W}^{path2cur}, \overrightarrow{Best}(\overset{\rightleftharpoons}{W}^{cur2nbr}))$

      if ($\overrightarrow{W}^{path2nbr} \prec \overrightarrow{C}^{E2E}$)
        if (MCP(nodeNbr, nodeDest, $\overrightarrow{W}^{path2nbr}$, $\overrightarrow{C}^{E2E}$))
          BacktracePath (nodeCur);
          return TRUE;
        end if
      end if
    end if
  end for

  UnmarkNode (nodeCur);
  return FALSE;
```

Figure 6. Use of the new operators in a path finding algorithm

If the end node is not yet reached, the $nodeCur$ is marked with the help of function $MarkNode$. This is a common practice in DFS-algorithms, which aims to prevent loops. In the following $foreach$ loop all neighbors of $nodeCur$ are considered that have not been marked. For each neighbor $nodeNbr$ a weight $\overrightarrow{W}^{path2nbr}$ of an path between start and $nodeNbr$ nodes is computed. Corresponding to Section IV-A, the best possible value of the considered segment weight $\overset{\rightleftharpoons}{W}^{cur2nbr}$ is aggregated with the intermediate sum $\overrightarrow{W}^{path2cur}$. If the computed weight of the new intermediate path still better than E2E-constraint $\overrightarrow{C}^{E2E}$, the MCP function is called recursively. This time, $nodeNbr$ is used to mark the end of the intermediate path. If the function returns TRUE, the node is saved in order to back trace the path; subsequently TRUE is returned. If the call to the MCP function was not successful, the next neighbor has to be considered likewise. If all neighbors have been considered without any success, the node $nodeCur$ is unmarked and the value $FALSE$ is returned.

Please note that for the sake of simplicity in this algorithm at most one connection between two nodes is supported. An extension for multigraphs would require an additional loop for all edges between two interconnected nodes. Furthermore, also the back tracking function should be extended in this case, in order to track not only nodes along the path but also along used edges.

VI. Conclusion and Future Work

In this paper, we have defined a novel schema for the generic treatment of network connection properties. In order to support operations on arbitrary properties of network connections, we propose to associate five functions with the ID of every supported property. These functions are summarized in Table I. Three of these functions, which are used for property aggregation and comparison, are mandatory. The mandatory function _Aggregate_LinkPart is dedicated to compute the property of connection based on only partial views at the same inter-domain connection. For elaborated discussion about its necessity we refer to [2]. The remaining selection functions aim to simplify handling with value ranges. These functions are not mandatory, as they can be easily derived based on comparison function.

Function class	Purpose
_Compare	Compare two values *a* and *b*. Result can be: "*a* is better", "*a* is worse", "*a* and *b* are equivalent"
_Select_Best	Optional function returning the best value of a given value set
_Select_Worst	Optional function returning the worst value of a given value set
_Aggregate_Links	Aggregate property values of two links or paths
_Aggregate_LinkParts	Aggregate two partial views at the same link to a single link weight

Table I
Functions for operations on a single QoS parameter

Together with [2] and [3], which present an information model and a multi-domain routing procedure, the solution presented here is an integral part of our ongoing work, which enables user-tailored connection services with guaranteed E2E quality. However, the generic operation handling proposed in this paper is not restricted to only our work and can be used in alternative routing algorithms that are considering multiple properties, such as [5] and [6].

The presented proposal leaves some aspects unsolved; they will be addressed in our further research as follows: In the first place, a meta-language for the description of property-related functions has to be selected; also, a structure for the registration tree has to be proposed. In order to achieve this, a profound evaluation of alternatives is needed. In the case that a single global registration tree has to be used by multiple organizations, like it is the case for the internet registration tree, the description of equivalence relationships between different entries has to be addressed. Furthermore, the quality parameters of different network layers as well as user-faced services depend on the quality of the underlying layers they are realized upon. Therefore, a general description of such interdependencies and parameter transformations is essential in order to offer customer-demanded quality based on network-specific information.

Acknowledgment

The authors wish to thank the members of the Munich Network Management Team (MNM Team) [10] for fruitful discussions and valuable comments on previous versions of this paper. The MNM Team directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering is a group of researchers at Ludwig-Maximilians-Universität München, Technische Universität München, the University of the Federal Armed Forces and the Leibniz Supercomputing Centre of the Bavarian Academy of Science.

References

[1] CERN, *LHC - The Large Hadron Collider Homepage*, [Online: http://lhc.web.cern.ch/lhc/], August 2010.

[2] M. Yampolskiy, W. Hommel, P. Marcu, and M. K. Hamm, *An information model for the provisioning of network connections enabling customer-specific End-to-End QoS guarantees*, Proceedings 7th IFIP/IEEE International Conference on Services Computing (SCC 2010), pp. 138–145. Miami, 2010.

[3] M. Yampolskiy, W. Hommel, B. Lichtinger, W. Fritz, and M. K. Hamm, *Multi-Domain End-to-End (E2E) Routing with multiple QoS Parameters. Considering Real World User Requirements and Service Provider Constraints*, The Second International Conference on Evolving Internet (INTERNET 2010). Valencia, 2010.

[4] F. A. Kuipers, *Quality of service routing in the internet: Theory, complexity and algorithms*, PhD thesis. Delft University Press, 2004.

[5] T. Korkmaz and M. Krunz, *Multi-constrained optimal path selection*, Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), pp. 834–843. 2001.

[6] P. Van Mieghem, H. De Neve, and F. A. Kuipers, *Hop-by-hop quality of service routing*, Computer Networks, pp. 407–423. Elsevier, 2001.

[7] M. Ziegelmann, *Constrained Shortest Paths and Related Problems*, PhD thesis. VDM, 2007.

[8] R. Bellman, *The theory of dynamic programming*, Proceedings of the National Academy of Sciences of the United States of America, pp. 716–719. 1952.

[9] G. Bertrand, S. Lahoud, M. Molnar, and G. Texier, *Inter-Domain Path Computation with Multiple Constraints*. 2008.

[10] *Munich Network Management Team (MNM Team) Homepage*, [Online: http://www.mnm-team.org], August 2010.

A Domain-driven Approach for Designing Management Services

Ingo Pansa[1], Felix Palmen[2], Sebastian Abeck[1]
Cooperation & Management
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
[1]{pansa, abeck}@kit.edu
[2]felix.palmen@cm-tm.uni-karlsruhe.de

Klaus Scheibenberger
IT Infrastructure and Services
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{scheibenberger}@kit.edu

***Abstract*— A service-oriented software solution to flexibly support changing business environments requires the existence of an adaptable management support system. Decoupling management processes from concrete tools by encapsulating needed management functionality into management services can help meet this requirement. However, creating management solutions is a difficult and challenging task. Due to the complexity of domain management, a formal approach based on a domain model and assorted design rules would help to increase the stability of engineered solutions. Existing approaches tend to gather only at the tools level, while neglecting process requirements, which result in solutions that are hard to adapt. In this paper, we discuss the value of domain modeling to address this situation and demonstrate how designing management services by using a set of assorted design rules can be achieved. This approach is exemplified within a concrete incident management scenario.**

Keywords- domain-driven design; service design; management service; incident management

I. INTRODUCTION

With the shift towards service-oriented computing, a decoupling of needed functionality and enabling implementation has been reached. While the functionality that is needed today is derived from business requirements, enabling implementation is bound to technology. This has led to a decoupling of technology-independent business processes and technology-dependent IT systems. Business processes can now be adapted to changing requirements more easily. For instance, adding a refined debit check to typical invoice processing can now be formulated in terms of financial semantics rather than in terms of technological attributes, because needed functionality can be added by searching for debit services rather than technology-bound debit calculating software components.

Considering these extended possibilities, operational support of these services has to be adaptable as well. The challenges in realizing this are numerous [2, 15, 23]. From an IT infrastructure perspective, IT services are often created using a vast number of different computing systems running various different applications, which are interconnected by using different networking technologies. It seems all but impossible to use or create one single management tool considering all the different vendor technologies or fulfilling the special requirements that IT organizations typically have. From the perspective of a management tools supplier, it seems like it is difficult for them to ship their tools with open and generically applicable information and function models to operate the various different component technologies, because several different approaches for describing management information exist.

To increase the complexity even further, IT organizations have now started to restructure their management activities to align with best practices or standard proposals that can be derived from approaches such as ITIL [17] or ISO20000 [3]. While integration problems on the technical level within the domain of IT management always existed due to heterogeneous environments, the alignment with management processes that has been adopted lately requires the introduction of completely new tools – or at least to extend the ones that exist with the functionality to support the adoption of process requirements. Applying service-orientation in solving these issues seems to be a pragmatic, yet powerful way to both integrate existing tools and management infrastructures, as well as to align with management processes. Therefore service-orientation not only promotes adapting to new requirements more easily, but also the ability to reuse existing components.

In order to utilize the principles of service-oriented computing and service-oriented architecture to solve these challenges in managing IT services, a clearly defined development approach is needed. This development approach has to consider today’s standards and best practices, as well as how to integrate existing management tools . Although some work that uses the application of service-orientation to construct management systems that are based on loosely coupled management services exists [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], little has been done to tackle this challenge on a conceptual level by focusing on the reusability and adaptability of future management systems.

This paper proposes an improved meta-model for the domain of process-oriented IT Management and an approach to applying the meta-model for the purpose of constructing reusable and adaptable management services. The approach is presented in the form of rules that allow a repeatable development method. Focusing on a domain meta-model that is built on the requirements of process-oriented standards enables constructing services that are aligned with the processes that the services are intended to support. Furthermore, these services serve as a central point for constructing integrative adapters to existing management tools. A formalized model enables the construction of

ISBN: 978-1-61208-105-2

development supporting tools, therefore the design of the management services system can be performed based on computational support. In our opinion, understanding the semantics of the terms the domain IT management is faced with is crucial in the construction of such a service-oriented management system. Therefore, having modeled the structure of the domain IT management for the purpose of deriving management services is a fundamental part.

The remaining parts of this paper are structured as followed: Section 2 introduces related work and provides the background for constructing a management platform that is built on service-oriented principles. In Section 3, we discuss the value of modeling the domain for a proposed solution and present an extension to the domain meta-model presented in [1]. Section 4 presents the benefits of this paper: We embed a domain-driven and rule-based development method into a typical software development process, demonstrate domain modeling applied to a typical management activity performed within incident management and introduce an assorted set of rules to support the domain-driven derivation of management services. Section 5 describes our experience with a prototypical implementation, where we applied the proposed method within a real world scenario. Finally, Section 6 concludes this paper and gives an overview of the work that is currently being done within our research group.

II. Background and Related Work

The efforts of recent years to structure and tackle the complexity that management solutions are faced with have led to standard specifications for the definition of IT service management processes. The most prominent representative is ISO20000-1:2005 [29]. This particular standard definition presents a taxonomy introducing minimum functional requirements that implementations of different management solutions have to fulfill. While ISO20000-1:2005 is mainly based on the best practice suggestions presented by the Information Technology Infrastructure Library (ITIL, [16]), a clear and formal representation of the proposed entities, activities or participants, is still missing. Nevertheless, since [29] introduces the elements of the domain, our aim is to construct software solutions, where the standard definition serves as one input for the creation of a commonly accepted ontology, with which a formalized meta-model could be developed.

Applying service-orientation to solve integration issues (adaptability, reusability) is assumed to be one feasible approach [2, 17, 22]. Based on the suggested best practices of the ITIL [16], much research focusing on the construction of service-oriented management solutions has been done. Tamm and Zarnekow [12] derive web services from a typical definition of an incident management process, but neglect the domain as part of this process. It seems difficult to give any statement regarding the adaptability or reusability of the solution that they present.

Mayerl and Abeck et al. focus the integration of existing management tools along process-oriented management scenarios [14, 15]. Although a systematic development method is proposed, neither domain modeling nor specific rules for designing management services are discussed. The approaches presented in [14, 15] are rather general and do not consider formal aspects. Furthermore, standard requirements are neglected. In [13], an automated management process is implemented based on web services, but both a structural analysis, and a systematic method are missing.

Aschemann and Hasselmeyer deal with the principles of a service-oriented architecture in supporting management systems [4]. While both domain modeling and systematic development methods are missing, at least some architectural guidelines can be concluded from their work. For instance, different components enabling communication between management services are needed. Furthermore, some functionality enabling the location and lookup of existing management services is also needed. Anerousis discusses an architecture for building scalable management services [5], however it lacks formalization of the given domain. Lu et al. examine management services on the managed resource level [7, 8, 9], but not a systematic and overall development method, which integrates both process and resource requirements. Standard specifications such as WS-Management [10] or WSDM-MUWS [11] only deal with the managed resource level.

Different approaches for introducing a service taxonomy have been suggested [18, 19]. Based on these ideas, we have observed that it is possibile to clearly distinguish different types of management services, which is why we refer to management basic services and management process services when clarifying different characteristics of these services.

Our proposed design method of applying some assorted rules for deriving services is similar to the one presented in [20], but extends it by capturing not only elements of business process models, but also model instances of an overall domain analysis model. A good overview of different approaches for domain analysis can be found in [21]. The authors argue that although many different approaches for constructing domain models exist, software systems that support different problem domains differ in many aspects, which is why there is no existing modeling approach that is suitable for every kind of scenario. Based on their evaluation results, our approach is based on functional decomposition using rule support for creating both domain models and designs for management services.

III. A Domain Met- Model for IT Management

Describing the semantics of the elements of a domain serves as a building block for developing a software solution that is tightly aligned to its requirements. In this section, we give a summary of our motivation for using domain modeling and present initial and ongoing work in the area of development tool support for creating standardized and formalized models for services within the domain IT Service Management.

A. *The value of domain modeling in IT Management*

As the process of creating software systems becomes more complex, formal descriptions are required to engineer these systems. This can be saidfor any of the disciplines, from eliciting the requirements that a solution has to fulfill to creating detailed models describing the structure of the software architecture or aspects concerning control flow. However, formal descriptions are hard to achieve. One building block is the description of the semantics of the single elements of the domain that the desired solution will use. Identifying this information in relation to similar problems leads to a classification schema, which can easily be reused. This is referred to as domain modeling. Domain modeling is a pragmatic approach utilizing modeling techniques that are well understood. Creating a domain model has several advantages, one of the most important ones might be the fact that software systems derived from domain models show a higher degree of reusability if extensions to these software systems are created from the same domain models.

Nevertheless, a domain model has to be abstracted in some way to really be adaptable. Therefore, we decided to create a domain model that is based on ISO20000-1:2005 [29], which was enriched with some of the typical patterns of the Information Technology Infrastructure Library [16] best practices and existing process modeling approaches, for instance a separation of atomic and composed activities can be found in the Workflow Management Coalition Meta-Model [30]. Using such a domain model allows designing management services that are tightly aligned with the domain that the services are intended for. Furthermore, automated design evaluators can be constructed measuring the overlap of domain model instances and the instances of service models, thereby allowing for automated design decision support.

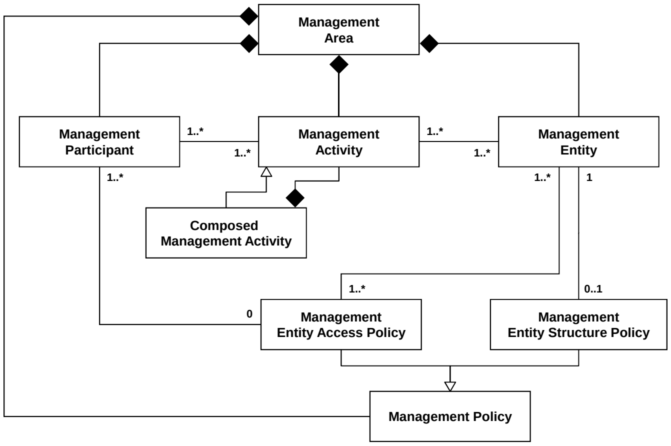


Figure 1. Conceptual Meta model of the Domain IT Management

A domain analysis method always consist of two things [21]: an ontology along with a taxonomy of this ontology defining a meta-model of the domain, and a process that allows the construction of model instances of this domain. The concepts of the Domain Meta-Model have been introduced in [1]. (see Figure 1). A Management Area contains Management Participants, Management Activities, Management Entities and also Management Policies, which can be refined to policies to define access to specific entities or policies to define the structure of entities. Management Activities can be refined to Composed Management Activities, e.g. a Management Area defines one single functional area such as Incident Management. Aiming at defining services aligned with models instantiated from this meta–model. Both the modeling element Management Activities and Management Area can be considered to offer capabilities which are independent of concrete realizations. These capabilities are later used to derive management services. To capture this aspect within the meta-model, the meta-model is simply extended as depicted in Figure 2.

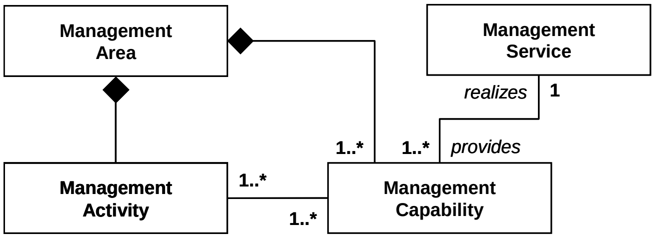


Figure 2. Extension for Deriving Management Services

Based on this domain meta-model and its extension, an assorted set of rules underpinning our development method is presented within the next section.

B. Rule-based derivation of management services

Focusing a refined derivation of management services by considering design decisions dedicated to the domain of IT management, a discussion of specific elements of the meta-model is necessary. The meta-model used so far provides elements for modeling management capabilities and the conceptual management services providing them. Those services are needed as part of the domain model because by considering them as elements of the domain, a clear semantic relation to the management capabilities and management activities is possible.

Of course, the sources used to model the domain do not mention management services because these are not part of the plain management view, but added to the domain by the decision to support management processes with a service-oriented architecture. Therefore, since adding management activities, management areas, management participants, etc. to the model directly from the text of e.g. ISO20000-1:2005 is a straightforward process, modeling management capabilities and management services involve design decisions. Therefore, in order to achieve reproducible results when modeling the domain, derivation rules are needed for naming and coupling management capabilities with management services. This repeatability is needed to preserve the value of the domain model, such as the increased degree of reusability in the resulting service design.

The conceptual management services given in the domain model do not yet support an implementation of a service-oriented architecture. We still need a service model that describes the services in detail with all service operations and their signatures. For such a service model, a specialized UML-derived modeling language like SoaML

[22] is a natural choice. Again, a requirement for the service model is repeatability, so we will introduce more derivation rules for the transition from the domain model to a SoaML model of the final service interfaces.

In order to maximize the benefit of our approach, these rules must be defined carefully. One important factor is consistent naming, so that the semantics of a service and its operations can be understood by looking at the domain model. Therefore, rules for naming are strict and do not leave room for individual decisions. Another aspect to keep in mind is reusability of individual management services. By defining rules that result in the modeling of exactly one service per management entity, the resulting services have few interdependencies and a single service can easily be exchanged, for example by using an adapter for an existing management tool.

IV. Applying Domain Modeling for Desiging Management Services

For demonstrating the value of domain modeling, a concrete scenario is presented. The next sections address a typical incident management process that serves as an input artifact for deriving adaptable yet business-aligned services. Although our approach refers to the term management process, a structural analysis of the domain is performed that not only take the dynamic parts of a process into account, but also the static relationship of domain elements involved within this process. Following this approach, the derived services can be reused in further development efforts if extensions to an existing system are necessary.

A. *Overall development method*

Since structured development methods for the purpose of deriving and designing a service-oriented software solution are common today, we briefly describe the necessary steps to perform in order to create a set of management services that are aligned with a model of the domain.

First of all, an analysis of the standard specification for process-oriented IT Service Management ISO20000 leads to an overview of the activities, entities and participants that constitute the management capabilities of one management area, for instance incident management. Adding policies (entity structure policies and entity access policies), the elements of an overall domain model are given.

Within the next step, these domain elements are modeled. To avoid misconceptions of the relationships of these elements with each other, a formal meta-model is needed that clearly defines the syntax and semantics of each single domain element. Such a meta-model can be found in [1].

As an improvement to the approach presented in [1], the derivation of services is performed using an assorted set of rules. These rules take several aspects of a model-to-model transformation of an instance of the domain model to an instance of a model of management services into account. Since the design of services is a highly complex task, in which several design decisions have to be made, we propose a two-step approach, stemming from domain models to a model of service candidates and finally a model of technology-independent service interface descriptions. Introducing such a two-fold step enables grouping several service candidate operations into combined service definitions if a service within a given management system already exists.

B. *The Incident Management Process*

The Incident Management Process is one of the critical processes dealing with service disruptions. Incident Management is established in nearly each service provider’s organization offering defined IT services to customers utilizing IT services for supporting IT-based business processes. Since customers are directly faced with incident management for service failures, providers are interested in controllable execution of this process. As business requirements change requirements for supporting services management support and flexible management components are needed. According to [24], the Incident Management Process has a high recurrence rate and a high organizational structure. This process is mostly well suited to workflow support, which we intend to realize using a service-oriented architecture.

ISO20000:1-2005 defines the objective of the incident management process as the ability "to restore agreed upon services as fast as possible or to respond to service requests" [3]. In analyzing the definition of incident management, the following elements of the domain model can be identified:

Entities: Incident Record, Workaround Record,

Participants: First Level Support, Second Level Support

Activities: manage impact, record incidents, prioritize, determine business impact, classify, update, escalate, resolution and formal closure of all incidents.

Policies: Incident Entity Structure Policy, Incident Entity Access Policy

Having identified these elements, a formal model of the domain can now be constructed. For the sake of simplicity, we will look at the assorted management activity Prioritize Incident that is used to determine the impact of a service failure and to add a priority value to the related incident record.

C. *Modeling one Management Activity*

Modeling of a management activity in the ITSM domain model is done in several steps. First, sub-activities are identified from the ISO20000-1:2005 text. The definitions of other management processes are also looked at in order to pick up interconnections with other processes. In the second step, known patterns and principles of management architectures, such as OSI management [25], WBEM [26], etc., are considered in order to find matching sub-activities that directly model the usage of functionality provided by existing components. Finally, the management capabilities needed to perform these management activities are modeled and the conceptual management services providing these capabilities are defined.

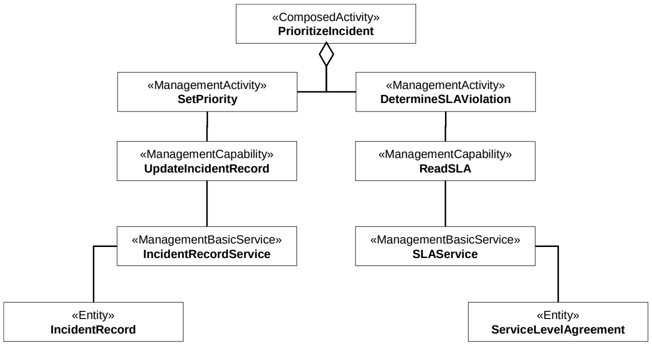


Figure 3. Management Activity PrioritizeIncident with process-related sub activities

In order to achieve a consistent model, rules are applied. The notation [Entity] in these rules puts the name of the entity instance here.

(Rule 1) Management capabilities that create, read or update an entity are named as follows: Create[Entity], Read[Entity] and Update[Entity].

(Rule 2) All management capabilities operating on an entity are provided by a single management service called [Entity]Service.

(Rule 3) Management capabilities that communicate an entity to another participant are named Send[Entity] and Receive[Entity].

(Rule 4) All management capabilities communicating an entity are provided by a management service called [Entity]TransferService.

(Rule 5) For all management capabilities that are not provided by a service after applying rules (Rule 2) and (Rule 4), a management service is introduced per management activity and named [ManagementActivity]Service.

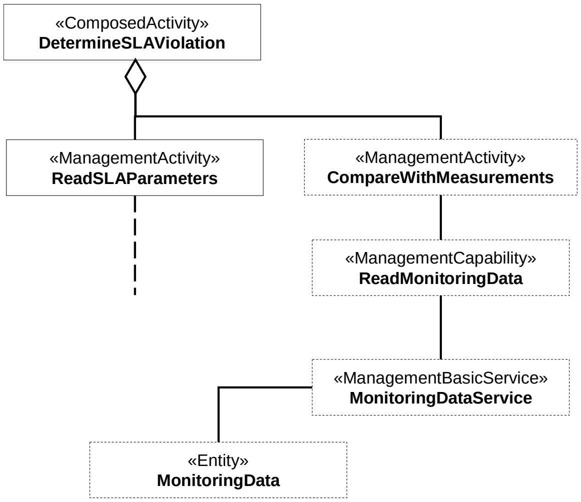


Figure 4. PrioritizeIncident - technically motivated sub activities

Figure 3 illustrates how these rules are applied to the management activity prioritize incident. The two sub-activities are both process motivated, found from the requirements given in ISO20000-1:2005. The management capabilities and services are named according to rules (Rule 1) and (Rule 2).

In Figure 4, one sub-activity is further extended, exploiting the fact that all major management architectures provide methods to read measurement data of managed components. The naming of the management capability and the management service is again done using the rules (Rule 1) and (Rule 2).

D. *Designing management services*

After the domain model was used to identify the conceptual management services needed and the management capabilities they should provide, the services can be modeled using SoaML. This is done by using some more transformation rules.

(Rule 6) Each conceptual management service in the domain model translates to a SoaML Capability of the same name.

(Rule 7) A management service found by applying rule (Rule 2) is given “CRU” (Create, Read, Update) operations named Create[Entity], Read[Entity] and Update[Entity]. A delete operation is intentionally left out because deletion of entities is never done according to ISO20000-1:2005.

(Rule 8) A management service found by applying rule (Rule 4) is given to the operations Receive[Entity] and Send[Entity].

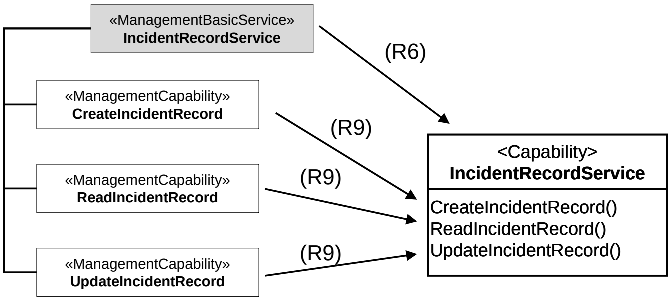


Figure 5. From the domain model to the SoaML Capability

(Rule 9) Operations are added to each SoaML Capability, according to the management capabilities this service should provide, as long as they are not already present after applying rules (R7) and (R8). The operations are given the same names as the management capabilities they should support.

(Rule 10) A SoaML ServiceInterface is created for each management service that exposes the corresponding SoaML Capability.

(Rule 11) A SoaML DataType is created for each Entity with the data fields given by the corresponding entity structure policy.

(Rule 12) The operations Update[Entity] and Send[Entity] are given an input parameter of type [Entity] (the SoaML DataType).

(Rule 13) Operations Create[Entity], Read[Entity] and Receive[Entity] are given an output parameter of type [Entity].

(Rule14) The operations Read[Entity] are given an input parameter of type String, named [Entity]ID. The [Entity]ID is the unique identifier for an entity.

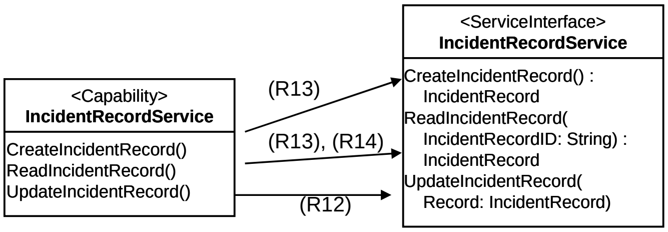


Figure 6. Designing the ServiceInterface

Figure 5 illustrates rules (Rule 7) to (Rule 9) as applied to the IncidentRecordService. In Figure 6, rules (Rule 10) to (Rule 14) are applied in order to model the final ServiceInterface for the IncidentRecordService.

E. Integrating existing tools

In the previous sections, the integration of existing management tools was prepared by introducing management activities motivated from common concepts of management architectures. Therefore, to actually integrate an existing tool, all that has to be done is that the appropriate management basic service that exposes the capabilities provided by this tool has to be found.

For example, a tool like Nagios, which focuses on the technical management of components, provides everything needed for the "MonitoringDataService" shown in Figure 4. In order to integrate this tool, development of a webservice adapter is needed, so that the tool exposes its functionality according to the service interface modelled following the rules given in the previous section. Different methods for the development of webservice adapters were suggested before, e.g., in [27], where the authors introduce the concept of mismatched patterns between an existing and a needed service interface. It will be up to the implementing organization to choose the method that best suits their existing tools and requirements.

V. IMPLEMENTATION EXPERIENCE

A discussion of the applicability of our approach includes both a presentation of the achieved results and a generic estimation of the benefit of our method. To address this, in this chapter we briefly present the artifacts that where created along an integration project we currently run at the ATIS, a mid-sized service provider that operates the IT infrastructure in responsible for the faculty of informatics at the Karlsruhe Institute of Technology.

One major goal of this project is to create an integrated management platform that enables both users of the provided IT services and the operators of these services to access relevant management information in one web portal. Furthermore, interfaces to provide management functionality should be created that can be used by external providers connected to the network of the ATIS. During the analysis of the actual situation it became obvious that in order to fulfill these three integrative requirements, a supporting architecture needs to be flexibly adaptable thus architectural elements that are highly reusable had to be engineered. As indicated by some internal examinations, the handling of service disruptions was one of the most urgent use cases that should be implemented at first.

According to ISO20000-1:2005, handling of service disruptions is performed by the Incident Management and Problem Management Process, which in turn is supported by Configuration, Change and Release Management Processes. While the Incident Management deals with restoring disrupted services as fast as possible Problem Management concerns itself with investigation of the root causes leading to recurring service failures. In orderto reduce complexity, we decided to first implement the Incident Management Process, followed by an implementation of Problem Management that can be realized based on the services we identified during the design phase for Incident Management.

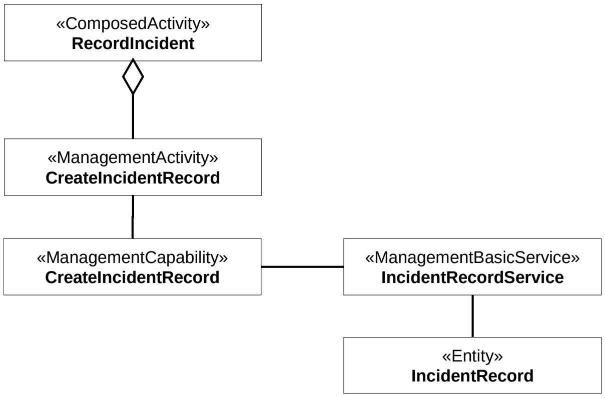


Figure 7. Domain Model for IncidentRecordService

Figure 7 shows an excerpt of the domain model that serve as a starting point for designing management services for the Incident Management Process. As mentioned in Section IV.B, the elements of the domain model can be identified using the definition of Incident Management given in [29].

While the construction of the domain model is fairly straightforward, applying the transformation rules to design the service models currently requires detailed knowledge of the semantics of the modeling elements. Tool support would be highly desirable in this step in order to minimize failures due to semantical misunderstandings. Nevertheless, it turned out to be useful to reflect on the derived service models by both the supporting analyst and the developers alike.

In Figure 8, the results of the transformation applying the rules given in section IV.C and IV.D to the domain model of the IncidentRecordService is given. For instance, applying rule 7 extends the capabilities of the identified operation CreateIncidentRecord with the respective Read and Update operations. As the initial milestone of the development project was rolled out and the additional requirement to implement the Problem Management Process came up, we could reuse the models designed so far and further use service functionality that was already considered during the design phase of the Incident Management services.

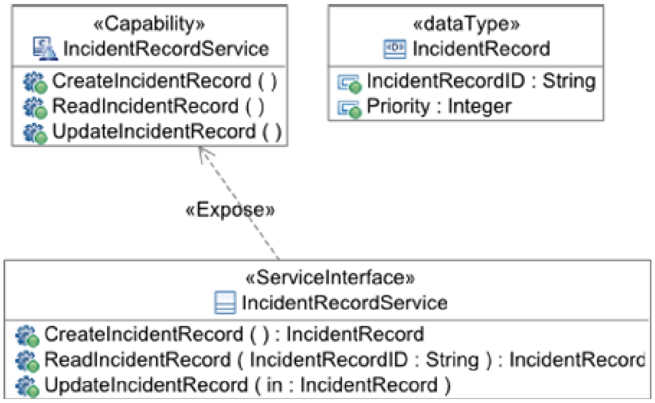


Figure 8. Domain-driven Design of a Management Service for Incident Management

Finally, these designed service interfaces can be implemented using Web Service Definition Language, as shown in Figure 9.

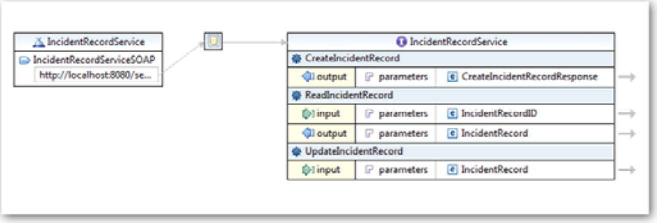


Figure 9. Implementation of the IncidentRecordService using WSDL

In reviewing the lessons learned this small excerpt of the development project shows that application of the method leads to a standardized vocabulary thereby enabling reuse of existing design models when extending systems to support further requirements. The long-term benefit is grounded in the fact that the more the processes will be implemented, the higher the degree of reusability of existing services will be. In order to estimate complexity, we are currently investigating an approach for creating a domain meta-model-based reference model, which includes relations of management functions of different management services for different management processes. This would allow determination of the best starting point for a concrete development project if requirements are clearly given. Figure 9 outlines an early result that served as a basis for our decision to initially implement Problem Management followed by Incident Management.

To sum up the additional overhead of introducing formal domain modeling fairly, we feel that at least basic skills in structural modeling are needed. As the domain meta-model and the related specific instances only make use of classes and their relationships and the fact that the transformation rules were given in natural language, acceptance by the developers involved was surprisingly high. This was because we could show the benefit of formal domain models when extension of given systems was focused. Since the example shows that some of the management basic services needed to implement Problem Management were already identified during analysis of the Incident Management Process, we expect that the implementation of further requirements (Release Management Process, Change Management Process, and Configuration Management Process) will take even less time in terms of c reated design models.

VI. CONCLUSION AND OUTLOOK

In this paper, we motivate the advantages of applying a structured and well-founded development approach in the design of adaptable management services. Since business requirements are constantly changing, the support of management systems for operating IT services has to be able to keep step with this development. Therefore, we promote organized management functionality in terms of loosely coupled management services to enable both a controllable execution of management processes and to adapt to changing requirements.

One of the most critical questions regarding the design of services is to ensure that certain design principles are met. For instance, services are expected to be technology independent, reusable, accessible with defined interfaces and protocols or aligned with business requirements. A major goal of our approach is to support developers of management systems to be supported with some typical engineering instruments enabling evaluation of their management services design against these typical service principles. To reach this goal, we introduced a development approach that is driven by a sound understanding of the domain IT Management. While key concepts of a meta-model for the domain were already introduced in [1], in this paper these key concepts have been revisited in order to integrate an automated verification of derived management services. Furthermore, we extended the development approach in [1] by applying a model-driven approach for designing concrete interfaces for management services. These interfaces can be used to both implement new management services or in scenarios where an integration of existing management tools is necessary, to create integrative adapters to legacy applications. A concrete outcome of our work is a defined set of management services that are needed to execute a typical incident management process. This set of services is aligned with the domain and fulfills several critical service principles, therefore we expect that not only can our conceptual contribution be applied in further scenarios, but also vendors of concrete management tools capture can verify the capabilities of their tools to evaluate the alignment with standard requirements.

Some of the aspects that are presented in this paper need to be discussed further. For instance, a formalized meta-model would allow creating development supporting tools, and simplifying the derivation of services as model-driven techniques could be used. Currently we are exploring approaches to formalizing the presented derivation rules, such as using Object Constraint Language (OCL) [28]. This would be integrated with a formalized meta-model to support a model-driven approach enabling the automated derivation of management services. Furthermore, we are currently considering the integration of existing management tools that would serve as implementation for some management services. For instance, in a typical provider scenario, it is very likely that at least trouble ticket tools exist to coordinate the execution of the incident management process. Creating integrative service adapters in a bottom-up driven way would allow both reuse of existing tools and thecreation of flexible support for workflow support of the management process. We expect that our concept of a combination of domain modeling and rule-based derivation of services can be applied in domains other than IT management.

REFERENCES

[1] I. Pansa, P. Walter, K. Scheibenberger, and S. Abeck, “Model-based Integration of Tools Supporting Automatable ITSM Processes”, Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, Page(s): 99 - 102

[2] V. Machiraju, C. Bartolini, and F. Casati, "Technologies for Business-Driven IT-Management", Proc. Extending Web Services Technologies: the Use of Multi-Agent Approaches“, edited by Cavedon, L., Maamar, Z., Martin, D. and Benatallah, B., Kluwer Academic

[3] G. Aschemann and P. Hasselmeyer, "A Loosely Coupled Federation of Distributed Management Services", Journal of Network and Systems Management, Vol 9, No. 1, 2001, pp. 51-65

[4] N. Anerousis, "An Architecture for Building Scalable, Web-Based Management Services", Journal of Network and System Management, Vol. 7, No 1., 1999, pp. 73-104

[5] C. Xiao, Z. Lv, and S. Zhang, "WS-CHMA: A Composite-Pattern Based Hierarchical WS-Management Architecture", Services - I, 2009 World Conference on (2009) pp. 773 – 780

[6] Z. Lu, Y. Wu, C. Xiao, S. Zhang, and Y. Zhong, "WSDSNM3: A Web Services-based Distributed System and Network Management Middleware Model and Scheme", The 9th International Conference for Young Computer Scientists, ICYCS 2008, 2008, pp. 392-397

[7] Z. Lu, J. Wang, Y. Wu, J. Wu, and Y. Zhong, "MWS-MCS: A Novel Multi-agent-assisted and WS-Management-based Composite Service Management Scheme", IEEE International Conference on Web Services, ICWS 2009, 2009, pp. 1041 - 1042.

[8] Z. Lu, J. Wu, S. Zhang, and Y. Zhong, "Research on WS-Management-based System and Network Resource Management Middleware Model", IEEE International Conference on Web Services, ICWS 2009, 2009, pp. 1051 - 1053.

[9] World Wide Web Consortium (W3C), "Web Services for Management (WS-Management)", Version 1.1.0

[10] Organization for the Advancement of Structured Information Standards (OASIS): Web Services Distributed Management (WSDM) - Management Using Web Services

[11] G. Tamm and R. Zarnekow, "Umsetzung eines ITIL-konformen IT-Service-Support auf der Grundlage von Web-Services", 7. Internationale Tagung Wirtschaftsinformatik 2005 (Bamberg), 2005, pp. 647-666

[12] A. Brown and A. Keller, "A Best Practice Approach for Automating IT Management Processes", Management of Integrated End-to-End Communications and Services, Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium, NOMS 2006, Vancouver, Canada, April 3-7, 2006, pp. 33-44

[13] C. Mayerl, T. Vogel, and S. Abeck, "SOA-based Integration of IT Service Management Applications", Proceedings IEEE International Conference on Web Services 2005, pp. 785-787.

[14] C. Mayerl, T. Tröscher, and S. Abeck "Process-oriented Integration of Applications for a Service-oriented IT Management", The First International Workshop on Business-Driven IT Management, 2006, pp. 29-36

[15] J. Sauve, A. Moura, M. Sampaio, J. Jornada, and E. Radziuk, "An Introductory Overview and Survey of Business-Driven IT Management", BDIM '06. The First IEEE/IFIP International Workshop on Business-Driven IT Management, 2006, pp. 1-10.

[16] Office of Government Commerce (OCG): IT Infrastructure Library (ITIL) – Service Support (ISBN 0113300158), 2000; Service Delivery (ISBN 0113300174), 2001; Planning to Implement Service Management (ISBN 0113308779), 2002; Application Management (ISBN 0113308663), 2002.

[17] V. Tosic, "The 5 C Challenges of Business-Driven IT Management and the 5 A Approaches to Addressing Them", Business-Driven IT Management, 2006. BDIM'06. The First IEEE/IFIP International Workshop on Buisness-Driven IT-Management, 2006, pp. 11-18.

[18] T. Erl: SOA Principles of Service Design. Prentice Hall Service-Oriented Computing Series, 2008.

[19] S. Cohen, "Ontology and Taxonomy of Services in a Service-Oriented Architecture", The Architecture Journal, Volume 11, 2007

[20] M. Gebhart and S. Abeck, “Rule-based service modeling”, Fourth International Conference on Software Engineering Advances, 2009. ICSEA '09. , 2009, pp. 271 - 276

[21] X. Ferré and S.Vegas, "An Evaluation of Domain Analysis Methods", In 4th CAiSE/IFIP8.1 International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD99)

[22] Object Management Group, “Service oriented architecture Modeling Language”, http://www.omg.org/spec/SoaML/1.0/Beta2/, 04/2009 (last visited 08/10/2010)

[23] V .Machiraju, C. Bartolini, and F. Casati "Technologies for Business-Driven IT Management", In: Extending Web Services Technologies: The Use of Multi-Agent Approaches (Multiagent Systems, Artificial Societies and Simulated Organizations), Springer, 2005, pp. 1-27.

[24] M. Brenner, "Classifying ITIL Processes – A Taxonomy under Tool Support Aspects", Proceedings of First IEEE/IFIP International Workshop on Business–Driven IT Management (BDIM 06), pp. 19–28, April, 2006.

[25] International Standards Organization, “Open Systems Interconnection – Basic Reference Model – Part 4: Management framework”, ISO/IEC 7498-4, 1998

[26] Distributed Management Task Force: Web-Based Enterprise Management (WBEM), http://www.dmtf.org/standards/wbem/ (last visited 08/10/2010)

[27] B. Boualem, F. Casati, D. Grigori, M. Nezhad, and F. Toumani, “Developing Adapters for Web Services Integration”, In Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE), Porto,Portugal CAiSE, 2003, pp. 415-429

[28] OMG, “Object constraint language”, Version 2.0, 2006.

[29] International Standards Organization, "Information technology — Service management —Part 1:Specification, ISO/IEC 20000-1, 2005

[30] The Workflow Management Coalition, "Workflow Management Coalition Terminology & Glossary", 1999

Studying in Web 2.0 – Virtual University as Virtual Community

Birgit Feldmann
Department of Information Systems and Databases
University of Hagen
Hagen, Germany
birgit.feldmann@fernuni-hagen.de

***Abstract*— Our experience and research on technology-supported learning and teaching have clearly confirmed the general understanding that students working collaboratively are more successful than students working alone. Therefore, it should be a logical consequence to integrate communication and collaboration as a key factor into a distance study environment. However, this is not a trivial task from various points of view. For instance, for public universities in Germany studies have to be free of charge – which then raises the question, how to finance highly interactive small classes? Another problem is the professional restrictions of working distance students: their time budget is very limited. The consequence is that students typically have very limited contact to their peers and their tutors until the final examinations. The drop-out rates have been extremely high. A lot of students study more than six years to reach a first degree. E-learning improved the situation substantially (even though poorly used by the teachers in many environments), but by far not enough. This paper shows how Web 2.0 opens up new possibilities to approach these challenges, and how it can be used to improve the situation substantially.**

Keywords - group types; e-learning; Web 2.0; collaborative learning; Virtual University.

I. INTRODUCTION

Distance study systems face fundamental problems like isolation of students and finding a compromise between requirements of private and professional life and studying [13]. To improve the situation the University of Hagen (FernUniversitaet), a distance teaching university with about 60.000 students, started to develop a Virtual University (VU) in 1996 [19]. The new form of teaching and learning through the Virtual University eased the situation of the distance students remarkably, but there remained a lack of social interaction and group-awareness. Various research projects as well as our own experience clearly show that being part of a group and having suitable communication partners lead to higher and more consistent motivation and therefore to more successful and faster studies [1, 2, 6, 7, 13]. An additional effect is that organizational support by the university gets less critical as students can easily, and very directly assist each other. This, in turn, reduces overhead at the university. A survey at our university also showed that most students are convinced that contact to fellow students, especially through different types of groups, is of utmost importance for successful learning [3]. They are not satisfied with the existing system and call for new and better communication and group support [3].

The obvious conclusion of these observations is that a new learning environment is necessary. The kernel concept of the vision developed here is to start out from the students' view and research results as described above – which is completely different from the classical approach to deliver content and to have group elements and communication as an add-on. To build this platform the integration of Web 2.0 technologies is essential. To provide such a new and community oriented environment we have to look closer into the various fields of groups and their mechanisms with the goal to support these groups with the necessary technical and organizational features. The necessary first step is to investigate the different group types and their meaning in a distance teaching setting. The paper exemplifies some group types through already existing or forming groups in the e-learning environment at the University of Hagen. These group concepts, their properties, the way they are used by students, their overall potential are the main topics of this paper. A detailed discussion of a complete e-learning system centered on social and community aspects cannot be given here due to space-limitations; further research is going on about how to build this kind of system.

The paper is structured as follows: Section II contains the state of the art, The following section III investigates the various group types relevant in a distance teaching environment and the consequences for a new type of learning system. Section IV describes the current situation and developments for group support at the University of Hagen. The paper concludes with a short summary of the findings and an outline for further necessary research.

II. STATE OF THE ART

Schulmeister [16] not only evaluated 23 existing studies about learning management systems but also undertook his own research about more than 62 learning management systems. He concludes that existing learning management systems typically focus on delivering content; they do not support building and establishing long-lasting student groups, or – if at all – they do it very poorly [16]. This correlates with our own results as only 19% of our students use the integrated communication features and only 2% the groupware functionalities [3]. If group oriented features are available, they are provided only for advanced students in the context of the provided content. These results are confirmed by research of Kerres [7].

Today, the importance of collaborative learning and working is without controversy in the research community [6-11, 17]. But the group processes and the various categories of group types in a more general meaning are still not well understood as will be investigated in section four of this paper, [6, 15, 17, 18]. Some valuable insights can be found in the field of community oriented learning [5, 8, 9, 12. 14]. However, the community types discussed in this field, like learning community or community of practice, do not sufficiently cover the needs of distance learning students as they are either too strictly structured (e.g. restricted to an exactly defined group of students like in classes) or just the opposite, they have no structure at all. Some essential group types are not considered.

Many different definitions of "group" exist in different disciplines (computer science, psychology, sociology etc.), but none of them clearly describes the different existing group types in distance education from a practical point of view [5, 6, 12, 17].

III. The Virtual University

The University of Hagen has extensive experience in distance education since more than 30 years. Therefore, it was a logical consequence to start using the Internet for learning and teaching purposes at a time when not many universities saw the opportunities. The benefits of distance education, time- and location independence, and the advantages of the Internet (fast information, easy communication and cooperation possibilities) were combined in the project Virtual University, Germany's first university to offer all its services in the Internet beginning in 1995.

When the project started a suitable software platform was not available in the market, so a platform was built based on Internet technologies and a commercial database system. The virtual university system now integrates all functions of a university into a complete, homogeneous, extensible system with an easy to use and intuitive student-centred user-interface. Currently, about 60.000 students are using the Virtual University of Hagen. The ongoing project includes experimenting with and evaluating different forms of teaching and learning in the Internet.

It turns out that the most popular teaching events are highly interactive events like virtual seminars, practical training and online exercises. Our experience shows that teaching methods with a high rate of group activity using electronic communication have the quality to break down the students' isolation. Students using electronic communication within the virtual university are much more motivated than before; the dropout rate is decreasing and the feedback is entirely positive [1, 2, 10, 13].

We also learned that some of the groups students took part in during a teaching event lived and were active long after the event itself. In some cases the relationship between the group members established during a teaching event in the internet was stable long after the students finished their studies [1, 2, 10]. Others researchers, i.e., Leh [8], Palloff and Pratt [12] also state the importance of communication and cooperation.

The expectations of students regarding the teaching and learning environment have dramatically changed over the last years. In a study with more than 2000 participants we found that about 43% of our students want more Web 2.0 functionality in the university learning environment [3]. This includes communities, blogs, wikis and especially social networking. Related research is consistent with these findings, e. g., Kerres [7] and Peters [13].

Motivated by these results we have been looking into the various types of groups and their mechanisms with the goal to support the initializing of groups and to secure their stability and liveliness.

As a first step we investigate the different group types already existing in e-learning environments. As an example we use the University of Hagen.

IV. Group Types

The importance of collaborative learning and working is meanwhile without controversy in the research community [611]. But the description of group processes and group types in a more general meaning is still open. A discussion of the definition of "group" itself would definitely exceed the length of this paper, as there are many different definitions. In this paper, we will use the expression "group" in the sociological sense of "social group", which essentially means a group is defined as a collection of humans who share certain characteristics, interact with one another, accept expectations and obligations as members of the group, and share a common identity [17].

In contrast to a psychological definition, where clearly a face-to face contact is required [15], we concentrate on mainly "virtual" groups, which mean groups where face-to-face contact is not given or is given only via electronic channels. These "virtual" groups are essential for students in e-learning environments.

Out of the context of e-learning situations we define three group types:

- Study group
- Working group and
- Learning group.

A. Study Group

The study group, as the most complex of the three group concepts, consists of students of one or more than one university. The affiliation to a certain faculty, discipline or the participation in a certain teaching event is not important. Normally, study groups do not span more than one university, but it is possible and also desirable. Study groups are also a chance for interdisciplinary exchange. Organized interdisciplinary networks can be highly successful as is demonstrated by some open knowledge communities in the Internet. An example is http://www.wer-weiss-was.de, the largest German speaking knowledge community.

The study group is an informal group, meaning it is not formally initialized by an institution as is the case with the working group. The main issue of the study group is to create a social network.

The group members find each other spontaneously, e.g., in events for new students, via voluntary work, through the student council, in discussion groups concerning students' issues like finding jobs and so on.

The size of such a group varies between small (starting with three) to very large with open end.

Study groups often continue throughout the complete study time; sometimes they turn into social networks that live long after the studies are finished.

Since the American sociologist Granovetter published "The Strength of Weak Ties" [5] in 1973 the positive effects of social networks have been well known [9, 12, 14]. For example, access to relevant information concerning the private and professional life is highly facilitated for members of such a group. An interdisciplinary study group elegantly supports the following situations, and may smoothly lead to a network with obvious benefits for its members far beyond the original idea of a "learning network":

- A construction engineer urgently needs legal information. In case he knows a lawyer, he will gain the information a lot faster than in case he knows none.
- Job vacancies are first known to the members of the study group; professional support for your children at school is best given through a known teacher, and advice in medical questions can be given by doctors.

Our evaluations and a study of the HIS Company very clearly show that students are aware of the advantages of study groups, especially interdisciplinary ones, and - this is important - they expressly wish that the university supports them. The reason is that they want a homogenous social environment, closely linked to and compatible with their learning environment [3, 4].

However, this type of group has virtually no support by universities [3, 4, 7]. So far, most distance teaching universities do not understand that this is a central concept leading to a closer relationship with the university, a feeling of "being part", which then leads to a decreasing number of dropouts and finally a supporting community of alumni.

Currently, the members can only use the given electronic communication channels like chat, email, different groupware tools, forums, etc. Mostly, they use the university newsgroups. The problem is that newsgroups are very large, it is difficult to follow certain topics, the members are more or less anonymous, there is high fluctuation, and there is definitely no socially binding element to stabilize the community. Conference systems are only useful for small groups of users, so they are not adequate, either. Learning management systems are hardly applicable for the purpose of social networking, as they are focused on courses and learning events. Tools like wiki webs and weblogs would make sense for the use in study groups, but they have to be integrated into a homogenous environment, and be organized in a meaningful way for students.

Students have the possibility to use existing free student interest communities, one of which is the German community study-board [http://www.study-board.de] or the community NASPA [http://www.naspa.org/about/index.cfm] for US students. Both provide services for all students, irrespective of faculty, number of terms and grade. As these communities are very widespread and their purpose is to offer services for mainly on-campus students, the provided contents and contacts are mostly too general for distance teaching students. Some of the most important topics in a distance teaching environment are not addressed and hard to address in such a general environment, such as the question about the best strategies to organize work life, private life and studies. Furthermore, the use of these services is completely outside of the student's learning space.

By far the most efficient solution for distance teaching universities is to provide a community platform, tailored to the students' needs, and integrated with the e-learning space. Apart from the fact that this solution simplifies the life of distance learning students substantially, this is a central feature to create the long-term bond and the "I belong" feeling for students – an effect that a distance teaching university has a much harder time to achieve than traditional universities.

B. *Learning Group*

Learning groups are related to study groups, as they are also informal groups with varying members and no formal enrollment. The members are usually students of one university and the same field of study, but in principal students of other universities with similar studies could also take part in the learning group. In contrast to working groups, learning groups are not limited to just one learning event; usually they stay together throughout the duration of the study and some even longer [1, 2, 10]. The members find each other in related learning events, like introductory meetings or basic instruction lessons. The size of a learning group varies between two or three members up to a maximum of ten. Experience shows that the average number is about four members. The main intention of learning groups is not, or at least not in the first place, social networking as is the case with the study group, but collaborative learning for the preparation of exams and for motivating each other to keep up with the studies . Also, all forms of organizational advice related to the studies are given by the learning group (which courses should be taken, how to prepare best for an exam, where to find good information resources, etc.).

The reliability of the group members is more important than in the study group, as undependable members could disturb the whole group. The average lifespan of such a learning group is to the end of the studies of its members, even if some members stay in contact beyond that time. As these groups are informal groups, fluctuation of members is a known phenomenon (members move, terminate their studies, new members join in, etc.). Learning groups are highly suitable for large teaching events with a high contingent of self-study, such as introductory courses, beginners' seminaries etc. Tutors should encourage the building of learning groups.

Good suitable components for the technical support of these groups are groupware tools to communicate and to share files. Also wiki webs, weblogs and e-portfolios are interesting options for a learning group. An increasing number of students are also using audio conference tools like Groupspeak or Skype, which are very convenient for small groups. Learning management systems are partly suitable for the purpose of collaborative learning, if they provide groupware functionalities.

C. *Working Groups*

Working groups are the most common and widely-used form of groups in e-learning, and also the best investigated type of group [6, 12, 14, 15]. The main difference to the other two group types is that their purpose is very clearly defined by the issue of the learning event this group belongs to. Another difference is the guidance by the tutor, which is essential for the groups' success [1, 2, 6, 15]. This type of group is clearly an institutionalized formal group. Working groups are ideally appropriate for small teaching events, such as seminars, colloquies or practical trainings; they are closely related to project groups in companies.

The building of working groups depends on the organization of the learning event itself. Some tutors prefer to arrange the group themselves, according to their specific didactic preferences. Other tutors leave the building of the group to the students themselves. For both methods, the moderation of the group is very important, as the failing of one member could cause a drawback for the whole group, which in turn may lead to the failure of the whole group. Experience shows that a restriction to the number of up to five students is useful. Otherwise, the group is difficult to organize and the risk of disappearing members is high [1, 6, 10]. Normally, a working group lives only as long as the learning event itself. In some cases working groups alter to long lasting learning groups. For this type of group various tools can be used:

- Collaborative Work Software
- Conferencing tools (audio and/or video)
- Learning Management System
- Wiki webs and social networking tools

Which tool to choose depends on the didactic issues of the tutor, the number of students, the availability and accessibility and also the personal likings of the tutor and the students. A typical example for university working groups is the virtual seminar:

A total number of fifteen up to twenty students divided into small groups of a minimum of two and a maximum of four students together create contributions about aspects of a specific topic, e.g., knowledge management. The contributions cover up to twenty pages of text per person. To prepare these contributions, students use a groupware system. In intervals each group presents its results via audio or video conference, followed by a discussion with their fellow students and the teacher. The whole seminar runs throughout one term (about three months).

D. *The current situation of group types in Hagen*

The University of Hagen is the typical distance teaching institution with fully employed students, students with small children, students with handicaps. Participation in ongoing groups and group work is only possible for most of them if that is possible from a distance, i.e., if these groups are Internet-based.

Nevertheless, it is still difficult for the students to start groups and to participate in existing ones: there remains the challenge to find each other and then to keep the group running.

A few students started study groups on external platforms, but – for the reasons given earlier (no integration into the e-learning space, diversity, not visible in the university information space etc.) – these are not very well known. However, the fact that these groups exist in spite of the described difficulties proves how useful this kind of group is considered by students.

One of the central problems for students to start study groups on external platforms is to gain access to the necessary data of their fellow students. The only possibility so far is the general newsgroup for distance students with more than 80 threads a day. A start has been made with electronic contact lists, provided by the students' council (called ASTA), but these lists often don't contain the necessary data like faculty, degree, interests, etc.

A community in the sense of social networking is not available, but it is absolutely necessary. A technical reason is the wide variety of tools which students have to use if they have to start study groups on "foreign" platforms. The essential point, however, is that these functions need to be kernel features of the university e-learning and information space, well integrated and easy to use. It is important for students to be part of a group from the very beginning [3, 4]. So, instead of just adding some social software tools to the university infrastructure, a concept is required which integrates social software as a fundamental feature and thus creates the added value mentioned earlier for students and the university alike.

V. Conclusion

Three types of groups have been identified in this paper. These groups have different characteristics, and technical and organizational requirements. The support of these groups leads to essential improvements for distance teaching organizations. Some of the benefits are:

- The social binding of the students to their university will be strengthened; students will identify themselves with their organization.
- Students with a strong bond to their university embedded in a well-working community are supposed to be more successful as students without this support [4].
- A working infrastructure for group support helps to decrease the costs for small learning and teaching events as students are able to organize themselves and to support each other before they apply to the responsible teacher [3, 4, 13].

ISBN: 978-1-61208-105-2

- Well-organized group support possibilities help to decrease the organizational support load (e.g., less questions of the type “What course should I take?," if adequate study groups are available with collegues to discuss this issue). Decreasing work load also means decreasing costs and additionally a satisfied customer.
- Students are able to organize their time budget effectively by using spare time and unplanned free hours with easy access to a well organized communication and collaboration space.

Note that the support of groups requires not only the appropriate technical infrastructure, but first of all an integrated social and organizational concept. A possible solution is the implementation of a personal learning and community environment (PLCE) as suggested by students themselves [3, 4]. This PLCE should fulfill the following requirements:

1. Easy access, intuitive to use
2. Professionell information management
3. Awareness-function
4. Integrated communication and interaction possibilities like:
   a. Interdisciplinary communication and interaction, e.g., by integrating popular social networking sites like Facebook
   b. Infrastructural support to set up and to support different group types
   c. An Alumni network
   d. Private rooms without access for teaching staff.
5. High security measures
6. Integrated linkage to existing Web 2.0 tools (Messaging tools, Social Networking Sites, blogs, social bookmarks etc) for instance via the Open Social API
7. Personalized information learning and knowledge management
8. Intelligent search engine

- The students’ needs are clearly identified [3, 4, 10, 13] by now and the task of the university is to improve the current situation according to the given suggestions. It is of utmost importance to restructure the current learning environment with a strong focus on the support of communication and interaction processes by installing community oriented features as described above. Not content and organizational functionalities are central, but finding adequate communication partners and being part of a group as early as possible and as long as possible. Becoming part of a group is useful even before enrolment. Students, teachers and staff should form a virtual community for learning and teaching, supported through adequate technology. This platform must provide easy to use functionality for
  - organizing, discussing and publishing content collaboratively
  - discussing and solving specific problems together
  - creating different types of groups.

To achieve this goal, it is necessary to develop a new learning portal according to the students’ needs. The detailed description of this new environment (architecture, features, interface, necessary restructuring) of this University as Community is part of the doctoral thesis of the author.

REFERENCES

[1] Feldmann-Pempe, B. and Schlageter, G. (1999) Internet-based Seminars at the Virtual University: A Breakthrough in Open and Distance Education. Proceedings of the ED-Media 99, Seattle, AACE, pp. 887-892

[2] Feldmann, B. and Schlageter, G. (2001). Five Years Virtual University – Review and Preview. Proceedings of the WebNet01, Orlando: AACE, pp. 355-361.

[3] Feldmann, B. (2010). Fernstudium n.0 – Fernuniversität als Gemeinschaft [Distance Study n.0 – Distance Study as Community]. To be published: Proceedings of GeNeMe 2010, Dresden.

[4] Kleimann, B.; Özkilic, M. and Göcks, M. (2008] Studieren im Web 2.0 [Studying in the Web 2.0], HIS Projektbericht. https://hisbus.his.de/hisbus/docs/hisbus21.pdf.

[5] Granovetter, M. (1976). The Strength of Weak Ties. American Journal of Sociology, 78 (May), pp. 1360-1380.

[6] Haake, J., Schwabe, G. and Wessner, M. (2004). CSCL-Kompendium. Oldenburg.

[7] Kerres, M. and Nübel, I. (2005). The Status of E-learning at German Higher Education Institutions. In: Dittler, U,, Kahler, H., Kindt, M. and Schwarz, C. (Eds.), E-learning in Europe – Learning Europe. How have new media contributed to the development of higher education? (Vol. 36, pp. 29-50). Waxmann..

[8] Leh, A. S. C. (2001). Computer-Mediated Communication and Social Presence in a Distance Learning Environment. International Journal of Educational Telecommunication 7 (2), pp 109-128.

[9] Mason, R. (1994). Using Communications Media in Open and Flexible Learning. Kogan Page.

[10] Mittrach, S. (1999). Lehren und Lernen in der Virtuellen Universität: Konzepte, Erfahrungen, Evaluation. [Teaching and Learning in the Virtual University: Concepts, Experiences, Evaluation]. Shaker Verlag.

[11] Ogata, H. and Yano, Y. (1998): Supporting awareness for augmenting participation in collaborative learning. Proceedings of the WebNet98, Charlottesville: AACE.

[12] Palloff, R. M. and Pratt K. (1999) Building Learning Communities in Cyberspace. Effective Strategies for the Online Classroom. Jossey-Brass, San Francisco.

[13] Peters, O. (2004). Distance Education in Transsition. New Trends and Challenges. Arbeitsstelle Fernstudienforschung. Oldenburg. http://www.c3l.uni-oldenburg.de/publikationen/vol5.pdf. Last access: 08/18/2010.

[14] Preece, J. (2000). Online Communities. Designing Usability, Supporting Sociability.

[15] Rechtien, W. (1999). Angewandte Gruppendynamik. [Practical Group Dynamics] Beltz.

[16] Schulmeister, R (2005). Lernplattformen für das virtuelle lernen: Evaluation und Didaktik [Learning Management Systems for Virtual Learning: Evaluation and Didactic.].

[17] Tajfel, H. and Turner, J.C. (1986). The Social Identity Theory of Intergroup Behaviour. In: Worschel, S. and Austin, W.G. (Eds.). Psychology of Intergroup Relations (pp. 7-24). Nelson-Hall.

[18] December, J. (1996) Units of Analysis for Internet Communication. Journal of Communication 46 (1) Winter, pp. 0021-9916.

[19] The Virtual University of Hagen: http://vu.fernuni-hagen.de, Last access: 08/18/2010.

Archer: An Architectural Monitoring Tool

Vitor C. Alves, Rafael H.S. Rocha, Rodrigo de B. Paes, Evandro de B. Costa, Leandro Dias da Silva
Universidade Federal de Alagoas
Maceió, Brazil
{vitorcorreia.ufal, rafaelrocha.ufal, leandrodds, ebcosta}@gmail.com, rodrigo@ic.ufal.br

Gustavo R. De Carvalho
Pontífica Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brazil
guga@les.inf.puc-rio.br

Abstract - Software Maintenance is a continuous process in software development that begins when the software is first released and does not end while the software is being used. This characteristic makes it one of the most expensive processes in software development. Software engineering has identified some factors that increase software maintenance costs and presented good practices to face these problems. Good software architectures make a software easier to maintain and to evolve. Several reference architectures have been defined. Nowadays, there are software tools that provide architectural discovery and documentation tools, but they do not effectively protect the architecture from being compromised. This paper presents a software architecture monitoring tool called Archer, which was implemented as an eclipse plug-in. This tool aids the programmers with respect to software architecture through identifying architectural flaws introduced when coding. Also, Archer supports discovering existing architecture from a software project by using reverse engineering techniques, providing the architect with information to improve, or do not compromise, the software architecture in existing software.

Keywords - Software Engineering; Software Architecture; Architectural Enforcement, Maintenance.

I. INTRODUCTION

Software maintenance is an activity that begins when the software is released and users start to use it. It corresponds by up to 80% of total software costs [1]. Software documentation is an important practice to maintain a software. It aids programmers in the understanding of how the software was designed and how changes can be made without compromise its structure. However, only the software documentation is not enough to guarantee protection to its logical structure, sometimes programmers do not obey, either deliberately or unintentionally, the software architecture and break it. This problem normally appears when the programming team changes, and no further explanation about the software structure and architecture is passed to the new employees.

The problem stated above suggests that it would be desirable to have a solution that helps the programmers in the understanding of legacy software. The solution should also enforce that architectural decisions will not be broken by the programmers, at least unintentionally. The software tool presented in this paper aims to fulfill both requirements: (i) to help in the understanding of already developed applications and (ii) to specify and enforce architectural styles [16]. The remaining of this paper is organized as follows. Section 2 describes some related work and compares them to Archer. Section 3 shows in details how Archer works. Section 4 illustrates the Archer though a case study. Finally, in Section 5, we conclude the paper discussing the contributions, limitations and further improvements of the current proposal.

II. RELATED WORK

ARCHJAVA [2] is a tool to recover software architecture on legated systems written in Java. Their goal is to be able to recover architectures documented in the literature, such as MVC (Model-View-Controller) [3] and Layers [3] by defining architectures as domain-independent rules. These rules are based on static [17] and dynamic [17,18] analysis. Static analysis enables the verification of software structure and dynamic analysis verifies the objects behavior. However, in contrast to Archer, ARCHJAVA is intended to be used only in java based software.

A hybrid computer aided approach for close monitoring source code by using this same static and dynamic analysis methods is presented in [15]. On this approach, the verification process analyses design-implementation congruence: concrete rules such as coding guidelines, architectural components, such as design patterns [10] or connectors [14], and design principles such as low coupling and high cohesion.

In Harris et al. [19], a language to request parsed information to analysis is described: the source code query language. This language allows programmers to recover information from an abstract syntax tree. The idea is very similar to the Archer architectural analyzer (Section 3.3), but since this query language interacts directly with the source code, the entire program will have to be rewritten to support a new source-code language. Archer, on the other hand, is prepared to support new languages without this kind of effort. This is possible because its architectural analyzer interacts only with the object oriented model (Section 3.2) which is language independent.

Another tool similar to Archer on its objectives is Dali [4], a workbench that aids the analyst to manipulate and interpret recovered architectural information. There are works such as LSME [5] and RMTool [6], but their scope is very similar to Dali. The main difference between these works and Archer lies on the fact that they only work with legated systems, in other words, their concern is about recover software architecture to support the user in defining software architecture, but no further action is taken. Archer aims to use recovered information to protect existing architecture through the enforcement of architectural rules.

III. ARCHER

Archer is a plug-in that works integrated with Eclipse IDE [7]. It provides support for the software architecture enforcement and documentation. It is able to recognize architectural patterns in code and verify if a software is in agreement with a pre-established architecture. Its structure is divided in three parts: a parser, an object-oriented model and the architectural analyzer. Figure 1 illustrate the process of analysis.

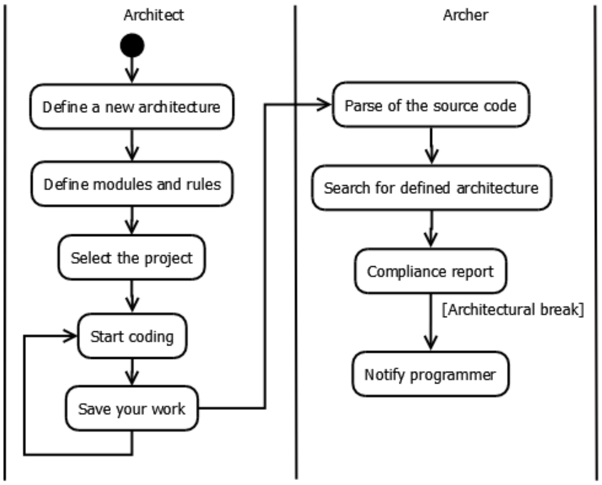


Figure 1. Archer process activity diagram

3.1 Parser

Through source-code analysis, all relevant information about the software is gathered by the parser. Examples of such pieces of information are classes, packages and relations between classes. This information is then organized on the Archer's object-oriented model. Although archer is designed to support the analysis of source code written in different programming languages, the current implementation works only with java. The parsing of the source code is made using the Eclipse/JDT Java Model [8] (Java Development Tools). Archer was designed for extension, then there are hotspots that may be extended for supporting parsers of other programming languages. Since all the information is stored into the archer's object-oriented model, archer can be used to analyze source code written in other programming languages, with no need of changes in the architectural analyzer. In this case it is necessary to change the parser. In other words, a new architecture analyzer (section 3.3) is not required to verify existing architecture patterns, just a new parser for other languages.

3.2 Object-Oriented Model

From the Archer point of view, the lowest abstraction level of a software's structure is its implementation (source code). A model represents this structure in a language-independent manner. It is composed of a set of elements which are present in object-oriented languages. Figure 2 shows the Archer meta-model. It is based on UML Meta-Model [9]. The main goal of this model is to represent the code structure. The model represents this throw in a set of objects which can be manipulated without language-dependent issues. The elements are defined in two main groups: relationships and named elements. Relationships represent connections between concrete elements. There are two relationships represented in the model: interface realization and inheritance ("Generalization").

The named elements are elements that have an identifier. They are defined in four main types: "Packageable Element", "Namespace", "Redefinable Element", and "Typed Element". The PackageableElement contains elements with a visibility type, e.g a class can be private, private is the visibility of the class. Namespace can contain other elements with names and can exclude equivalent elements within it. RedefinableElements are elements that can receive different values from other RedefinableElements that are equivalent or more specialized. TypedElement are elements that contains a "Type", i.e a primitive type or a class type. The "Project" is a "Package", since it can contain other packages as well as packageableElements. "Class" and "Interface" are "Classifiers". However, only "Class" is a type due to the reason that interfaces cannot be instantiated.

The Variable element is defined as a typed element only. Therefore, it cannot be analyzed in a redefinition context as a Property. However, in most cases it was not necessary to evaluate a constraint of an architecture. The Variable can be treated as the operations and properties since they are both "Typed Elements".

Figure 3 shows how a class could be graphically represented in the model in a simplified approach. It contains a generalization relationship, generalization contains oval and drawable component. Generalization is a directed relationship. Therefore, it has a source component and a target component.

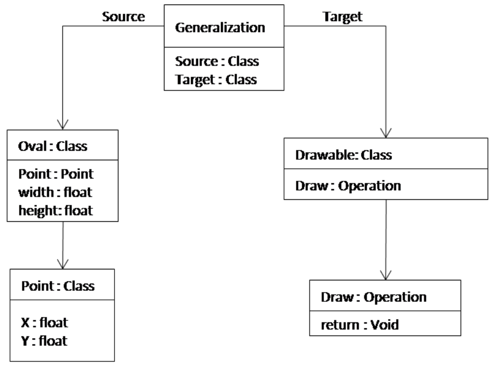


Figure 3. A Simple Class Representation in the Model

In the illustrated case, the oval class is the source component and drawable, the target. This means that oval inherits drawable component. Drawable has a draw operation, the oval component has a point and two float attributes and the point has two float attributes.

Usually, a software architecture is separated in modules. These modules can often be separated in smaller units, such as design patterns. These units are abstractions that commonly represent implementation constraints. The architectural analyzer contains modules composed of rules for representing these smaller units. The rules search for code samples implementing the constraints, e.g., Singletons [10], have private constructors. Therefore, a rule should search in classes for constructor methods with private visibility.

The tool use the rule constraints implemented in terms of the model components. The Analyzer retrieves classes from the code analyzed and tests them for each rule. Since the code was parsed to be represented in components of the model, comparison is possible. A matching percentage is given as evaluation result for a module. Each rule counts a point and the sum is divided by the total of rules from this module.

3.3 Architectural Analyzer

Having all the information required, the analysis process may start. Archer works with two important concepts for its operation: code rules and architectural patterns.

3.3.1 Code Rules

Code rules are built from the information contained in the object oriented model database. These rules are patterns found in source code. An example of a code rule is the verification of all classes that implements an observer. Code rules are used to search for architectural patterns. On Archer, these rules are implemented as a class using the Java language.

In the current version, Archer contains the patterns: Command, Singleton, Observer, Abstract Factory, Bridge. Frequently, these patterns are key elements for defining an architecture. Archer has also more practical rules, such as the detection of graphic objects, database access and event controllers (listeners). The problem of verifying the existence of patterns in source code was already studied in [21].

3.3.2 Architectural Patterns

Architectural patterns are known solutions that work efficiently to solve an architectural problem. It defines how classes interacts and sets how the software information flows in runtime [11]. In practice, it also defines some characteristics that classes should have to be part of an architectural structure such as a layer, for example. On Archer, we define an architectural pattern as a group of these characteristics code rules.

Archer analysis process starts everytime programmers save their work, by searching every class of the project for code rules. When the process is finished a list of found code rules is obtained, and then a process of comparison is made to check if the classes have any similarity with one of the architectural patterns contained in the database. This process will give a percentage, indicating the chance of a class to belong to a specific architectural structure from an architectural pattern. The higher the percentage is, better are the chances of the class to belong to that structure.

An important feature of Archer is that the architect can make his own architecture from existing code rules. This is possible because of its generic analyzer. When the information is loaded from the object-oriented model, it organizes the information on a matrix. The information contained in that matrix informs which classes contain which code rules. Then a process of finding similar classes on the matrix is performed in order to find possible members of the same architectural modules. Finally, an analysis is made to identify the architecture itself, based on the characteristics found on these architectural modules.

After the analysis process is done, the results are saved and programmers will do their work as usual, but anytime the programmers team would otherwise compromise the architecture, Archer will send an alert warning about unwanted changes.

The process of enforcement is completed after the analysis process. If a programmer tries to break the architecture, Archer will analyze its database and check if the change is harmful to the architecture. If it is, Archer will discourage the programmer to continue with his changes. That way, the architecture will be safe. For example, if a programmer tries to put graphic objects on the model abstraction of the MVC architectural pattern, Archer will warn about unwanted code rules on this abstraction (model does not implement graphical objects). This way, an act that would otherwise compromise the architecture will be prevented.

IV. CASE STUDY

To illustrate Archer’s functionality, a simple calculator and an artificial intelligence simulation software made as a chess game [20] were analyzed under the MVC Architecture [3]. In this section, we present how Archer was used to match existent software architecture to a previously defined architecture.

The calculator is separated in three main packages, the Model is represented by the Calculator class which contains the manipulated data and it is observed by the view’s component. The View is represented by the Window class which contains graphical components and observers the model. The Command component is represented by the listener classes, the ActionListener acts as an Observer interface. Figure 4 shows a simplified version of the UML representation of the calculator.

Archer was setup to use the MVC Architecture. It means that it will verify whether the source code of the application is in conformance with the MVC. The architecture was defined in three modules (model, view, control). Each module implemented it own rules (see Section 3.3). The code parser retrieves the code from the selected project and parses it into objects of the object oriented model (see Section 3.2).

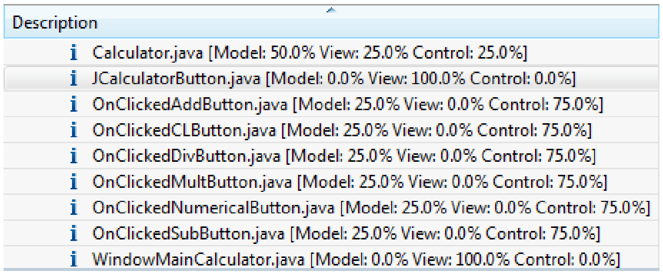
The process of analysis is a sequential search for correspondence with the rules defined in the architecture. It is made with a binary vector that stores the rules needed to define each component of the architecture design. Figure 5 illustrates part of the evaluation process.

The Analyzer verify if a component obey(x) or disobey (-) a rule. In the illustrated case AddListener follows the same rules that define the control component. After the analysis is made, the tool compares the results of each class with the modules of an architecture attributing a percentage of correlation.

Each attribute of a class is already in the model at evaluation time. Therefore, if the tool needs to analyze an attribute of a class before analyze the class itself, it is possible. This feature allows the tool to evaluate components that are defined in terms of others, e.g the classes of view component contain classes from control component.

An architecture is defined in the module "Architectural Composer" each composer contain "Architectural modules" and each module a set of "Rules". A rule is defined as a boolean function which gets a "Class" from the model and evaluate it. The function is defined as a implementation constraints. The complete project structure is available for a rule function. However, it evaluates a class per time. If an attribute of a class must be evaluated before it reaches a conclusive response, the rule pass the attribute to the analyzer to it be evaluated first.

The results of the analysis show the percentage of compatibility of each class with the modules of the architecture as illustrated in Figure 6. Although the results were acceptable, they could be improved.


Description

Calculator.java [Model: 50.0% View: 25.0% Control: 25.0%]

JCalculatorButton.java [Model: 0.0% View: 100.0% Control: 0.0%]

OnClickedAddButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

OnClickedCLButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

OnClickedDivButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

OnClickedMultButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

OnClickedNumericalButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

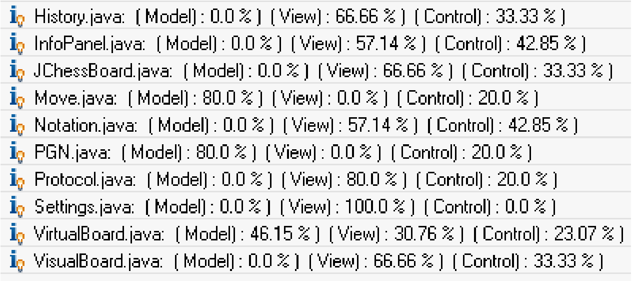
OnClickedSubButton.java [Model: 25.0% View: 0.0% Control: 75.0%]

WindowMainCalculator.java [Model: 0.0% View: 100.0% Control: 0.0%]

Figure 6. Calculator Evaluation

For the JChess, Archer concluded that the software did not followed the MVC architecture, Figure 7 shows the UML representation of the application.

Archer was used to analyze the architecture of this application according to the MVC Architecture, Figure 8 show the results. Some classes implemented the Model of MVC, e.g. Move and PGN which reaches high compatibility with this module, these classes contains data information that is accessed by the view module, JChessBoard contains graphical interfaces and modifies the data of model classes directly so it is unevenly distributed over view and control. Most of the classes are not well defined, it shows the inconsistency of the application with this architectural pattern.


History.java: (Model) : 0.0 %) (View) : 66.66 %) (Control) : 33.33 %)

InfoPanel.java: (Model) : 0.0 %) (View) : 57.14 %) (Control) : 42.85 %)

JChessBoard.java: (Model) : 0.0 %) (View) : 66.66 %) (Control) : 33.33 %)

Move.java: (Model) : 80.0 %) (View) : 0.0 %) (Control) : 20.0 %)

Notation.java: (Model) : 0.0 %) (View) : 57.14 %) (Control) : 42.85 %)

PGN.java: (Model) : 80.0 %) (View) : 0.0 %) (Control) : 20.0 %)

Protocol.java: (Model) : 0.0 %) (View) : 80.0 %) (Control) : 20.0 %)

Settings.java: (Model) : 0.0 %) (View) : 100.0 %) (Control) : 0.0 %)

VirtualBoard.java: (Model) : 46.15 %) (View) : 30.76 %) (Control) : 23.07 %)

VisualBoard.java: (Model) : 0.0 %) (View) : 66.66 %) (Control) : 33.33 %)

Figure 8. JChess Evaluation

V. CONCLUSION

It was developed a parser that reads java codes and generates a language-independent OO model was developed. It allows to represent the collected data without loss of information and it is free of language details. The permits analyze the code of a project by comparing a set of classes to a design template.

Archer can define rules at a low level of abstraction. In the current version, these rules are created by programming in java language using model's classes provided by Archer API. We are aware that there is a need to define the architecture in a higher abstraction level. We have already tried to represent these architectural rules in ADLs (Architectural Description Language) such as ACME [12] and Wright [13], however, they do not allow us to express the level of details it is needed to perform the enforcement of the architecture.

Archer has some features integrated to eclipse. It can notify the developers if some architecture has a potential problem. At the moment they save their code, the evaluation appears in the Eclipse problems panel. As future work, the tool will be integrated with subversion version control system. This feature would allow verifying whether an architectural rule is broken at the moment developers commit their code. It is being studied a way to represent the architecture through an ADL. The idea is to follow a bottom up approach, i.e., in the current version, the architecture must be defined programmatically using the Archer API. Each architecture defined is included in the architecture database. Once the database becomes bigger, it will be possible to reuse code rules of previously defined architectures. Then, as the level of reuse increases, it will be possible to create a language to represent these code rules.

ACKNOWLEDGMENT

This work is partially supported by CNPq/Brazil under the project PROCOCO (620063/2008-4)

REFERENCES

[1] Ambler,S.W.(1997), Análise e projeto orientados a objetos, Infobook, Rio de Janeiro.

[2] Carvalho F., Barroso L, Seufitele V., Vasconcelos A. ArchJava: Reconhecimento de padrões arquiteturais em sistemas Java, CEFET CAMPOS.

[3] Buschmann, Frank et al. Pattern-Oriented Software Architecture: a System of Patterns. Wiley, 1996.

[4] Kazman, R. and Carrière, S. J. 1999. Playing "Detective: Reconstructing Software Architecture from Available Evidence". Automated Software Eng. 6, 2 (Apr. 1999), pp. 107-138.

[5] Murphy G., Notkin D. Lightweight Lexical Source Model Extraction. In ACM Transactions on Software Engineering and Methodology, Vol. 5, No. 3, July 1996, pp. 262-292.

[6] Murphy, G., Notkin, D., and Sullivan, K. "Software Reflexion Models: Bridging the Gap between Source and High-Level Models". in Proceedings of the Third ACM SIGSOFT Symposium on the Foundations of Software Engineering, (Washington, D.C.), October 1995. pp. 18-28.

[7] Eclipse, http://www.eclipse.org/ 04.14.2010

[8] JDT, "Java Development Tools", http://eclipse.org/jdt/ 04.14.2010

[9] Unified Modeling Language, http://www.uml.org/ 01.29.2010

[10] Gamma, E., Helm, H., Johnson R., Vlissides, M. J. "DesignPatterns: Elements of Reusable Object-Oriented Software."

[11] Sommerville, I. "Engenharia de Software" Addison Wesley 8ª ed.

[12] Garlan, D., Monroe, R. T., and Wile, D. 2000. "Acme: architectural description of component-based systems". In Foundations of Component-Based Systems, G. T. Leavens and M. Sitaraman, Eds. Cambridge University Press, New York, NY, pp. 47-67.

[13] Allen, Robert J. "A Formal Approach to Software Architecture" (Ph.D. Thesis, CMU-CS-97-144 ed.). Carnegie Mellon University. (May 1997).

[14] Shaw M., DeLine R., and Zelensnik G. "Abstractions and Implementations for Architectural Connections". Technical Report CMU-CS. pp. 95-136, CMU, March 1995.

[15] Sefika, M., Sane, A., and Campbell, R. H. 1996. "Monitoring compliance of a software system with its high-level design models". In Proceedings of the 18th international Conference on Software Engineering (Berlin, Germany, March 25 - 29, 1996). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, pp. 387-396.

[16] Abowd, G., Allen, R., and Garlan, D. 1993. "Using style to understand descriptions of software architecture". In Proceedings of the 1st ACM SIGSOFT Symposium on Foundations of Software Engineering (Los Angeles, California, United States, December 08 - 10, 1993). D. Notkin, Ed. SIGSOFT '93. ACM, New York, NY, pp. 9-20.

[17] Olshefski D. P. and Code A. "A Prototype System For Static and Dynamic Program Understanding". In Proceedings of the Working Conference in Reverse Engineering, Baltimore, MD, USA, 1993. pp. 93 - 106.

[18] Ritsch H. and Sneed H. M. "Reverse Engineering Via Dynamic Program Analysis". In Proceedings of the Working Conference in Reverse Engineering, Los Alamitos, CA, USA, 1993.

[19] Harris, D. R., Reubenstein, H. B., and Yeh, A. S. 1995. "Reverse engineering to the architectural level". In Proceedings of the 17th international Conference on Software Engineering (Seattle, Washington, United States, April 24 - 28, 1995). ICSE '95. ACM, New York, NY, pp. 186-195.

[20] JChess, http://jchessboard.sourceforge.net/ 20/08/2010.

[21] Blewitt, A., Bundy, A., and Stark, I, "Automatic Verification of Design Patterns in Java". In ASE 2005: Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, Long Beach, California, USA, November 7–11, 2005, pages 224–232. ACM Press, 2005.

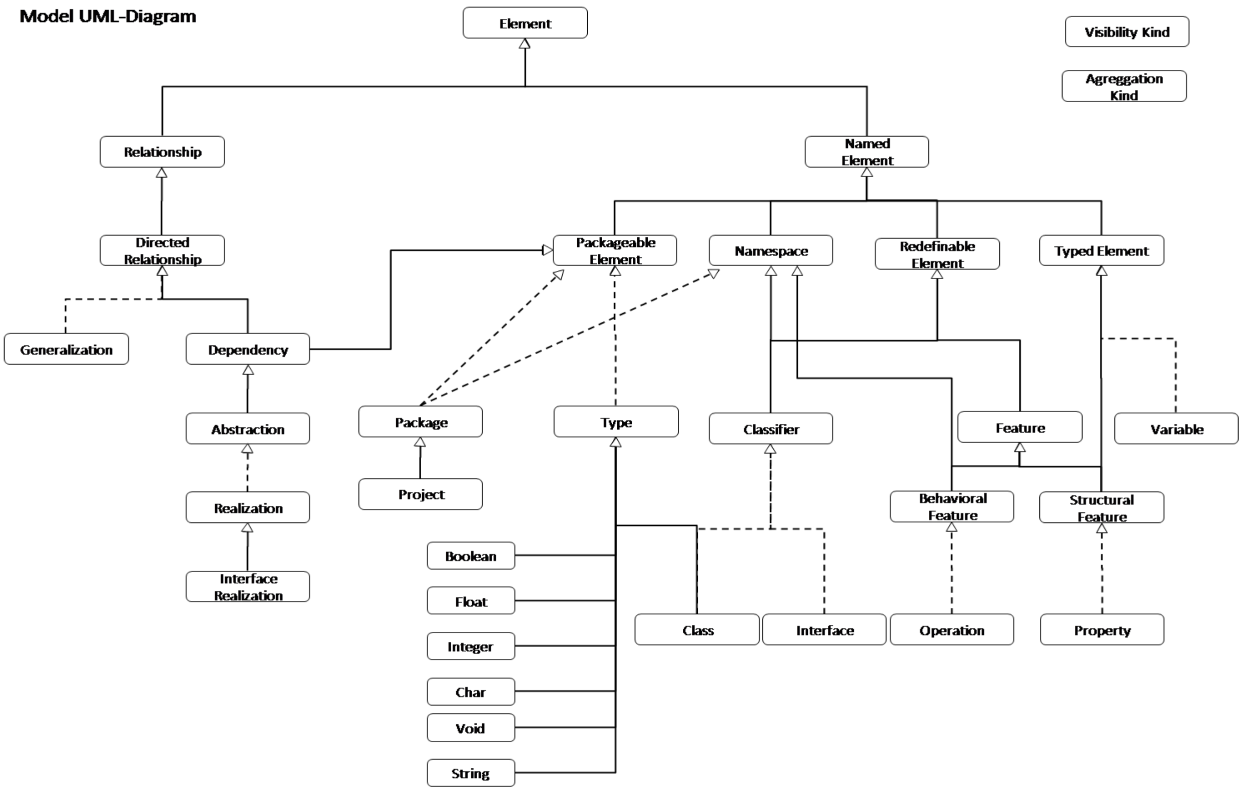


Figure 2. Simplified Representation of Archer Meta-Model

ISBN: 978-1-61208-105-2

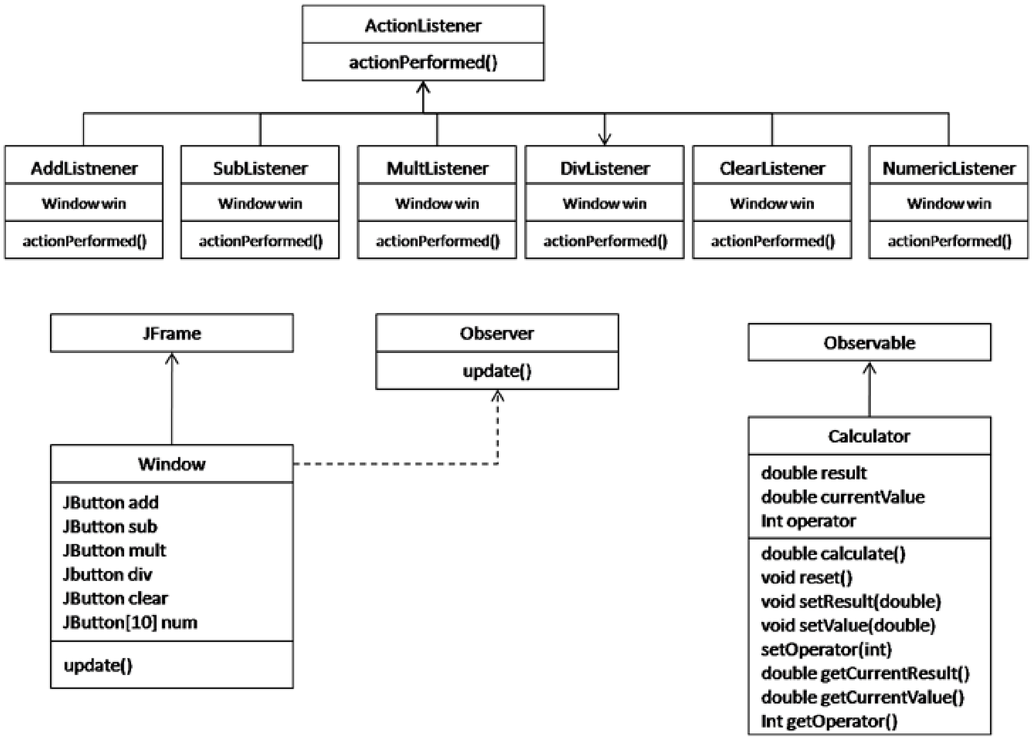
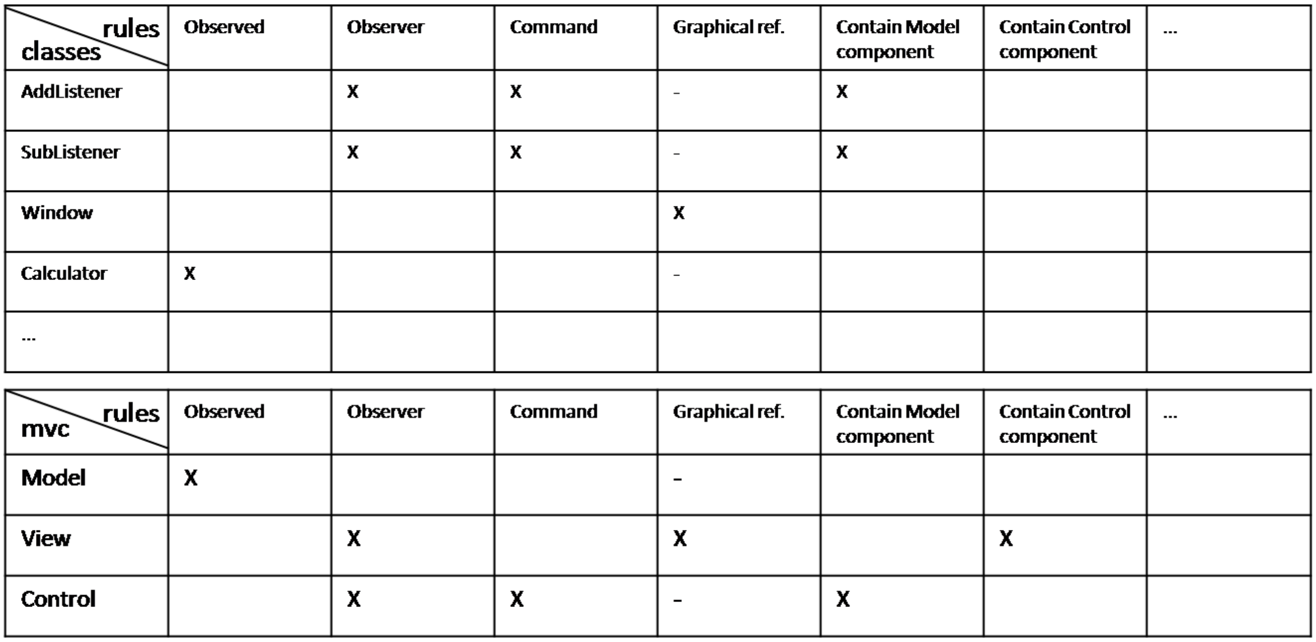


Figure 4. MVC Calculator Sample UML Description



rules classes	Observed	Observer	Command	Graphical ref.	Contain Model component	Contain Control component	...
AddListener		X	X	-	X		
SubListener		X	X	-	X		
Window				X			
Calculator	X			-			
...							

rules mvc	Observed	Observer	Command	Graphical ref.	Contain Model component	Contain Control component	...
Model	X			-			
View		X		X		X	
Control		X	X	-	X		

Figure 5. The evaluation process

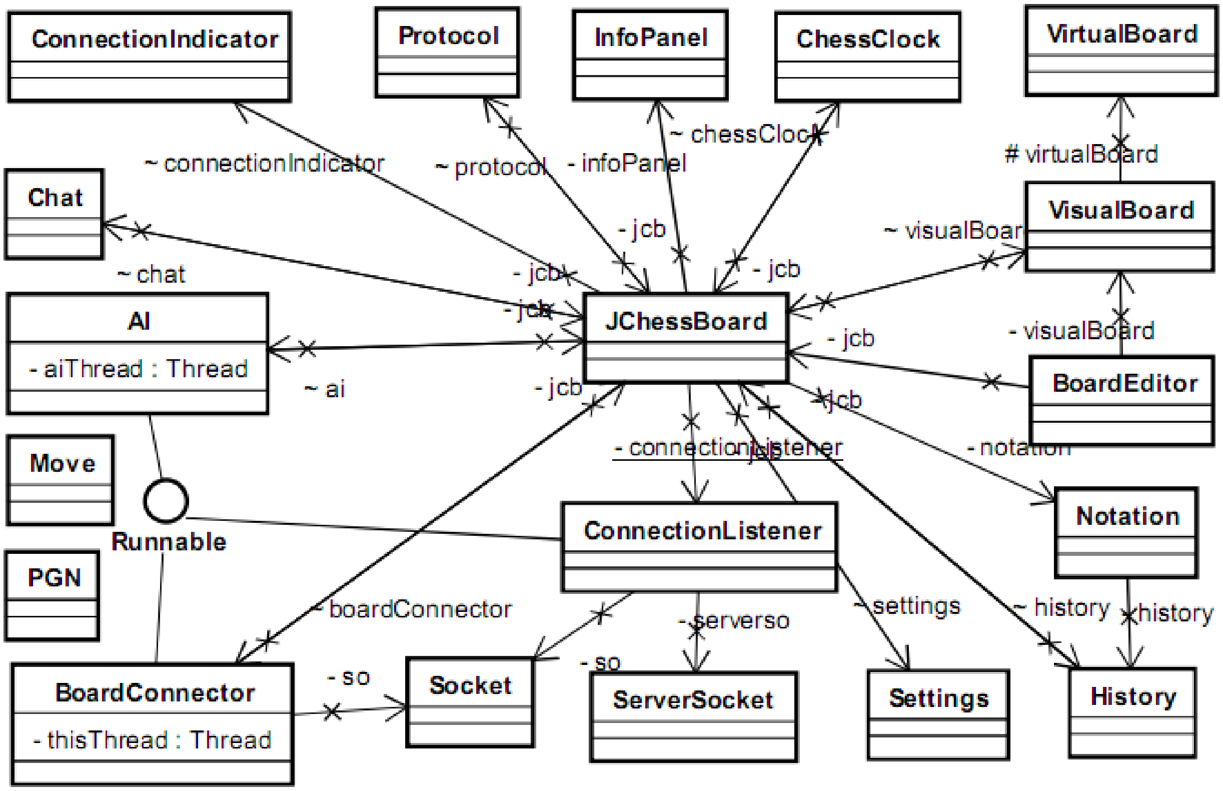
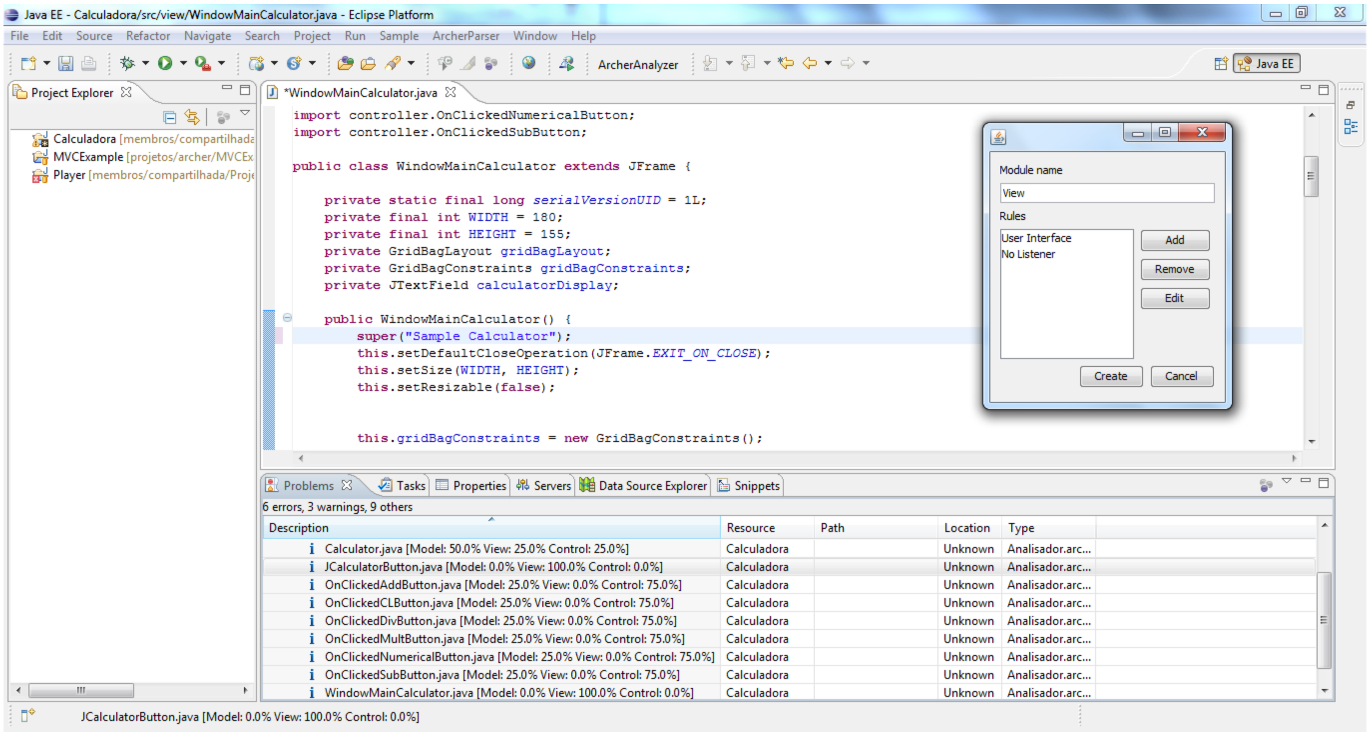


Figure 7. JChess UML Description


Java EE - Calculadora/src/view/WindowMainCalculator.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Sample ArcherParser Window Help
ArcherAnalyzer
Java EE
Project Explorer
Calculadora [membros/compartilhada
MVCExample [projetos/archer/MVCEx
Player [membros/compartilhada/Proje
*WindowMainCalculator.java

```
import controller.OnClickedNumericalButton;
import controller.OnClickedSubButton;

public class WindowMainCalculator extends JFrame {

    private static final long serialVersionUID = 1L;
    private final int WIDTH = 180;
    private final int HEIGHT = 155;
    private GridBagLayout gridBagLayout;
    private GridBagConstraints gridBagConstraints;
    private JTextField calculatorDisplay;

    public WindowMainCalculator() {
        super("Sample Calculator");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(WIDTH, HEIGHT);
        this.setResizable(false);


        this.gridBagConstraints = new GridBagConstraints();
```

Module name: View

Rules:
User Interface
No Listener

6 errors, 3 warnings, 9 others

Description	Resource	Path	Location	Type
i Calculator.java [Model: 50.0% View: 25.0% Control: 25.0%]	Calculadora		Unknown	Analisador.arc...
i JCalculatorButton.java [Model: 0.0% View: 100.0% Control: 0.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedAddButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedCLButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedDivButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedMultButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedNumericalButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i OnClickedSubButton.java [Model: 25.0% View: 0.0% Control: 75.0%]	Calculadora		Unknown	Analisador.arc...
i WindowMainCalculator.java [Model: 0.0% View: 100.0% Control: 0.0%]	Calculadora		Unknown	Analisador.arc...

JCalculatorButton.java [Model: 0.0% View: 100.0% Control: 0.0%]

Figure 9. The Archer plug-in

A Hybrid Instance Migration Approach for Composite Service Evolution

Jianing Zou, Hailong Sun, Xudong Liu, Kun Fang, Jingjing Lin
School of Computer Science and Engineering, Beihang University, Beijing, China
{zoujn, sunhl, liuxd, fangkun, linjj}@act.buaa.edu.cn

***Abstract*—Composite service evolution is one of the most important challenges faced in the field of service composition. And how to migrate running instances to the evolved definition is a critical issue for correct service evolution. In this paper, we proposed a hybrid instance migration approach in the aim of both increasing migration efficiency and flexibility. Based on Single Entry Single Exit fragments and process structure tre, we reduce the change region calculation algorithm's time complexity from exponential time to linear time. Moreover, our instance migration approach also includes data dependence analysis to avoid data flow problems during migration. This makes our approach more practical since data flow correctness preservation is critical in real world application. Finally, prototype system is given and experiments are carried out to prove the feasibility of our approach.**

Keywords- Instance migration; service composition; composite service evolution

I. INTRODUCTION

Service composition is widely considered as an efficient approach to building complex applications through composing loosely coupled component services [1]. Due to the highly dynamic network environment and ever-changing user requirements, it is imperative to maintain the flexibility of composite services. Thus, composite service evolution, which provides an appropriate solution to enhancing flexibility, has attracted many researchers' attention in recent years. Composite service evolution includes component service changeability and structural adaptation of process models [2]. In this paper, we focused on structural adaptation issue in composite service evolution.

How to migrate running instance during composite service evolution is an important challenge e.g., under the scenario that imperative government policy or business law changes happen. Besides, when the evolved composite service has a long running lifecycle, it is not acceptable to use other techniques such as version management to deal with running instances' evolution problem. For example, for a mortgage composite service whose execution cycle lasts several decades, it is unreasonable to maintain each instance's execution on its original version when the composite service changes each month. This will result in too many versions existing in the system, and make management quite complicated. However, live instance migration can be adopted to lower the complexity of runtime management as well as enhance the composite service flexibility in coping with changes. Therefore, instance migration has attracted more and more researchers' attention in service computing.

Instance migration problem includes control flow and data flow correctness preservation. The former one is the main focus of most published research. As what is mentioned in [9], control flow correctness mainly aims for maintaining the soundness during migration i.e., migrated instance won't result in execution deadlock or improper termination. Existing approaches solving this problem can be classified into two streams: one based on change region computation between old and new composite service models [6][9]; The other based on compliance notion to find an equivalent state of the instance on the evolved model [1][12]. Adopting the first approach will reduce the time of migration determination, because all running instance of the same composite service model will share the same change region calculation result. However, it sacrifices migration flexibility by forbidding any migration of an instance entering the change region which may not break the soundness. The more live instances are forbidden to get migrated, the bigger waste of time is needed in dealing with composite service evolution, because all instances should be rolled back and redone and these work are not necessary if the migration approach is flexible enough. Besides, the complexity of change region calculation i.e., $O(n^4(n!)^2)$ is quite high [9]. Thus the calculation of the change region between original and evolved composite service model restricts efficient instance migration during evolution. The second approach based on instance compliance determination is more flexible than the first approach, such as tolerating changes in the loop or deletion changes during migration. But the restriction of this approach is that it inevitably faces the state explosion problem when the composite service model grows complex. When the nodes number of the model exceeds 50, it takes minutes to determine whether valid instance migration exists. Considering the possibility of existing large number of live instances in a system whose underlining model is evolving, the total migration time may become really long. Therefore, we propose a hybrid instance migration approach by combining the advantages of these two mainstream approaches in order to increasing the migration efficiency as well as its flexibility. We adopt the control flow analysis approach of decomposing the composite service model into a Single Entry Single Exit (SESE) fragment set and constructing a process structure tree (PST) based on that set. Chang region is calculated by identifying the changed SESE fragments in the PST, thus reducing the algorithm complexity to linear time [5]. Only instance running inside of a change region need individual migration determination. And its instance log is replayed on the local reachability graph of the entered change region, thus reducing the

ISBN: 978-1-61208-105-2

possibility of state explosion since the input model size of reachability graph calculation algorithm is much smaller.

Data flow correctness preservation is another inevitable aspect of instance migration problem. However its solution is seldomly discussed until now. This makes current instance migration solution fail to avoid the potential data flow flaws such as data missing or data mismatch. Recent work [1] points out that control flow change may also bring in data flow change during composite service evolution. However it does not illustrate the problem that dynamic instance migration may also result in data flow problems. In this paper, we elaborated the potential data flow problems during instance migration and propose a solution of combining change regions based on data dependence analysis. In this way data flow problems during migration can be avoided.

System administrators are the users of this approach, because when they are in charge of maintaining the whole system, they have to deal with the model change problem. This approach will greatly reduce their work of rolling back running instances as well as redoing the existing work. However, the data flow analysis in this paper is still not sophisticated enough for real life application. Concrete data structure analysis should be carried out in order to deal with different types of application in different scenarios. However, the data dependence analysis in this paper can effectively help reducing data flow problems, such as data missing or data mismatch. These problems are all critical problems in applying instance migration into real world application.

The rest of the paper is organized as follows. In Section 2, we discuss the motivation example of this paper. Section 3 introduces the preliminaries. In Section 4, we give the design overview of our hybrid instance migration approach, which is illustrated in detail in Section 5. Section 6 describes prototype demo and experiments. Finally, we wrap up this paper with some conclusions and future work in Section 7 and 8.

II. Motivating Example

To motivate our example, we refer to a real business loan application demo cited from IBM company web page. The application's Business Process (BP) model described in BPMN [2] is shown in Figure 1. a After the loan officer receive an loan applicant's detail information, including loan amount, repay plan, personal information, income information, and credit information, the process will split according to the loan amount. If the amount is less than 10,000$, the application is directly passed by a fast track approval. Otherwise, credit check and employment check have to be done before the application is recommended to his loan manager. Then the loan manager will firstly review monthly loan sales activity (i.e., company's current loan sale activity) and then loan history (i.e., past loan sales activity) to assess the bank company's running situation. After that, a loan approval decision will be made, based on the bank's business status as well as the applicant's information. If result gets passed, the notification and loan contract will be generated. In the last step, a reply email will be sent to the loan applicant. During loan composite service evolution, the process model later generates a new version shown in Figure 1. b There are three changes carried out between the two versions of loan model. Firstly, employment check and credit check are now required to proceed in parallel to reduce overall processing time. Secondly, a loan plan adjustment activity is inserted, allowing the loan officer to modify applicant's loan amount or repayment plan during the loan approval time after communicating with the applicant. Thirdly, in order to lower repayment failure, a third party risk check service is inserted before the loan manager manually makes the loan approval decision.

The earlier an instance can be migrated to an evolved composite service model, the more advantage of the new model such as performance improvement and functional adjustment it can enjoy in its future execution. However, not every point in the process model is safe for instance migration. For example, if transferring an instance on an unsafe migration point in Figure 1.a, it will result in execution deadlock after migration, whereas safe migration points won't cause this problem. Therefore, it is critical to find as many as possible safe migration points to enhance composite service flexibility while calculation complexity should not be too high.

III. Preliminaries

A. SESE Fragments and Process Structure Tree

In this paper, we used a process graph V = (N, E) to represent the BP model. A process graph has a finite node set N and control flow set E. N is classified in two types, action nodes and control nodes. Action nodes are in charge of concrete work implementation, such as Service Task in BPMN; whereas control nodes control the execution flow, such as Gateways or Start Event and End Event in BPMN

Process Graph, in general, can be decomposed into Single Entry Single Exit fragments [5]. One decomposition approach is to ensure all composed SESE fragments not overlapping each other on the same hierarchical level. This type of SESE fragment is called *canonical fragment*. Canonical fragments can be organized in a hierarchical way, i.e., a canonical fragment can be divided into child canonical fragments or compose a higher level parent canonical fragment. All Canonical fragments of process graphs in Figure 1. a and Figure 1. b are visualized by a surrounding of dotted lines.

In this way, a process graph can be represented by a process structure tree (PST) [5]. The root of PST is the entire process graph which contains all the canonical fragments. We use *parent(f)* to denote the fragment in PST which directly contain fragment f. Besides, fragments in the process graph have order relation in position between one another. Through a depth first graph searching algorithm, the precedence relation of fragment positions is determined. We use *precede(fx,fy,BP)* to denote that fx's position is always before fy's position in the BP graph regardless of the different depth first searching tree. And *paths(fx,fy,BP)* denotes all possible control flow paths that connect fx and fy in the BP graph.

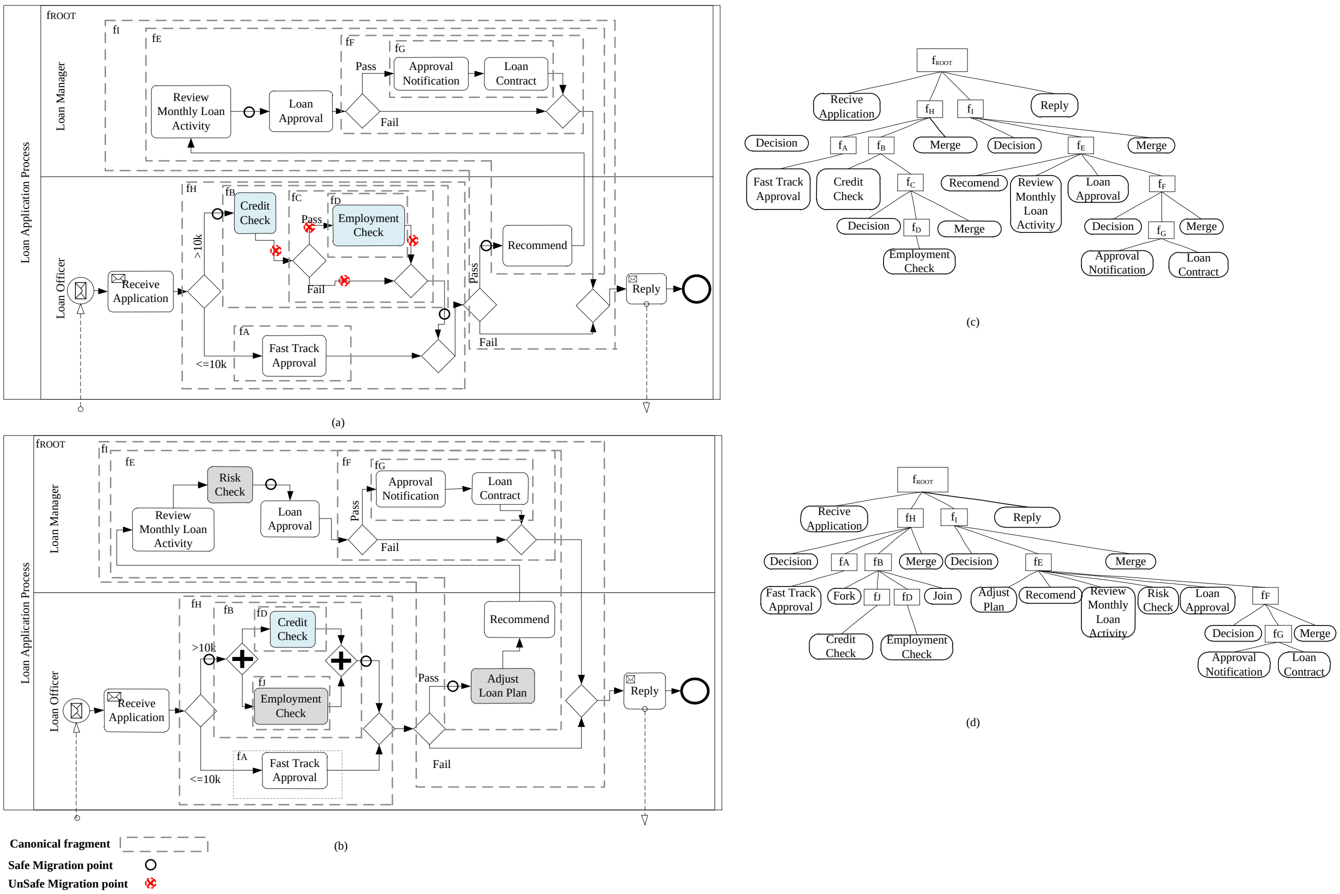


Figure 1. Versions V_1 and V_2 of bank loan business process modeled in BPMN and corresponding PST tree

B. *Soundness of Process Graph and SESE Fragments*

Similar to the soundness definition of a workflow graph, a sound process graph can be easily defined [5]. Process graph soundness ensures the execution correctness of the BP model, including liveness criterion which says each execution can be completed normally and safeness criterion which says each completion of a run is terminated properly with no tokens left inside the graph. Therefore sound process graph is free of execution deadlock or lack of synchronization. Soundness analysis of a process graph is made easier by calculating its PST and component canonical fragment set. According to the theorem 2 in [5], a process (workflow) graph is sound if and only if all its child fragments are sound and the process (workflow) graph that is obtained by replacing each child fragment with an activity is sound. And what's more strictly, if a fragment that is of any of the three types -- well-structured, unstructured concurrent and unstructured sequential fragment -- is sound then all its child fragments are sound. Therefore, we only decompose the process graph in a way that all component fragments are one of those three fragment types. In this way, every component fragments in the PST is guaranteed to be sound.

IV. DESIGN OVERVIEW

Figure 1. shows an overall view of our hybrid instance migration approach. Our approach is divided into two parts, static analysis and runtime analysis. Static analysis generates the changed region between two versions of process model, and mainly includes three steps. Firstly, change operation set is automatically calculated through comparison between the versions of a process model, using the approach proposed in [4]. Secondly, the change region set is calculated by identifying the affected region by each change operation. Thirdly, changed fragments in the changed region are combined according to the data dependence relation among them to avoid data flow flaws during migration.

After getting the change region set during evolution, we enter the runtime analysis part. Every instance on the same process model can use the same static analysis result, i.e., the changed region set, to implement the first step in runtime analysis. Firstly, instance state is collected from the instance log repository. It is compared with the changed region to decide whether the instance has entered a change region. If not so, the instance can resume execution directly on the new model. Otherwise it is required to carry out a compliance determination step which is the second step. In this step, new model's reachability graph is generated (using the approach mentioned in [10]) and a path which is identical with the migrated instance’s log is searched on the reachability graph. If search succeeds, then the instance’s state is transformed to the new state where the search stops. This is the third step of runtime analysis. Otherwise, it means the instance is not compliant with the evolved model and will be postponed migration until it steps out of the changed region. This is the last step of runtime analysis

V. HYBRID INSTANCE MIGRATION APPROACH

Dynamic instance migration can easily break the correctness of execution, such as the dynamic change problems mentioned in [6] and data flow flaws mentioned later in this paper. These problems are caused by transferring an instance’s running state on one process model to a different process model. From another point of view, this equals to running an instance on a merged process model. The merged model connects the old process faction which is between its start point to the instance migration point and the new process fraction which is between the instance migration point and its end point together. Thus the migrated instance’s underlying model correctness (including the control flow and data flow correctness) is very vulnerable to be broken if migration point selection is not controlled. In this section, to ensure the merged process model’s correctness, we propose a safe migration point selection approach based on change region set between old and evolved process models. Instances are only allowed to be migrated on these safe points in the aim of avoiding any error of live instance migration.

A. *Soundness Preservation*

If the merged process model with joints on a migration point set between old and new process model is sound, then migration on those points can avoid deadlock or lack of synchronization problems. Analogous to what is mentioned in [9], if migration points are outside the changed region during evolution, then the soundness of merged process model is guaranteed; otherwise migration is quite likely to end in execution error. We propose in this paper that, the changed region that guarantees control flow soundness during migration is the changed canonical fragment set during evolution. Because this approach does not need instance runtime information, it facilitates valid migration determination to be done once for all instances running on the same process model to be migrated, thus saving a lot migration time. When an instance is running outside the changed region, its state can directly mapped onto the new process model without any transformation, and continue execution with the new model without breaking control flow soundness.

The time to carry out migration thus is controlled with the help of *safe migration points* which are outside the changed region. If an instance’s running stage does not step on safe migration points, it will continue execution until it is on. Then, the instance will transfer to the new process model and finish execution in the end.

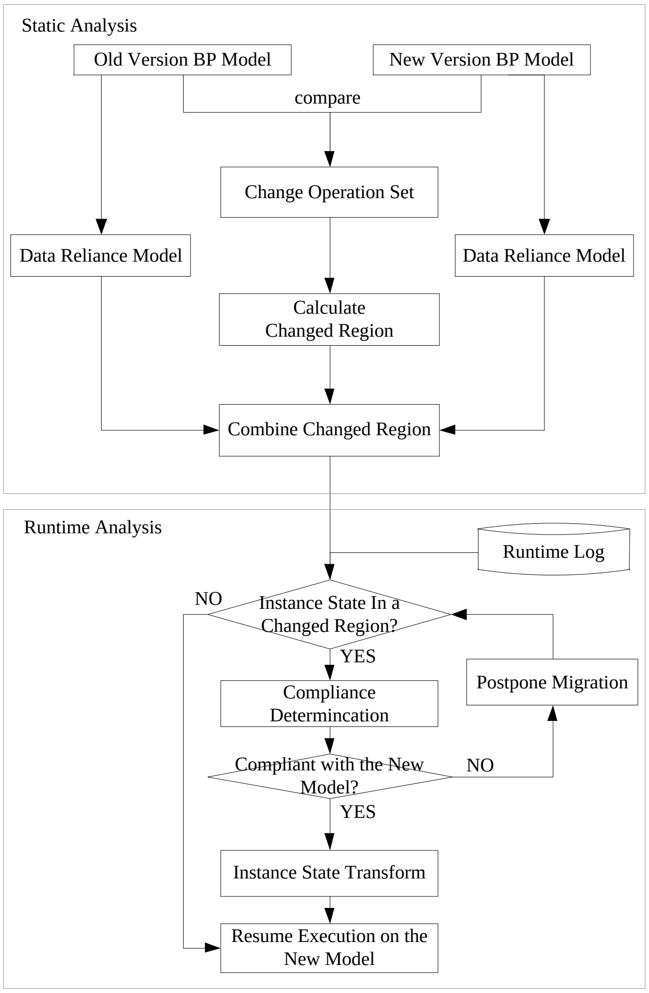


Figure 2. Overview of the hybrid instance migration approach.

B. *Computation of Changed Region*

Changed region calculation is implemented through analyzing change operation set which converts the old process model to the new one. To get the change operation set during evolution, change log can be recorded intentionally or automatically calculated through the approach mentioned in [4]. Because each type of change operation has a specified change effect area, the changed region set between two versions of the process model can be calculated by iterating each change operation in the change log and determining its corresponding effect region. According to what is mentioned in the previous section, the effect region of each change operation will always be the set of canonical fragments in order to guarantee control flow soundness during migration.

In TABLE I. , we enumerate 7 basic types of change operation (that is the same as what is mentioned in [4]) and their corresponding effect region function. Most of the effect region functions are straightforward to understand, such as the insertAction operation will only affect the region on its insertion point, though some may need more detailed explanation. The moveAction(V,x,a,b) operation, for

TABLE I. CHANGE OPERATION AND ITS EFFECT REGION

Change Type	Change Operation	Explain	Effect Region (ER)
Action-oriented	InsertAction(V,x,a,b)	Serially insert a new fragment x between two succeeding nodes a and b in process V. Changes effect is restricted to the insertion point.	ER=paths(a,b)
	DeleteAction(V,x)	Delete an existed action node x in V. Changes effect is restricted to the deletion point.	ER=x

	MoveAction(V,x,a,b)	Move an existing fragment x to the point between two succeeding nodes a and b in V. Change affected region is extended from x to x's new position.	if x < a,
			ER = paths(x,a)
			If x > b, ER = paths(b,x)

Fragment-oriented	InsertFragment(V,f1,a,b,f2) The generic operation InsertFragment is realized by: •InsertParallelFragment •InsertAlternativeFragment •InsertSequentialFragment •InsertCyclicFragment •InsertUnstructuredConcurrentFragment •InsertUnstructuredSequentialFragment •InsertComplexFragment	Insert a new fragment f1 between two succeeding nodes a and b in process model V, copying the structure of f2, and reconnection of control flow. Insertion type can vary according to fragment type of f2, such as parallel insertion or sequential insertion. Change affected region is f2 which is the parent fragment of f1 in the new process model.	ER = f2
	DeleteFragment(V,f1)	Delete fragment f1 from process model V.	ER = f1
	MoveFragment(V,f1,a,b)	Analogous to moveAction operation.	if f1 < a, ER = paths(f1,a) If f1 > b, ER = paths(b, f1)

instance, has an effect region extended from the action x's original position to its new position between a and b. If x is moved to a upstream position, then the ER function should equals all the fragments on the path from b to x; otherwise equals all the fragments on the path from x to a. Each change operation and its change effect are elaborated in detail in TABLE I. .

C. Data Flow Correctness Preservation

Uncontrolled instance migration can also bring in data flow flaws, such as data missing, data mismatch. This is because changed fragments may have data dependent relations, like downstream actions reads data written by upstream ones. If this data dependent relation is not considered, migration outside the changed region may cause data flow flaws. Action nodes in the merged model may read data not written by any nodes nor initialized, resulting in a missing data error. In other circumstances, data definition may be changed in the new process model, causing a mismatch between activities who write and read the same data. Thus, in order to ensure data flow correctness during migration, data dependent fragments in the change region set should be combined, i.e., to include data dependent fragments and the area between them in a larger changed fragment. Migration is only allowed to be taken outside the combined changed region in the aim of avoiding any data flow flaw. Data dependence relation between fragments in the changed fragment set is calculated by leveraging the data flow analysis technique as described in [7].

D. State Transformation in Instance Migration

Our hybrid approach allows instance migration in a change region if instance's execution log is compliant with the change region's new process model. If there is one execution path in the new process model that is equivalent with the instance log, then the instance is called compliant with the new model. Based on compliance, migration flexibility is enhanced, thus reducing the cost of aborting and redoing finished works. However, under this circumstance, instance state cannot be directly mapped to the new model and state transformation during migration is necessary [12].

VI. PROTOTYPE AND EXPERIMENT

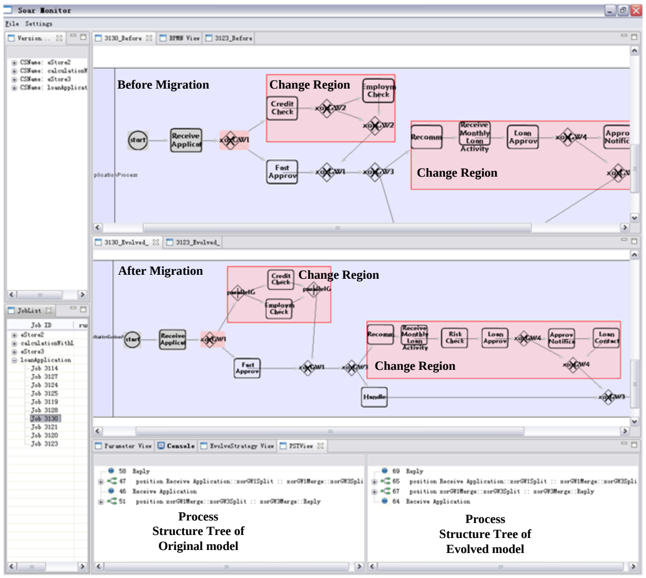
First, a snapshot of our prototype tool is given in Figure 3. It shows migrating a running instance of bank loan composite service in Figure 1. from model V_1 to a new model version V_2. First of all, the process structure trees of two process models is calculated and shown in the PSTView (in the bottom area) of the prototype tool. Then, change region set is calculated and is depicted by the rectangle box in the right middle of the prototype tool. After that, each instance is determined whether it can be safely migrated with the help of the change region set. For example, the state of the instance in Figure 3. is currently executing exclusive gateway of the process model. This instance is safe to be directly migrated because its state is outside the change regions (i.e., the red boxes). And its new state on the new process model is correctly calculated.

We carried out simulation experiments to demonstrate the performance of our algorithm in practice. In all simulation, we assume that the old and the evolved composite services are sound. The simulation experiment shows the relationship of composite service model complexity (represented by the number of action nodes in the process model) and change region calculation time. We vary the number of action node number in the composite serivce from 12 to 54 with incremental step length of 7. The results are depicted in Figure 4. It is shown that, when process complexity is increased to around 50 action nodes, the algorithm running time can still be counted by millisecond unit. And the increasing trend is linear, which is consistent with our change region calculation approach complexity. However, we don't give the comparison between change region algorithm in [9] and our approach, because the time complexity of their approach is too high, i.e., $O(n^4(n!)^2)$ and simple process model with only 19 action nodes cost around 0.5 second to compute its change region set. Our change region calculation algorithm is, however, millisecond unit, therefore much more efficient.

VII. Related Work

Existing instance migration approaches focus more on control flow correctness preservation, including change region based approach [6][9] and instance compliance based approach [1][12]. Change region calculation algorithm of former approach is quite slow due to its exponential complexity, thus is the bottleneck of this solution. Approach based on compliance notion has to adopt the reachability graph analysis so that instance log can be replayed and corresponding state can be found on the new definition. But when input model is complex, it will encounter state explosion problem.

Few work until now takes data flow correctness into consideration when dealing with instance migration. Rinderle-Ma et al. give the pre-conditions [1] of each dynamic change operations to protect data flow correctness during composite service evolution. Their description of pre-conditions, however, is too long-winded to express formally and can be quite fault-prone due to manual definition. [13] solves this problem by data dependence analysis, which is similar to our method in this paper.



```
58 Reply
47 position Receive Application::xorGW1Split :: xorGW1Merge::xorGW3Spli
46 Receive Application
51 position xorGW1Merge::xorGW3Split :: xorGW3Merge::Reply
```

```
69 Reply
65 position Receive Application::xorGW1Split :: xorGW1Merge::xorGW3Spli
67 position xorGW1Merge::xorGW3Split :: xorGW3Merge::Reply
64 Receive Application
```

Figure 3. Instance Migration Prototype Tool

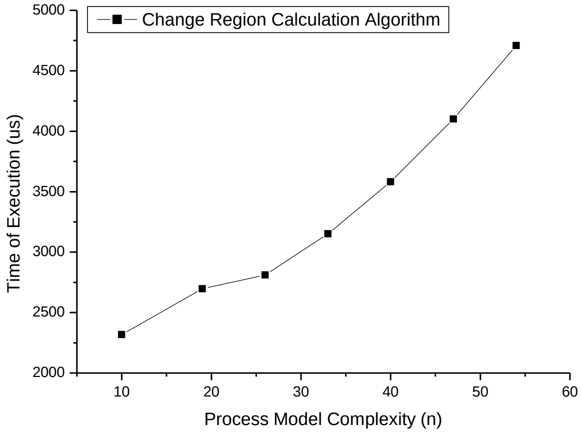


Figure 4. Time Complexity of Change Region Calculation Algorithm.

VIII. Conclusion

In this paper, we introduced a hybrid approach which calculates change region set and implements state transformation using reachability graph to solve the instance migration problem in composite service evolution. First, we introduce the SESE fragment and process structure tree definition and propose a change region calculation approach based on process structure tree comparison. This approach is linear time complexity which is proved by experiment results. Second, data flow problems that may occur during live instance migration is elaborated and data dependence analysis is adopted to solve the problem. Finally, we prototype and experiments are performed to show our approach's feasibility and effectiveness. Our future work includes instance migration in composite service protocol evolution.

Acknowledgment

This work was supported by the National High Technology Research and Development Program of China (863 program) under grant 2007AA010301, 2006AA01A106 and 2009AA01Z419.

References

[1] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*. 2007: Beijing : Tsinghua University Press.

[2] F-Q, Yang., *Thinking on the Development of Software Engineering Technology*. Journal of Software, 2005. **16**(1): pp. 1-7.

[3] Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification.Technical report, Object Management Group (OMG), February 2009.http://www.bpmn.org/..

[4] J. Küster, C. Gerth, A. Förster, and G. Engels, Detecting and resolving process model differences in the absence of a change log, in: M. Dumas, M. Reichert, M.-C. Shan (Eds.), BPM 2008, LNCS, vol. 5240, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 244–260.

[5] J. Vanhatalo, H. Völzer, and F. Leymann, Faster and more focused control-flow analysis for business process models though SESE decomposition, in: B.J.Krämer, K.-J. Lin, P. Narasimhan (Eds.), ICSOC 2007, LNCS, vol. 4749, Springer, 2007, pp. 43–55.

[6] C Ellis, K Keddara, and G Rozenberg. Dynamic change within workflow systems. Proceedings of the Conference on Organizational Computing Systems, Milpitas, California. ACM SIGOIS. New York: ACM Press, 1995:10–21..

[7] S. Moser, A. Martens, K. Gorlach, W. Amme, and A. Godlinski. Advanced verification of distributed ws-bpel business processes incorporating cssa-based data flow analysis. In SCC 2007, pages 98-105, Salt Lake City, Utah, USA, 2007. IEEE Computer Society.

[8] S. Rinderle-Ma, M. Reichert, and B. Weber. Relaxed Compliance Notions in Adapt ive Process Management Systems. in 27th Internat ional Conference on Conceptual Modeling (ER). 2008.

[9] W.M.P. van der Aalst. Exterminat ing the Dynamic Change Bug: A Concrete Approach to Support Workflow Change. Informat ion Systems Front iers 2001, 3(3): pp. 297–317.

[10] X. Ye, J. Zhou, and X. Song, *On reachability graphs of Petri nets.* Computers & Electrical Engineering, 2003. **29**(2): pp. 263-272.

[11] M. Reichert, S. Rinderle-Ma, and P. Dadam: Flexibility in process-aware informat ion systems. LNCS Transact ions on Petri Nets and Other Models of Concurrency(ToPNoC) 2 (2009) 115-135

[12] J Zeng, JP Huai, HL Sun, T Deng, and X Li. LiveMig: An Approach to Live Instance Migrat ion in Composit e Service Evolut ion. in IEEE International Conference on Web Services. 2008.

[13] LH Lam, Q Tang, ZL Zou, L Fong, and D Frank. Identifying Data Constrained Activities for Migration Planning. in IEEE International Conference on Services Computing. 2009

Distributed and Passive Web services discovery middleware for Pervasive services at the edges of Internet

Abdul Haseeb, Mihhail Matskin
KTH, Royal Institute of Technology, Sweden
Email: {ahaseeb, misha}@kth.se

Peep Küngas
University of Tartu, Ülikooli 18, 50090, Tartu, Estonia
Email: {peep.kungas}@ut.ee

***Abstract*—The advent of mobile computing devices and development of wireless and ad hoc networking technologies as Bluetooth, RFID etc has led to the growth of infrastructure-less pervasive environments. Most often, these environments lie at the edges of Internet, i.e., they are disconnected or sparsely connected to rest of the world. In order to exploit the access to such edges of Internet, an interoperability middleware, capable of dealing with lack of communication infrastructure, is needed. In this paper, we propose a solution that synergizes P2P technology, message queuing support and a passive distributed UDDI to exploit Web services in infrastructure-less edges of Internet. We abstract communication heterogeneity and prove that passive communication mode performs better than active mode of communication (i.e., existing solutions in literature) in terms of Web services discovery.**

I. INTRODUCTION

Edges of Internet represent disconnected or sparsely connected sub-networks in a cascaded flow/collection of networks and associated technologies. These edges can range from wired to Wi-Fi and to ad hoc networks (for instance: mobile networks, war front networks, fire fighting networks, futuristic shopping malls, robot swarms etc.). In other words, we view a huge number of heterogeneous devices pervasive in everyday life (ranging from high-end servers to mobile devices and resource-constrained entities). These pervasive devices create communication clusters in multiple interacting yet independent networks. In some cases (due to mobility) these environments remain in isolation and disconnected from rest of the world, thus forming edges of Internet.

There is a need to exploit the access to such diversified natured and disconnected network environments without any manual intervention, i.e., an interoperability platform to glue various edges of Internet together. Interoperability of edges of Internet must deal with connectivity aspects (internal and connectivity with Internet). Internal connectivity can be either based on direct message passing or via shared memory. Connectivity with Internet, however, depends upon the openness of the system either using available connectivity or by using disconnected reconciliation-mode using mediators.

Web services technology has become a standard for interoperability problems. It relies on the ability to locate and acquire specified services. For such purposes UDDI provides the functionality to search for Web services. Existing UDDI technology uses a centralized repository to store pointers to registered Web services. Centralized approaches have many well-known and discussed drawbacks [8]. Moreover it is not viable to exploit centralized/semi-centralized solutions in infrastructure-less environments. Current solutions [16, 17] of service discovery in infrastructure-less environment are primarily broadcast based and inefficient in network resource utilization. These solutions do not address the issues of scalability and network-wide reachability. Moreover, issues of mobility and adaptability of protocols to dynamic environments is not addressed.

Apart from a decentralized UDDI requirement, there is a need to establish asynchronous interactions between entities. By using techniques developed as part of traditional Message Oriented Middleware (MOM), asynchronous messaging can be built on top of synchronous interactions by introducing a queuing system for storing and forwarding of messages. We refer to such queuing system as message post boxes. Mediation is useful in scenarios where P2P communication between entities is not possible.

In this paper we propose and evaluate an interoperability middleware that synergizes P2P technology, message queuing support and distributed UDDI to exploit Web services in infrastructure-less edges of Internet. This paper gives a detailed evaluation of a passive distributed UDDI, a loosely coupled distributed UDDI for infrastructure-less communicating and heterogeneous systems. We use robot swarms [3] for our evaluation, but the proposed principles are applicable to any infrastructure-less environments.

The rest of the paper is organized as follows. In Section II, we analyze Web services solutions in pervasive environments; Section III elaborates a representative edge. In Section IV and Section V, we present and evaluate our proposed distributed Web services discovery solution.

II. WEB SERVICES AND PERVASIVE ENVIRONMENTS

Pervasive environments can be classified into infrastructure-based and infrastructure-less environments. We focus on Infrastructure-less/ad hoc environments. Examples of such environments can range from fire fighting, war-front activities, robot swarms to space exploration research.

Service discovery architectures like Jini [12], Salutation and Salutation-lite [13], UPnP [14] and Service Location Protocol [15] have been developed in the past to efficiently discover Web services in infrastructure-based environments. But these discovery architectures rely on a central lookup for service registration and discovery. Central lookup is inappropriate in ad hoc environments due to its dependence on a central point/node. Moreover the central point/node can be mobile and unreachable during certain points of system execution.

Solutions [16, 17] for service discovery for infrastructure-less environments primarily utilize broadcast-driven nature of the underlying ad hoc network. However, it was shown by [18] these approaches are not efficient and scalable for discovery in large-scale environments. Thus researchers tried a leverage P2P technology for distributed service discovery [2]. DUDE (Distributed UDDI Deployment Engine) [7] was among the earlier efforts that used distributed hash tables as rendezvous mechanism between distributed service registries. However, their approach cannot cope with high dynamism, mobility and in scenarios where entities cannot communicate in a P2P fashion. For infrastructure-less environment of mobile users/agents a messenger approach [9] was proposed as well, which provided mechanism for dynamic management of distributed UDDI in the absence of communication infrastructure. Basis philosophy of messenger is to update cached Web service descriptions at UDDI when a mobile user/agent comes in communication coverage area of some UDDI. This approach suits best when entities have unpredictable coverage, however, the solution did not consider collaborative service discovery. Similarly a proxy node at the edge of network to serve Web services to mobile users was proposed in [19]. But it dependence on wired infrastructure makes it inappropriate for infrastructure-less environments.

In summary, infrastructure-less environments are attributed by lack of stable connectivity [3, 9], mobility, heterogeneous communication capabilities [3] and lack of P2P communication. This puts forward the requirement for a solution which can overcomes the lack of availability of both service requesters and service repositories at the same time, asynchronous and mediator-based communication [5] for scenarios where P2P communication is not possible.

Our proposed methodology synergizes P2P technology, message queuing support and distributed UDDI to exploit Web services in infrastructure-less edges of Internet. Our proposed solution assumes environment "mailboxes" or "message postboxes" (for instance RFID tags), which can be used for communication when P2P communication between entities is not possible. Our solution also uses a passive mode of communication and service discovery. To the best of our knowledge, no earlier solution has used a passive mode of communication. In our evaluation we prove the effectiveness of passive communication mode in terms of network load while maintaining an acceptable latency.

III. Swarm of Robots – a representative edge

We take swarm of robots as a representative use-case/edge [3]. The edge comprises of an environment of low-cost robots operating in a dynamic environment. There is no assumption on the availability of wireless capability of robots - i.e., few robots have wireless capability, while the others use RFID tags for reading/writing data. Our objective is to establish interoperability of swarm both with environment entities and with outside world in a symmetric way.

We use Web services for exposing robotic functionality. Thus each robot is abstracted to a Web service interface. Robotic actions are based on a local knowledge-base and collection of swarm knowledge is represented as a knowledge gateway. Knowledge gateway is assumed to be a server, equipped with wireless capability, which can provide resource-intensive services to swarm entities. Apart from that any swarm entity with connectivity to outside world can serve as swarm gateway. We do not assume wireless capability on each robot; however, it is a reasonable assumption to have at least one robot with wireless capability that can communicate to Internet (see Figure 1).

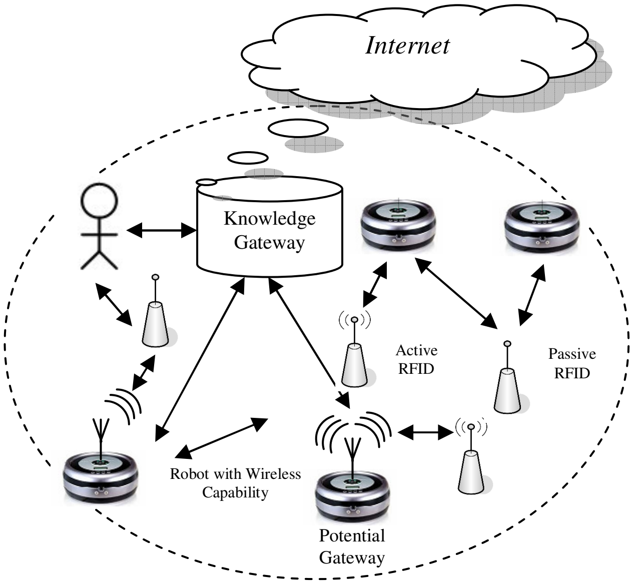


Figure 1. Swarm of robots – a representative edge.

IV. Distributed services discovery middleware

We take the basic design philosophy from JXTA [1] - a P2P stack and XML-based set of peer platform protocols and services. In contrast to JXTA not only does our middleware rely on communication capabilities on entities, but also due to dynamism (caused by entity mobility) it supports dynamic joining of entities in different clusters during execution.

A. Conceptual classification of entities

Entities are classified in the following manner:

- **Rendezvous computing entities (RCE)** are the entities with Wi-Fi (or some other high-level) connectivity which enables them to communicate outside a domain or to external world. RCE serve for dissemination of requests from one cluster[1] of entities to another as routing peers [2]. RCE can create a virtual overlay network with other RCE (using Rendezvous Peer View protocol – RPV) to serve clusters [6] around mediators.
- **Edge computing entities (ECE)** are the entities those can't communicate in a P2P fashion with other entities and require mediation for their communication.

[1] We refer to entities around MR as *entity-group/cluster.* I.e. Set of entities that can have a pair-wise communication among each other (P2P or via MR).

- **Message Relays (MR)** are the mediators (e.g. RFID tags) that serve ECE to communicate their messages.

For the sake of clarity, we elaborate few essential components. Detailed discussion about the architecture can be read in our previous work [4, 20]:

- **Message Transport Layer (MTL)** is an implementation of asynchronous communication channels. Using MTL, entities and associated services can join and leave at any point of system execution, thus achieving loose coupling.
- **Local Service Registry (LSR)** is a local cache for Web service discovery. For LSR we implemented a light-weight UDDI [4]. Incoming request are first searched from LSR before being cached to **Query Response Cache (QRC)**. QRC caches incoming service requests and propagates back response messages using MTL.
- **Entity Discovery Registry (EDR)** serves as a record for discovered entities. EDR is used to create a semantic topology based on set of services an entity advertises and its expertise (i.e., semantics of services an entity offers).

B. Semantic Query Propagation

Notion of RCE, RPV protocol and dissemination of messages from one cluster to another provides quick dissemination of messages across clusters, while MR performs dissemination of messages within a cluster. To incorporate semantics and to perform semantic-based query propagation to appropriate entities, we use a model in which entities publish their *expertise* (semantics of services) in the network along with their exposed Web services.

Basic philosophy of creating a semantic topology is to re-route the query to an entity which is likely to answer the query instead of broadcasting or sending the query to random entities. For this purpose we use a shared common ontology. Entities publish their expertise in the network and the knowledge of an entity about the expertise of other entities forms a virtual semantic topology. According to our definition, an entity knows another entity in semantic topology if it can compute the semantic distance of a query to target entity's published expertise. Examples of expertise can be *Weather Information Services*, *Cleaning Services* etc.

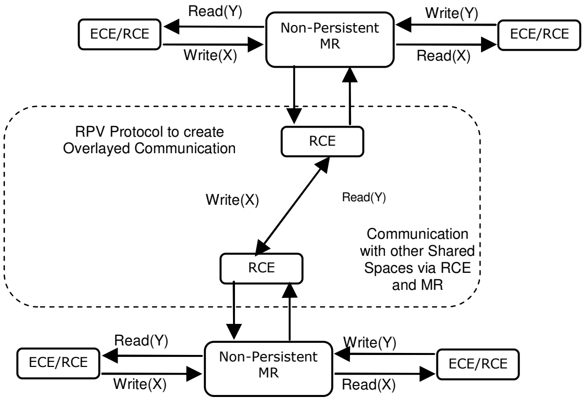


Figure 2. RCE overlay formation & communication between ECE & MR

C. Active and Passive communication mode

In term of communication, entities can follow two modes of communication.

1. Active Communication Mode

- Web service descriptions are pushed to other entities/MR.
- Active mode corresponds to a normal Web service publishing to UDDI.
- Service descriptions are first pushed inside a cluster via MR and later RCE disseminates them to rest of network.

2. Passive Communication Mode

- In passive communication mode a Web service discovery occurs when an entity's (RCE/ECE) request is answered by some entity/MR. In other words, entities don't publish their service descriptions unless requested.

MR are considered as non-persistent communication white-boards, which are prone to conflicts in case of concurrent access. In order to avoid conflicts entities employ a resource locking mechanism.

- **Conflicting mode** refers to a greedy mode in which entities don't wait for another entity to release MR.
- **Conflict-resolution mode** refers to a mode in which an active push or a passive service request locks the MR. Lock is released when either of the condition is true

1) Passing of service description or message request to another entity (i.e., at-least one entity has read the initiator's message).
2) Time-out

Due to constant mobility (and entities being out of communication range/coverage), lock to MR is released when at-least one entity has the initiator's message.

D. Entity and Web services registration

Upon a system bootstrap, entities (RCE and ECE) only have their own published Web services in LSR. Similarly their EDR is empty as well. Thus entities perform a registration process. For active communication mode registration includes Web services descriptions and a set of entity's expertise. However for passive communication mode registration only includes presence and endpoint related information.

Registration marks MR as an information container for services and entities surrounding it. Latter serves for discovery of entities when entities receive a list of other registered entities from the MR. This is a *naïve* epidemic way of discovering other entities in the environment.

RPV protocol enables RCE to organize them into a virtual overlay as shown by Overlayed communication in Figure 2. For RPV each RCE maintains a local list of active RV (Rendezvous View) and performs periodic exchanges of random RV to other RCE in its active RV. RCE also perform periodic purge of non-responsive RCE from its RV.

E. *Distributed Web services discovery*

In our middleware each entity serves as a small scale UDDI using its LSR [4, 20]. Dynamic and partial view of network doesn't allow the leverage of a hash function for storing service descriptions and utilization of rendezvous mechanism between multiple registries [11] on ECE or RCE. Rather, we consider the following approach:

1. Upon receiving a query, entities check their LSC and recommend a target entity if a mapping of query with known expertise in semantic overlay is found.
2. The requesting entity can then take following actions:
   1) Re-query for a semantically similar expertise (if exact match is not available)
   2) Trigger of discovery protocol to query the services of recommended entity.
   3) Forwarding of request via RCE to other clusters, if request remains unanswered or is published again.

A detailed elaboration and rationale of all the above principles can be read in our previous work [20].

V. Experimental Evaluation

The proposed middleware, in this paper, has been used in actual robot swarm [3]. Here in evaluation, we highlight the theoretical results using Player/Stage Simulator [10]. The communication infrastructure modeled for experiments is attributed by instability and partial communication coverage (attributed by robotic mobility). Web service descriptions are expressed in form of extended RDF triplets [4].

A. *Discoverable services*

Discoverable services refer to those services that have more than one point of discovery in the system. In other words, if a Web service is cached at another robot other than the service host/provider robot then that service is considered a discoverable service. Service discovery and invocation are considered different independent steps in service management and an additional host robot for Web services gives a higher probability of discovering a Web service when host robot is inaccessible (due to partial failures).

Measurement of discoverable services with respect to time measures the number of Web services that get cached at any other robot during the system execution. In other words, a higher number of discoverable Web services mean higher fault tolerance of system towards transient failures. Experimental result (Figure 3) shows discoverable services metric in the absence of conflict resolution. Experimental data is taken for varying Robot-RFID cluster ratios which is the number of robots sharing a MR/RFID.

Results reveal that the active communication mode achieves better performance in terms of discoverable Web services and with an increased Robot-RFID cluster ratio the performance deteriorates. This decrease in number of discoverable Web services, with higher Robot-RFID cluster ratio, is caused by increased number of conflict/messages-requests. This aspect will be shown in the comparison of active and passive communication mode

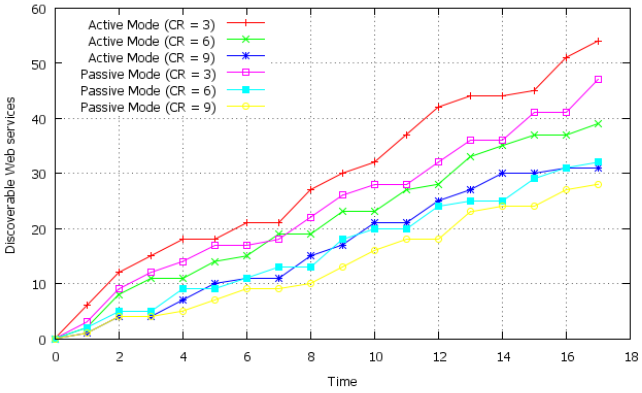


Figure 3. Discoverable Services without Conflict Resolution

Results reveal the effectiveness of active mode of communication over passive mode. This result gives a partial analysis, as we will observe latter that such improvement in number of discoverable Web services comes with an additional cost of increased bandwidth consumption in case of active communication and no real added value is achieved in terms of performance of actual Web services discovery.

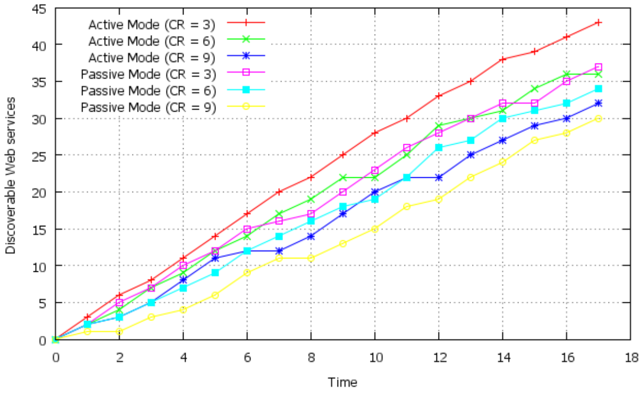


Figure 4. Discoverable Services with Conflict Resolution

Same experiment is performed with conflict resolution (Figure 4) as well. In this case the number of discoverable Web services with respect to time reveals a similar trend as that of non-conflict resolution (Figure 3). Curves of both active and passive mode (Figure 4) are mostly similar for various settings. Active communication mode still achieves better results as compared to passive communication mode i.e., more services are discoverable at any given time instant.

B. *Impact of Cluster Ratio on Average Messages at a MR*

In this experiment, we compare the impact of robot-RFID cluster ratio on the average number of messages disseminated in the environment (Figure 5). The result highlights a significant rise in average disseminated messages in active communication mode when robot-RFID cluster ratio is increased. On the other hand passive communication mode gives better results (i.e., fewer messages disseminated in the environment – less bandwidth intensive).

We can observe that, even with a higher robot-RFID cluster ratio, the rise in number of disseminated messages in passive communication mode is better to that of active communication mode. If RFID/MR memory size is increased, passive communication mode with a higher Robot-RFID cluster ratio as compared to that of active communication mode in same setting gives better results. This is shown in the overlap of “Active Mode – average msgs with RFID memory = 3” over “Passive Mode – Average msgs with RFID memory = 6”. In other words, passive mode of communication is less bandwidth intensive even when a higher Robot-RFID cluster ratio is used.

Results reveal that passive communication mode exhibits controlled losses even with increased robot-RFID cluster ratio. With a higher RFID memory the results of passive communication mode improves further as compared to that of active communication mode. Active communication mode on the other hand incurs more losses with higher robot-RFID cluster ratios.

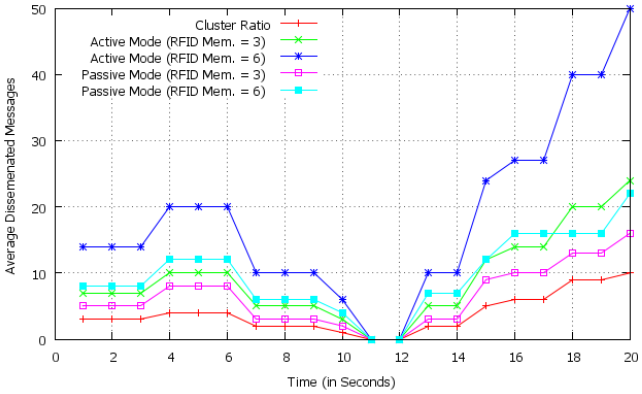


Figure 5. Impact of Cluster Ratio on Average Messages

C. *Message re-request overhead in Active vs. passive mode of communication*

Computation of number of message re-requests (see Figure 6 for non-conflict resolution mode) highlights the fact that, in general, there are fewer conflicts with smaller robot-RFID cluster Ratio. Passive communication mode serves better as it has less overhead (curves of passive mode with various settings show that numbers of conflicts are always lower than that of active mode of communication). In passive mode the conflicts are the number of request losses, while in active mode of communication conflicts are the service description losses. With a higher RFID memory size and lower robot-RFID cluster ratio both communication modes show similar results.

In case of conflict resolution mode (Figure 7) there is a significant difference in active and passive mode of communication. Such a difference is insignificant with higher robot-RFID cluster ratios. But with increased RFID memory size, passive mode of communication with conflict resolution gives improved results (lesser message re-request overhead). The curve for “Passive Mode – robot-RFID cluster ratio = 6” over “Active Mode – robot-RFID cluster ratio = 3” reveal this fact. The reason of such a behavior is the impact of service description memory requirement as compared to that of service requests in case of passive mode of communication. Such a behavior continues further with higher memory sizes of RFID/MR.

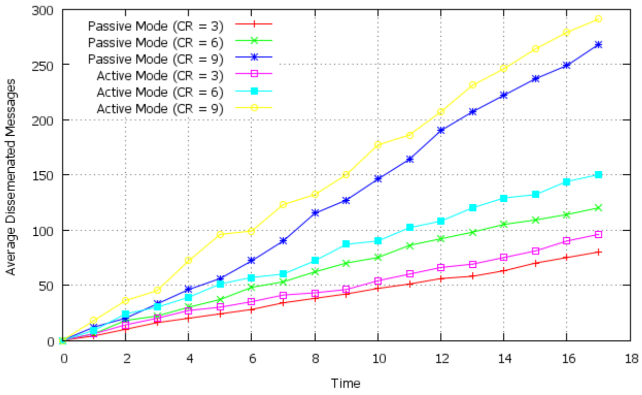


Figure 6. Active vs. Passive communication without CR

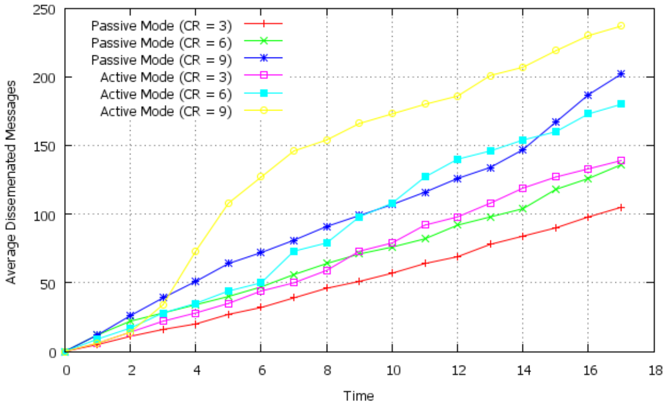


Figure 7. Active vs. Passive communication with CR

The comparisons (Figure 6 and 7) show that passive mode of communication serves much better in terms of number of conflicts and messages overhead as compared to active mode of communication. Passive mode shows better results with higher robot-RFID cluster ratio with higher memory size and with improved RFID memory it can support higher robot-RFID cluster ratios.

D. *Impact of caching and semantics in Service Discovery*

Lastly validation of the impact of caching and semantics in service discovery is done. Passive mode of communication is used with conflict resolution with different settings of robot-RFID cluster ratio. Results (see Figure 8) reveal that Web services discovery with caching augmented with semantics provides the best results (i.e., least number of hops/messages required for Web services discovery). There can be few abnormalities in result, for instance execution 14 shows better performance of syntactic discovery as compared to semantic service discovery with caching. This case represents a particular case exhibits a scenario in which a query is routed to a cluster (based on caching results) but due to mobility host robot has joined a different cluster – thus

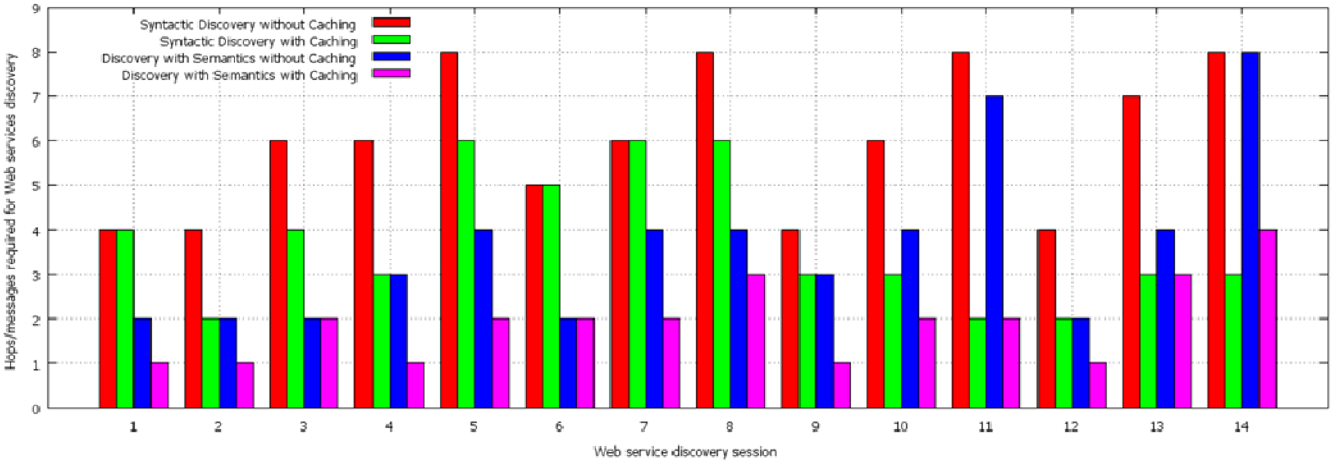


Figure 8. Impact of caching and semantics in service discovery

stale information in service cache effects additional number of hops across clusters for service discovery.

E. *Overall analysis*

Experimental results reveal the effectiveness of passive mode over active mode of communication. Active mode of communication (as employed in literature) gives result in number of discoverable services, however it does not fare well in terms of conflicts, higher robot-RFID cluster ratio and message re-request overhead. Moreover last result validates the improvements in service discovery and invocation of passive mode augmented with semantic discovery.

VI. CONCLUDING REMARKS

This paper gave a detailed evaluation of a passive distributed UDDI, a loosely coupled distributed UDDI solution for infrastructure-less communicating and heterogeneous systems. We proved, with the help of extensive experiments, that passive mode of communication performs considerably better than active mode of communication (i.e., existing solutions in literature) in terms of Web services discovery.

REFERENCES

[1] Sun Microsystems Inc., “JXTA-SOAP bindings”.

[2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: Scalable P2P lookup service for internet applications”, SIGCOMM, ACM, 2001, pp 149-160.

[3] Knowledge Environment for Interacting Robot Swarms, http://www.roboswar.eu , (last accessed: August 31, 2010).

[4] A. Haseeb, M. Matskin & P. Küngas, “Light-Weight Decentralized Autonomic Web Service Discovery for Systems with Heterogeneous Communication Capabilities”, 12th IASTED IMSA’08, Hawaii.

[5] B. K. Kim, M. Miyazaki, K. Ohba, S. Hirai, and K. Tanie, “Web Services based robot control platform for ubiquitous functions”, Proc. IEEE Int. conf. on Robotics and Automation, , 2005, pp. 691- 696.

[6] Z. Wang and Y. Hu, “A P2P Network Based Architecture for Web Service”, Proc. IEEE Int. conf. on Wireless Communications, Networking and Mobile Computing, IEEE, 2007, pp. 3446-3449.

[7] S. Banerjee, S. Basu, S. Garg, S.J. Lee, P. Mullan, and P. Sharma, “Scalable Grid Service Discovery based on UDDI”, Proc. ACM 3rd Int. workshop on Middleware for grid computing, 2005, pp. 1-6.

[8] M. Cai and M. Frank, “RDFPeers: A Scalable Distributed RDF Repository Based on a Structured P2P Network”. Proc. ACM WWW conference, 2004, pp. 650 - 657.

[9] Z. Maamar, H. Yahyaoui, and Q. H. Mahmoud, “Dynamic management of UDDI registries in a wireless environment of web services: Concepts, architecture, operation, and deployment”. Journal of Intelligent Information Systems, Springer, 2007, pp. 105-131.

[10] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems”. Proc. 11th Int. Conf. on Advanced Robotics (ICAR), Coimbra, 2003, pp. 317-323.

[11] Q. Lin, R. Rao, and M. Li, “DWSDM: A Web Services Discovery Mechanism Based on a Distributed Hash Table”. Proc. IEEE 5th Intl. conference on Grid and Cooperative Computing, 2006, pp 176-180.

[12] K. Arnold, B. Osullivan, R. W. Scheifler, J. Waldo, and A. Wollrath. “The Jini Specification (The Jini Technology)”. Addison-Wesley, Reading, MA, June 1999.

[13] The Salutation Consortium Inc 1999. Salutation Architecture Specification (Part 1), Version 2.1 Edition. http://www.salutation.org, last accessed: August 31, 2010.

[14] R. John, “UPnP, Jini and Salutaion - A Look at some popular Coordination Frameworks for Future Network Devices”. Technical report, California Software Labs, 1999.

[15] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, “RFC 2165: Service Location Protocol”, June 1997.

[16] S. Helal, N. Desai, and C. Lee. “Konark-A Service Discovery and Delivery Protocol for Ad-hoc Networks”. Proc. IEEE conf. onWireless Communication Networks, USA, 2003, pp: 2107 - 2113.

[17] D. Tang, C. Chang, K. Tanaka, and M. Baker. “Resource Discovery in Ad hoc Networks”. Technical report, Stanford University, August 1998. CSL-TR-98-769.

[18] D. Chakraborty, "Service Discovery and Composition in Pervasive Environments", PhD Thesis, June 2004.

[19] C. Pullela, L. Xu, D. Chakraborty, and A. Joshi. “Component based architecture for mobile information access”. Proc. Intl. Workshops on Parallel Processing, 2000, pp: 65 - 72.

[20] A. Haseeb, M. Matskin, and Peep Kungus, “Distributed Discovery and Invocation of Web Services in Infrastructure-less dynamic environments”. Pro. IEEE Intl. Journal of Web Services Practices (IJWSP), Vol. 4, 2009.

Federated Authentication Mechanism with Efficient ID management

Ryu Watanabe and Toshiaki Tanaka
KDDI R&D Laboratories, Inc.
Ohara 2-1-15 Fujimino Saitama, Japan
Email: ryu@kddilabs.jp, toshi@kddilabs.jp

***Abstract*—In order to enhance user privacy and reduce management costs for identity providers, in this paper, a federated authentication mechanism with cryptographic ID management method is proposed. Based on the proposal, a proto-type was implemented and performance evaluations were carried out. The evaluation results shows feasible performance for practical implementation.**

***Keywords*-Identity management; Single Sign-on; PKI.**

I. Introduction

The OpenID [1] authentication mechanism is one of several such authentication mechanisms that make possible the realization of a single sign-on (SSO) service for Internet web sites (WEB services). Under the SSO environment, once a user is authenticated on an authentication site, the user can visit related service sites without the need for any additional authentication on each service site [2][3][4]. Therefore, the SSO technique liberates users from the nuisance of individual ID (in this paper, the term ”ID” indicates the identifier of the user account) and password handling. Therefore, users have to keep secure only one ID and password pair. The ID form in OpenID is ideally suited to existing Internet technology because it employs the URI or XRI form. Currently, OpenID is being widely adopted by Internet services such as blog sites or social network service (SNS) sites. In addition, the fact that some major service providers provide the OpenID authentication service has also contributed to its spread. Currently, however, almost all OpenID providers (OP) issue IDs to users via simple user confirmation using e-mail. Therefore, it can be said that such IDs are not assured for SSO services. For this reason, these IDs are not appropriate for economically significant services such as a shopping service. If the ID is issued via strict checking of user identity using some kind of credentials, the IDs can be regarded as having high assurance. However, this is not realistic because it is difficult to perform checks of all the IDs that have been already been issued to users through simple registration.

On the other hand, in order to make a contract for a cellular phone, users normally have to submit some kind of identification to a mobile phone company (especially in Japan). Therefore, the IDs which are associated with mobile phones are highly assured by means of the registration process. If an OpenID can acquire the assurance of a cellular phone’s ID, it can be said that the OpenID also has assurance.

For the reason given above, the authors had already proposed a federated authentication technique with cellular phone [5]. In our previous proposal, we federated OpenID with PKI-based authentication on cellular phones, which require a strong off-line identity check for contacts. By binding a user’s OpenID with an ID of his/her mobile phone, the level of assurance for user identity is increased. In addition, the mobile phone is also used in each authentication for service use. Thus, it is used like a security token and it also contribute to enhancing user authentication.

However, in our previous work, a user ID management problem remained. The OpenID basically uses a unified ID as the user identifier on each service site, called a relying party, in order to be identified as the same person on each service site. This policy is applied for user convenience. However, the use of a unified ID for a single sign-on technique causes a privacy problem called linkability. In order to resolve this problem, the use of a transient pseudonym is widely adopted. The OpenID can also use this idea. However, in this case, an OpenID provider has to handle many generated IDs. For this purpose, we introduce a cryptographically generated ID management technique. By using this technique, transient IDs are generated from a unique user ID retained by the identity provider and the identity provider only keeps the keys for ID generation. Therefore it is expected that the ID management cost imposed on identity providers can be greatly reduced. In this paper, we describe our implementation based on our proposal and also report the evaluation results.

Here, we outline the structure of this paper. In this section (Section I), the authors give the background to our research as an introduction. Then related work relevant to our research is presented in Section II. In Section III, the concept of use of cryptographical ID management is described. Then, in Section IV, an implementation based on our proposal is presented and the evaluation results are shown. Finally, our research is summarized in Section V.

II. Related work

In this section, related work relevant to our research is described. At the end of the section, the purpose of this paper is stated.

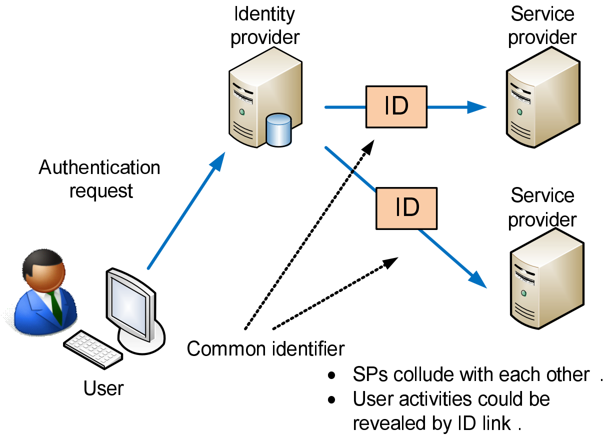


Figure 1. Problem of linkability

A. Single Sign-on (SSO)

SSO is a method of authentication and access control. Once authenticated on one identity provider, users have permission to use resources or visit various websites, etc. Therefore, this technique liberates users from password fatigue caused through having to handle many different ID and password pairs. In addition, the fact that a user merely has to keep a single unique ID password pair contributes to the enhancement of user security.

In order to realize an SSO scheme, there are many specifications or implementations. The Liberty Alliance Project (LAP. Currently, the Kantara Initiative has taken over this project.) [6][7] , security assertion markup language (SAML) [8], OpenID, and CardSpace of Microsoft are examples. Originally, they were specified or implemented only for an SSO scheme. Currently, the scope has been enlarged and they deal with user information (identity) and are referred to as an identity management mechanism (IdM).

1) Linkability: Under the SSO environment, there is a known privacy problem called ”linkability” due to the treatment of user identifiers. In the ordinary flow of an SSO technique, a service provider delegates a user authentication to an identity provider. Then, the identity provider checks the condition of local user log-on and sends the results to the service provider. In this case, in order to distinguish between the identity provider and the service provider, some identifier is used. If the same identifier is used by an identity provider and also by multiple service providers, a privacy problem arises. If service providers collude with each other, the user’s activities on each service provider are linked using this common ID (Figure 1).

B. OpenID

The OpenID mechanism is a decentralized authentication scheme for the SSO mechanism. OpenID users identify themselves with a URI and XRI. In the first specification of the OpenID mechanism [9], the idea of an identifier federation using a handle to connect an identity provider and a service provider like the SAML mechanism was not employed. In the OpenID scheme, the identity provider and service provider are referred to as the OpenID provider (OP) and relying party (RP), respectively. An RP does not have to prepare a local account (local ID) for users. Instead of a local account, an RP can use the user’s OpenID as the identifier for user identification. Under the concept of the OpenID mechanism, users are supposed to be identified with the same identity (that is a user’s OpenID) on every RP site. So the OpenID mechanism is completely unable to avoid the linkability problem because it is convenient for users to be identified as the same identity (person) on every blog site. However, the next version of the specification (OpenID Authentication 2.0 [10]) introduces an OP identifier, which indicates the OP location, to the specification and the users do not have to announce their OpenIDs to RPs. Therefore, the linkability problem can be avoided by implementation on an ID generation and distribution mechanism. Therefore, the ID generation mechanism for this problem is still an open issue. The simple answer is to use random IDs between an RP and an OP for user identification.

C. ID Assurance

The ID assurance level refers to the identity check of users; that is, when an ID provider generates a user account and issues an ID to a user, it refers to how they confirm the user’s identity.

For instance, ordinary Internet services such as blog or BBS sites require only an e-mail address for generating user accounts. A user who holds a free mail address can generate user accounts on a service. But the provider does not confirm the user’s real identity. Therefore, the provider cannot verify the existence of the user. In this case, it is said that the assurance level of the ID is low.

On the other hand, for contract of mobile phones, users usually have to show some kind of identification such as a driver’s license or passport (at least in Japan) in an off-line procedure. In this case, therefore, the assurance level of an ID bound to the mobile phone is high and the ID also provides strong assurance for online services.

The electronic authentication guideline of the NIST standard [11] (the guideline for e-authentication) is one example of a standard used for user identity proofing for registration. In the guideline, the registration levels are defined in accordance with the OMB guidance [12], which describes four identity authentication assurance levels for e-government transactions.

D. Federated Authentication Mechanism

In order to enhance the assurance level of user ID and the security level of user authentication, we have already proposed a federated authentication mechanism.

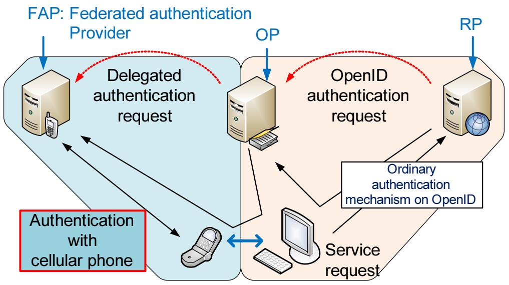

FAP: Federated authentication Provider
OP
RP
Delegated authentication request
OpenID authentication request
Ordinary authentication mechanism on OpenID
Authentication with cellular phone
Service request

Figure 2. Federated Authentication Mechanism

The concept of the mechanism is shown in Figure 2. In Figure 2, the mechanism is implemented with an OpenID authentication scheme. First, an RP delegates its user authentication to an OP. Then the OP re-delegates the request to an FAP (federated authentication provider). The FAP authenticates the user by using the user's cellular phone and returns the authentication result to the OP. The ID bound to a cellular phone has a high level of assurance. Therefore, the identity of the OpenID owner can also be confirmed. In addition, in our proposal, cellular phones are also used in each authentication. This fact also contributes to secure authentication. So cellular phones are used as a security token.

E. Objective

As described in the subsection dealing with OpenID in this section, the ID generation mechanism among an OP and RPs is not specified in an OpenID scheme and the simple answer is to generate random identifiers as handle IDs and record them bound to the user ID on an OP. However, this scheme has a problem. If the handle identifiers are persistent between an RP and an OP, the number of generated ID is limited. Therefore, it is not difficult for the OP to record them all. However, if these IDs are transient for each session even though they are used on a unique RP and OP pair in order to realize a strong anonymity for OpenID users, the number of IDs increases explosively and an enormous management cost is incurred.

III. Cryptographical ID Management

For the purpose described in previous section, we have introduced a cryptographic ID generation scheme on the OP site. By using this method, transient pseudonyms are generated from a user's unique identifier on the OP cryptographically. The binding between a user ID and the handle identifier is embedded in the handle ID itself as an encrypted form. Thus, the relationship between the user and handle identifier can be found by decrypting the handle identifier. Therefore, OPs do not have to keep all handle identifiers. Only keys for handle ID generation are stored safely at OPs. Therefore, this method contributes effectively to reducing management cost.

In our proposed cryptographic ID Management, user handle IDs are generated by means of Equation (1). Here, E_{key} means encryption with a key, and UID_{OP} means the user's ID on an OP that generates the handle ID, respectively. The value "$time$" is ID generated time and the value "$info$" is optional information for this ID generation. The mark "||" means concatenation.

$$ID = E_{key}(UID_{OP}||time||info) \tag{1}$$

The value "$time$" is used for ID type. If a user wants to use a transient ID for user identification on an RP, the ID generated time is used for this value. In this case, if the time interval is limited to a short period, the generated ID is a time-based unique ID because the value "$time$" is perfectly unique for each generated ID. In contrast, if a user wants to use a persistent value for the handle ID, a fixed value such as all zero is used for the value "$time$". In this case, the generated handle ID is always the same. Which type of ID to use depends on the policies of the user and RP. For this purpose, both users and RPs register their ID generation policy.

A. Analysis

If an IDP (OP) uses randomly generated IDs for transient pseudonyms, the number of IDs that the IDP has to keep is enormous because the IDP has to generate IDs for each user/services/sessions combination. On the other hand, by using the cryptographical ID generation method, the number of IDs managed in an IDP is greatly reduced compared to the random ID use. The relationship between a user ID on an IDP and handle ID is hidden in the generated ID itself. An IDP has to keep the encryption key for decryption. This means that the total amount of data is lessened, thereby contributing to a reduction in management cost at the IDP.

In the case of the cryptographical ID management method, the relationship among IDs is hidden in the ID itself and only the entity which knows the encryption key can decrypt generated IDs. Thus, user privacy is also protected. However, if the encryption key is leaked, the secret information could be revealed. As a measure to deal with this problem, we can use multiple keys. If different keys are used for each service, the harm is limited to a particular service. In our implementation of the prototype, therefore, dedicated keys are prepared for each RP.

IV. Implementation and Evaluation

In this section, we describe our implementation and the performance evaluation results.

A. Implementation of FAP

For the implementation, PCs (CPU: Core2Duo (E6400) 2.13GHz, memory: 1GByte) and a cellular phone are used as the FAP, OP, RP, user's PC and user's phone. For the OP and RP, a PHP based OpenID module is used. The module

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-105-2

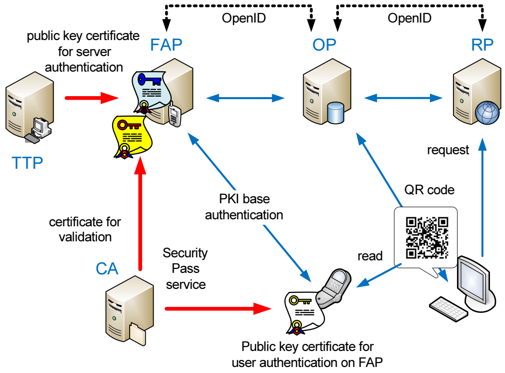

OpenID
OpenID
public key certificate for server authentication
FAP
OP
RP
TTP
request
QR code
PKI base authentication
certificate for validation
CA
Security Pass service
read
Public key certificate for user authentication on FAP

Figure 3. Implementaion overview

Table I
SYSTEM SEPCIFICATINS

Modules	
Platform	CentOS(linux 2.6.16)
OpenID module	php-opneid [13]
Crypto algorithm for ID generation	
encrypto alogorithm	AES (key length 128 bits)
Public key certificate specification on cellular phone	
Format	X.509 version 3 (RFC 2459)
Key algorithme	RSA (key length: 1024 bit)
Hash algorithm	SHA-1

is customized with a cryptographical ID generation scheme. The OP module is also customized for the communication between the OP and FAP. Between the OP and FAP, a customized OpenID module is used and ID federation can be executed between them so that they can securely exchange a handle pseudonym for their local IDs. For the authentication at the FAP, a PKI-based service (Security Pass [14]) is used. The cellular phone holds the public key certificate issued by the CA operated by a cellular phone company. The FAP also holds the public key certificate issued by a trusted third party for server authentication. Both certificates are used for mutual authentication and the construction of an SSL connection between them. In the construction of a secure connection, the FAP can extract the CN (common name) value from the certificate and use it as a user ID. Figure 3 is a schematic representation of the implementation and Table I shows the specifications for the implementation.

The flow sequence of our proposed federated authentication scheme is described below. Before this authentication flow, ID federation between an OP and FAP has been completed successfully and a handle pseudonym, which is fixed type handle ID is shared by both the OP and FAP.

1) A user accesses an RP.
2) The RP shows the login screen for the OpenID authentication mechanism.
3) The user inputs the OP identifier (in the case of OpenID authentication 2.0).
4) The RP identifies the OP from the OP identifier inputted at step (3) and redirects the user browser to the OP site.
5) The OP indicates the select screen for authentication methods (ordinary or federated authentication).
6) The user selects a method (federated authentication).
7) The OP shows the input screen for user OpenID.
8) The user inputs his or her own OpenID.
9) The OP generates an authentication delegation request to the FAP and redirects the user to FAP.
10) The user sends the authentication request to his or her cellular phone from the FAP screen.
11) The user's phone checks the URL and jumps to the FAP site on his cellular phone.
12) The FAP authenticates the user using a mobile ID and creates authentication results. Then the FAP sends the results to the OP.
13) The OP checks the results and the browser jumps to the RP site using redirection. The user can use the service on the RP.
14) The FAP sends the result to the mobile phone.

B. Implementation of ID generation

The figure shows how ID generation between OP and RP occurs. The User ID at an OP is of variable length. Therefore, we prepared a fixed length user ID bound to the user ID. In order to generate a handle ID, a fixed length ID is used. After authentication is finished using a mobile PKI service at the FAP, the authentication result is notified to the OP. At the same time, the user ID is also announced from the FAP. The ID is converted into a fixed length ID for ID generation at the OP. The procedure is described in detail below. The procedure is used for transient ID generation.

1) An OP looks up a fixed length ID (32 bits) for ID generation from the ID notified from an FAP.
2) Time value (32 bits) and reserved bits (16 bits) are added to the fixed length user ID. From the concatenated data, the CRC value is generated and also added to the data.
3) The total concatenated data is encrypted with the key for the RP which delegates user authentication to the OP.
4) The ID type identifier is added to the encrypted data. This identifier is used for ID type check at the OP and RP.
5) In addition, a service identifier is added to the data. The service identifier is assigned to each RP from the OP and used for searching for an encryption key for decrypting the handle ID at the OP.
6) The whole concatenated ID is converted to ASCII data using base 64 encoding.
7) Finally, the FQDN of the OP is added to the end of the converted data. This ID is used between the OP and RP.

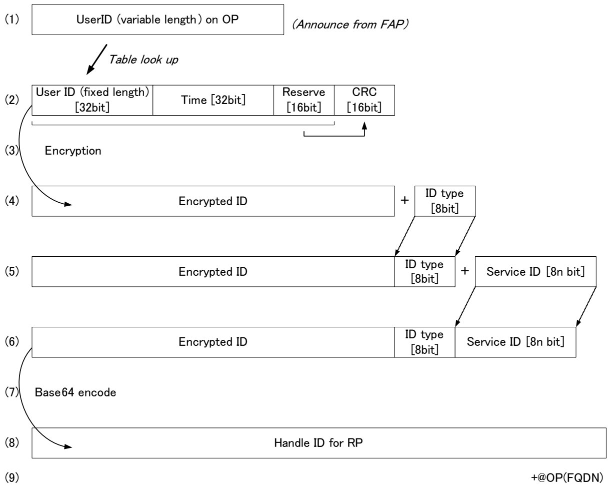


Figure 4. ID generation scheme

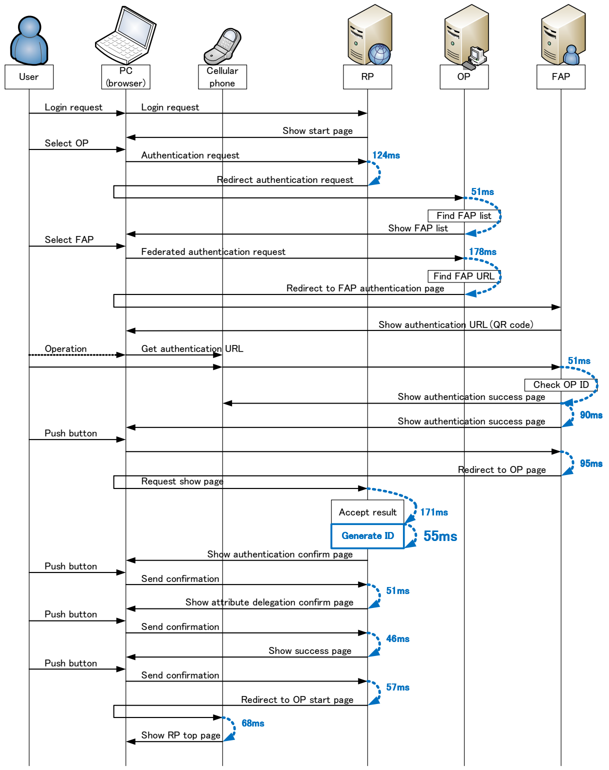


Figure 5. Federated Authentication Sequence

C. Evaluation

The time taken for user authentication on our proto-system is measured as an indication of performance. Figure 5 shows the sequence with user operation. However, the sequence for the federated authentication flow requires several manual operations such as using a camera for QR code, and pushing buttons on a cellular phone. Therefore, the total time for one operation could vary widely. Thus, we focused on the main operations on the RP, OP, and FAP. The measured time is also shown in Figure 5. The figures are the averaged value of fifteen trials. Mobile communication between the cellular phone and the FAP server depends on network conditions, and can therefore vary widely and uncontrollably. However, it was confirmed that the other operations, which do not need manual operation by users take a total of less than one second on the RP, OP, and FAP. The total time for one trial is a few seconds for ordinary cases.

In addition, the ID generation time is approximately fifty milliseconds as shown in Figure 5. This value is as fast as the table lookup case. So the total operation duration is feasible.

V. Conclusion

In this paper, we described our implementation for a PKI-based authentication scheme with cellular phone for an OpenID single sign-on scheme. For strong authentication, a public key certificate on a mobile phone is utilized; therefore, the mobile phone is used as a security token. In addition, an ID management technique based on a cryptographic technique is recommended in order to reduce management cost on the server side. Based on our proposal, a proto-system was implemented and performance evaluations were carried out. From the evaluation, each ID generation time is around 50 milliseconds. The results show that cryptographical ID generation contributes to reducing management cost in OP construction.

References

[1] OpenID, http://openid.net/ 16.08.2010.

[2] A. Pashalidis and C. J. Mitchell, “A Taxonomy of Single Sign-On Systems,” Proc. Information Security and Privacy, 8th Australasian Conference (ACIPS 2003), Springer-Verlag, 2003. vol. 2727/2003, pp. 249-264, DOI: 10.1007/3-540-45067-X_22.

[3] A. Pashalidis and C. J. Mitchell, “Impostor: A Single Sign-On System for Use from Untrusted Devices.,” Proc. IEEE Global Telecommunication Conference (GLOBECOM ’04), IEEE Press, Dec. 2004, vol. 4, pp. 2191-2195, doi:10.1109/GLOCOM.2004.1378398.

[4] T. Nishimura and H. Sato, “LESSO: Legacy Enabling SSO,” Proc. The 10th Annual International Symposium on Applications and the Internet (SAINT 2008), July 2008, pp. 301-304, doi:10.1109/SAINT.2008.76.

[5] R. Watanabe and T. Tanaka, “Federated Authentication Mechanism using Cellular Phone - Collaboration with OpenID”, Proc. 6th International Conference on Information Technology: New Generations (ITNG 2009), IEEE press, Apr. 2009, pp. 435-442, doi:10.1109/ITNG.2009.111

[6] Kantara Initiative,
http://kantarainitiative.org/ 16.08.2010.

[7] B. Pfitzmann, “Privacy in Enterprise Identity Federation - Policies for Liberty Single Signon -,” Proc. 3rd workshop on privacy enhancing technology (PET 2003), Springer-Verlag, 2003, vol. 2760/2003, pp.189-204, doi:10.1007/978-3-540-40956-4_13.

[8] OASIS SAML V2.0, http://www.oasis-open.org/specs/index.php#samlv2.0 16.08.2010.

[9] D. Recordon and B. Fitzpatrick, “OpenID Authentication 1.1,” OpenID foundation, May 2006, http://openid.net/specs/openid-authentication-1_1.html 16.08.2010.

[10] “OpenID Authentication 2.0 - Final,” OpenID foundation, Dec. 2007, http://openid.net/specs/openid-authentication-2_0.html 16.08.2010.

[11] “Electronic Authentication Guideline”, NIST, 2006, NIST special publication 800-63, ver. 1.0.2.

[12] “E-Authentication Guidance for Federal Agencies”, OMB, 2003, OMB M-04-04.

[13] janrain, http://www.janrain.com/openid-enabled 16.08.2010.

[14] Security Pass, KDDI, http://www.au.kddi.com/service/kino/securitypass/index.html (Japanese web site) 17.08.2010.

MBPR：A Business Process Repository Supporting Multi-Granularity Process Model Retrieval

Jiangjun Zhu, Hailong Sun, Zicheng Huang, XuDong Liu
School of Computer Science and Engineering
Beihang University
Beijing 100191 China
{zhujj, sunhl, huangzc, liuxd }@act.buaa.edu.cn

***Abstract*—Business process repository aims at business process model discovery and reusing. However, most of current approaches for process matchmaking have the limitation that the process models should be in the same granularity, wherease a process repository with good support for multi-granularity business process retrieval is still rare. In this paper, we propose a similarity measurement mechanism which can effectively calculate the similarity between business process models in different granularity levels. A case study is used to demonstrate how modelers can search multi-granularity business process. At last, we conduct extensive experiments based on real dataset to study the performance of the repository.**

Keywords-process similarity;multi-granularity; repository.

I. INTRODUCTION

The capabilities to easily find useful business process become increasingly critical for business process repository [1,2], as more and more business process accumulated with the evolvement of enterprises' and organizations' business procedure. Companies document their daily routines in the form of business process models. Business process helps companies understand, communicate upon, or reengineer working procedure to enhance competitiveness. Modeling the business processes of an enterprise is an essential part of any IT development or implementation process. However, model design consumes a considerable amount of time and requires determining of activities to be performed, ordering of their execution, handling exception cases that might occur, etc. [3]. Benefiting from the already developed process models in the resource repository of company, reusing seems to be a promising approach to reduce the time consumed to develop new models [3].

Current repositories [1,2] for process retrieval are based on the matchmaking between business processes. Metrics are mostly limited to the matching of their syntactic [4], semantic [5], structural [6] and behavioral [7] information. As surveyed by paper [8], due to the large number and different granularity levels of processes, business process models are most commonly described by a hierarchy [9], [10], [11]. The models to be compared can have different granularity levels for achieving the same functionality. For example, one business process has a single activity to achieve certain functionality, while in another business process the same behavior is achieved by composing several activities. But current repositories rarely consider this situation as their retrieval scope is generally limited in processes with the same granularity, which makes their retrieval ability not flexible enough in this context. Thus, new mechanism is required to expand the retrieval capability of process repository to meet the multi-granularity requirement.

In this paper, we propose Multi-granularity Business Process Repository (MBPR) supporting similarity measurement of business processes with different granularities, and the idea in this paper is to decompose the coarse-granularity business processes into fine-grained before similarity calculation. The main contribution of this paper is a novel approach for measuring similarity between multi-granularity business process models which include: business process similarity calculation algorithm adapted from graph matching algorithm, coarse activity decomposition mechanisms. The decomposition mechanism includes three parts: (i) a control flow segment and markup language (SCMT), which can segment and mark the control flow information of annotated information contained in coarse activity, (ii) a series of decomposition rules, (iii) and corresponding algorithm based on SCMT.

In the next Section, we indentify the problem of multi-granularity retrieval and characterize the solution roadmap of MBPR. Section 3 presents existing approaches for business process retrieval and shows their drawbacks. In Section 4, we propose a business process similarity calculation method based on graph matching algorithm. In Section 5, we propose the decomposition mechanism which includes (i) granularity classification definition, control flow segment, (ii) markup language and the decomposition rules, (iii) and algorithm. In Section 6, we present the implementation of MBPR and a case study. In Section 7, several experiments are conducted to study the performance of MBPR. Finally, section 8 presents ongoing work and conclusion.

II. RELATED WORK

This paper mainly relates to research of similarity between business processes, based on which business process repository support process model retrieval.

The topic of retrieving business process based on model similarities has gained a lot of attention recently. The approach to measure the similarity between two process models has been addressed from different perspectives. The syntactic metric calculate an optimal matching between the activities in the process models by comparing their labels based on string edit distance [4]. To exploit semantic features paper [5] have utilized wordNet synonyms. Structured data of business process have been taken into account, paper [6] discuss a structural match method based on classic graph matching algorithm, in which, business process are converted into directed graph, then edit distance between two directed graph is calculated to represent the similarity. Mendling, *et al*. [7] presented a preliminary discussion of influence of behavior in determining the business process similarity, it first convert business process models into causality graph footprint vectors, then compute the cosine value between two vectors to assess similarity. However, these approaches' context is that business processes to be compared are with same granularity, as we analyzed in Section 2, they are not flexible enough to handle the scenario in Fig. 1.

There is rare existing work on calculating similarity between processes on a degree of multi-granularity. Paper [18] proposed two new graph edit operations to take into account the difference of granularity levels, but it only consider the one-to-two relation. In this paper, the idea to solve this problem is decomposing the coarse activity process into fine-grained before the implementation of matchmaking. In Business Process Reengineering (BPR), refinement means explain exiting business process in more detail from multiple perspectives [19]. As shown in Fig. 2, the modeler use annotated information of activities to refine the coarse-granularity business process.

In summary, business process repository technology supporting retrieval based on business process matchmaking has been widespread concern, and has achieved some results. But most of the work focus on measuring similarity between business processes at same granularity level, there is lack of a multi-granularity matchmaking mechanism to expend retrieval capabilities.

III. OVERVIEW OF MBPR

In this section, we first present a scenario requiring multi-granularity similarity measurement. The scenario is situated in the context of developing of Enterprise-Specific Business Process Models. Then we present the multi-granularity retrieval procedure of MBPR.

Based on the prevalent granularity division, paper [8] further defined CBPM(Content-Based Process Modeling) which concentrate on developing of Enterprise-Specific Business Process Models with the help of existing business process, the procedure of this approaches can summarized as follows:

a) High level modeler (Enterprise analyzer or Manager) draw a business process at a coarse-granularity level to identify the general characteristics of the enterprise's business procedure.

b) Low level modeler (IT employees) need to refine the coarse-granularity business process into detailed fine-grained business process which comprehensive describe the enterprise's working procedure.

c) To reduce the developing time, low level modeler can retrieve those fine-grained business processes which are similar with the coarse-granularity one with help of repository search tool.

d) Fine tune the selected model to ensure that all relevant processes have been included, and unnecessary processes eliminated, and this is suggested business process caters the need of the implementing enterprise.

We can see the crucial step is c) and it implies a multi-granularity business retrieval scenario. Fig. 1 shows a overview of the multi-granularity retrieval scenario, take a coarse-granularity business process as an input, then measure similarity with all the business processes in repository, the most similar ones are recommended.

As shown in Fig. 1, the upper part is the input of coarse-granularity business process editor, including information of process structure, events, activities, etc. (Dotted P1 part). The lower part of Fig. 1 shows the available well-refined fine-grained business processes in repository.

However, the traditional retrieval approaches are not flexible enough for above scenario as they rarely take the granularity level information into account. A phenomenon is an coarse activity can map to a fragment of other business process according to their annotated information, as shown in Fig. 2, coarse activity "*hotel service*" can be decomposed into a BPMN [12] fragment which execute three tasks

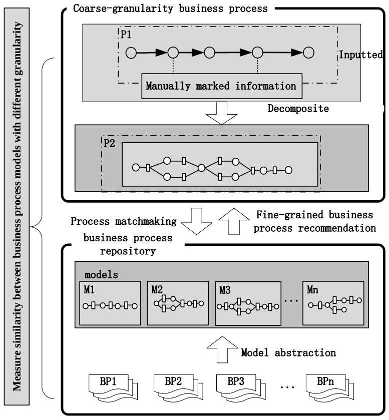

Measure similarity between business process models with different granularity
Coarse-granularity business process
P1
Inputted
Manually marked information
Decomposite
P2
Process matchmaking
Fine-grained business process recommendation
business process repository
models
M1
M2
M3
Mn
Model abstraction
BP1
BP2
BP3
BPn

Figure 1. Scenarios of multi-granularity business process retrieval

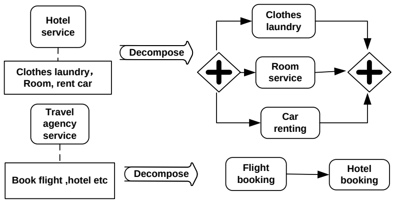


Figure 2. Example of coarse activities decomposition

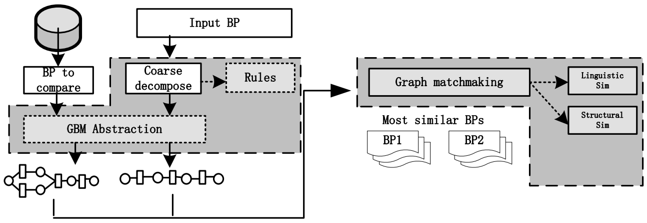


Figure 3. Multi-granularity retrieval procedure of MBPR

simultaneously include: “*clothes laundry*”, “*room service*” and “*car renting*”, but traditional similarity measuring approaches usually do not handle this situation, for processes which describe a same business logic in different granularity level, the similarity between them calculated by traditional approaches is low, so the retrieval result can not be satisfactory.

In order to effectively measure similarity between business processes with different granularity, we decompose the coarse activities of business process into fine-grained before similarity calculation. So in Fig. 1, before matchmaking, we add a step that executes the decomposition operation according to manually marked information (Dotted P2 part). The multi-granularity business process retrieval procedure of MBPR is shown in Fig. 3, business process models are abstracted to memory models and pushed into database server of MBPR, coarse-granularity process is decomposed before matchmaking, after similarity measurement, the most similar ones are recommended. Next, we will elaborate on the multi-granularity similarity measurement mechanism of MBPR.

IV. Similarity measurements

In this section, we adapt an graph matchmaking algorithm to calculate the similarity between business process models modeled with BPMN.

A. *Process model abstraction*

Currently, BPMN has become one of the mainly used modeling languages during business process development. In this paper, we use BPMN as business process modeling language. To simplify the problem, we only discuss a core subset of BPMN. We refer to the definition of core BPMN proposed by [13], which is a subset of BPMN specification includes the key elements used to describe process control flow, core BPMN can meet most of the business process modeling needs.

As BPMI (Business Process Management Initiative) does not provide strict theoretical standard, BPMN so far still has no definite execution semantics, leading to that business processes developed by different BPMN modeling tools have large semantic difference in the structure and behavior, which lead to the complexity of refinement of BPMN process model. We refer to the notion of WCBP (Well-structured core BPMN process) proposed also by [13]. WCBP add some constraints based on Core BPMN: Firstly, two parallel flows initiated by a parallel fork gateway, should be joined by a parallel join gateway. Secondly, two alternative flows created via a decision gateway should be synchronized by a merge gateway.

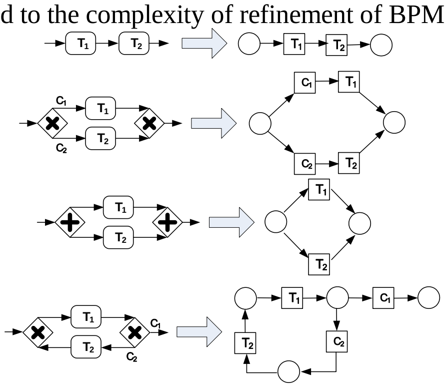


Figure 4. Four process and their GBM

First, we abstract a process model into a directed attributed graph, as following definition.

Definition 4.1 GBM(Graph based Business process Model)

Let Z be a set of labels. A GBM is a tuple (N,E,λ,Z), in which:

—N is the set of nodes;

—$E \subseteq N \times N$ is the set of edges;

—$\lambda: N \rightarrow Z$ is a function that maps nodes to labels

When abstracting a BPMN described model, we drop the types of nodes, Fig. 4 shows the GBM of a business process model. The left part shows the original process models. The right column shows the corresponding process graphs after abstraction. Each node is named with the node label, which originate from activity name or branch condition, gateways and events are abstracted into empty nodes

B. *Node similarity metric*

The similarity of nodes is determined by the similarity of node labels.

To exploit linguistic features we have utilized WordNet [14] as a background ontology, for each activity(node) in GBM we extract its name as main information [15] to measure its similarity with other activity, most activities' name are not formulated but most are presented as a phrase such as “*Credit Review*” or “*book ticket*”, we can use a vector $\vec{v}$ < *verb* , *noun*> extract from a activities' name to describe the activity.

With the definition of $\vec{v}$, we improve the linguistic metric from [15]. The original linguistic function used by [15] is *L(n1,n2)*, it can calculate unorganized similarity value between two node' labels based on wordNet. Since all the nodes' label have been abstracted into $\vec{v}$ < *verb, noun*>, we propose function *sim*（$\vec{a}$, $\vec{b}$ ）to compute the similarity between two node, each label is donated by vector $\vec{a}$, $\vec{b}$. W is a semantic similarity matrix, W_{ij} represents the similarity value between the pair a_i and b_j, which can obtain using wordNet APIs.

$$sim(\vec{a},\vec{b}) = \frac{\vec{a} W \overrightarrow{b^T}}{|a||b|} \quad (1)$$

The two vectors may contains different number of words，word pair with the same syntax will be chosen to calculate similarity.

C. *Adapt graph matching algorithm for process similarity*

We use exiting graph matchmaking algorithm to measure similarity between GBMs. Paper [16] overview four main graph matching algorithms based on graph edit distance, but all of them are time consuming, to simplify the matching complexity, we use path index based algorithm *GraphGrep*, which is proposed by [17]. There are three basic steps of *GraphGrep*: (1) build the index to represent the database of graphs as sets of paths, this step is done by repository system previously, (2)filter the database based on the submitted query and the index to reduce the search space, and (3) perform exact match. The algorithm details can refer to the paper [17] .

Synthesize with GraphGrep and node similarity metric, let *G1 = (N1;E1; λ1)* and *G2 =(N2;E2; λ2)* be two GBM, the total Graph similarity are measured as follows:

$$SynSim(G_1, G_2) = \frac{\sum_{i=0}^{\|PH_1\|} \max\left\{Sim(P_i, P_j) \mid P_j \in PH(G_2)\right\}}{\|PH_1\|}, \qquad (2)$$

$$PH(G_2) \text{ donate set of paths of } G_2$$

$$Sim(P_1, P_2) = \frac{GrapGrepSim(P_1, P_2) * \sum_{i=0}^{\|N_1\|} \max\left\{Sim(\vec{a}_i, \vec{a}_j) \mid a_j \in N(P_2)\right\}}{\|N_1\|},$$

$$N(P_2) \text{ donate } \textit{nodes} \text{ of } G_2$$

Equation (2) is a comprehensive similarity metric which takes accounts of the linguistic, structural information. As we discussed in Section 2, the granularity information must be taken into account when measuring the similarity, (2) do not fulfill this request, in the next section, a coarse activity decomposition mechanism will be proposed to handle this situation.

V. COARSE ACTIVITY DECOMPOSITION MECHANISM

In this section, we propose the decomposition mechanism for coarse-granularity business process.

A. *Basic definition*

For a coarse activity, its activity names may be a brief description of several tasks. To handle this situation, usual practice is attaching annotated information to the activity to explain the details. For example, As shown in Fig. 2, an activity "*Hotel service*" has annotated information which indicate that the single activity need to perform several tasks in a certain order, including: "*clothes laundry*", "*room service*", "*rent car*", etc. According to the annotated information, it can be decomposed into a fragment. As displayed in Section 2, the inputted coarse-granularity process is decomposed by MBPR before implementing matchmaking. In this paper, we take annotated information as basis to decompose coarse granularity activities.

Previous discussion prompts us to assess the granularity level of business process from the perspective of annotated information. Several basic definitions are given as follows:

Definition 5.1 coarse activity

An activity is coarse if and only if it has annotated information. We say an activity is decomposable if it is coarse.

Definition 5.2 fine-grained business process

A business process is fine-grained if and only if all its activities are not coarse.

Definition 5.3 coarse-granularity business process

A business process is coarse-granularity if it is not fine-grained. We say a business process is refinable if it is coarse-granularity.

B. *Control flow markup Tags and decomposition rules*

Based on the definition of granularity level, the idea of the decomposition is： Decompose all the coarse activities of coarse-granularity process until it become fine-grained. So the decomposition of a coarse activity is basic, which will replace the activity with a process fragment.

The procedure of the decomposition of a coarse activity can be divided into four steps: firstly, activity names are extracted from the annotated information as a set for new activities generation; secondly, the logic relations between new activities is analyzed, corresponding to WCBP as described in last section, our main concern is the Sequence, Switch, Loop and Parallel control flow relations; thirdly, new activities, arcs and gateways are created according to the activity names and relations; finally, the coarse activity is replaced with the generated fragment.

It is hard to extract the logic relations and activity names of annotated information attached to coarse activity by understanding natural language. To make it feasible, a small quantity of manually marks need to be added to annotated information, so a small markup language is introduced in the following segment.

Back to the Section 2, consider this situations, when a low-level modeler get a coarse-granularity business process, he mark and adjust the annotated information with some special tags complying with certain scheme, which MBPR can understand, then accuracy of the retrieval will be improved. To do this, reference to the traditional programming language design principles, we design the Control flow segment and markup Tags (SCMT). Using SCMT, a low-level modeler can mark the crucial logic relation and activity names in the annotated information to affiliate retrieval.

Existing business process modeling language usually supports the four basic control flow patterns: sequential (sequence), select (switch), concurrent (parallel) and loop, corresponding to the four relation presented by BPMN, we design four group of tags as shown in table 1: *&THEN* mark sequence relation, &*SIMU* mark parallel relation, *&IF* and &*ELSE* mark switch relation, &*WHILE* and &*REPEAT* mark Loop relation. In table 1, *A*,*B* and *C* can be a complete sentence or part of it. Through the segmentation tag, each of them can be divided into several activity names： {*A0, A1......An*}, {*B0,B1……Bm*}, {*C0,C1…… Ck*} .

The complete definition are defined by BNF in definition 5.4, we only concern the control flow representation. To improve the user experience, the compiler of MBPR does not have strictly syntax restrictions on the user marked annotated information.

Definition 5.4. **Control flow segment and markup Tags(SCMT)**

<DecomposableInformation>::=<Clause>{ <Clause> }
<Clause> ::= < Sequence >|< Parallel >|< Switch > |< Loop >
< Sequence > ::= <Element> **&THEN** <Element>
<Parallel>::=&**SIMU**<Element><Element>{<Element>}
<Switch>::=&**IF**<Element>&**ELSE**<Element>{&**ELSE** <Element>}
<Loop>::=&**WHILE**<Element>|&**REPEAT** <Element>
<Element>::=<Segments>|<Clause>{<Segments>|<Clause> }
<Segments>::=<ActivityName>{**∗**<ActivityName>}
<ActivityName>::=human readable phrase

Fig. 5 shows decomposition rules of these basic control flow patterns from SCMT to BPMN. BPMN use a number of tasks to describe the sequence of behavior, as shown in Fig. 5 (a), the switch pattern is decomposed to a BPMN fragment triggered by an exclusive gateway, determine the appropriate conditions and started a number of optional activities set, each of which corresponds to a specific chosen branch, as shown in Fig. 5 (b), the parallel pattern is decomposed to a BPMN fragment triggered by a an parallel gateway, start a number of concurrent collection of activities, shown in Fig. 5 (c), Similar to the mapping of switch pattern, the loop pattern is decomposed to a BPMN fragment which contains two optional branches, as shown in Fig. 5 (d).

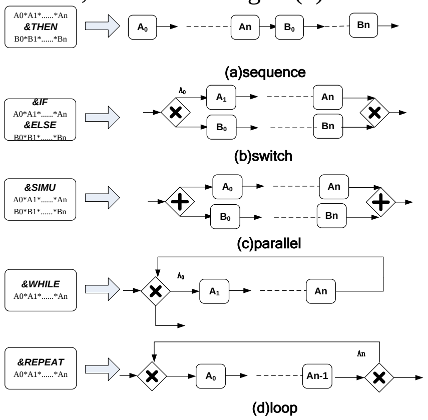


Figure 5. Decomposition rules between SCMT patterns to BPMN fragments

TABLE I. BASIC MARKS OF SCMT

Relation	Mark	Pattern
Sequence	&***THEN***	A &***THEN*** B
Parallel	&***SIMU***	&***SIMU*** A B C
		&***SIMU*** A B
Switch	&***IF*** &***ELSE***	&***IF*** A &***ELSE*** B &***ELSE*** C
		&***IF*** A &***ELSE*** B
Loop	&***WHILE***	&***WHILE*** A
	&***REPEAT***	&***REPEAT*** A
	∗(Segmentation tag)	***A0∗A1∗......∗An***

C. Decomposition algorithm

According to the decomposition mechanism, we design an coarse activity decomposition algorithm, which can understand the annotated information marked by SCMT. As shown in algorithm 1, the SCMT marked information is compiled and the coarse activity is decomposed into business process fragment.

Described as algorithm 1, activity names are extracted from SCMT marked information according to segmentation tag, when encounter the key elements marks the logic relations, choose appropriate function to handle it with the assistant stack and generated the correspondent BPMN fragment.

Algorithm 1 decomposition of coarse-granularity activity

```
Input：coarse-granularity BPMN business process
Output: fine-grained BPMN business process
begin
A：= {a| activity set};
A*：={a| a∈A & a has annotated information};
      stack =Φ; //Stack of activities
for each a∈A* {
      In(a);//activities attached to a and prior to a
      Out(a);//activities attached to a and succeed to a
      S=a.getAnnotatedinformation()；
      for each e∈ S {
         if (e is a ActivityName)
               New activity and push to stack；
         else if(e is a tag of Sequence)
            HandleSequence();//handle sequence relation
         else if(e is a tag of Switch)
            HandleSwitch ();//Switch relation
         else if(e is a tag of  Parallel)
            HandleParallel();//Parallel relation
         else if(e is a tag of Loop)
            HandleLoop ();//Loop relation
      }
      attach In(a) to the beginning of fragment；
      attach Out(a) to the end of fragment；
end
HandleSwitch(){//other functions are similar
      a1 = stack.pop;
      a2 = stack.pop;
      new DatabasedExclusiveGateway g；
      attach g，a1； //create switch branch
      attach g，a2；
}
```

VI. CASE STUDY AND EXPERIMENTS

In this section, we implemented MBPR with the architecture presented by Fig. 7 and Fig. 8 shows corresponding perspective of MBPR.

The case study in Fig. 6 displays the inner process of loan applications which are modeled using BPMN. Part(a) describe the application procedure in coarse-granularity with annotated information to explain its details, part(c) is a fine-grained business process exits in database server of MBPR, suppose that part(a) is query input and part(c) is the target of retrieval, it is obvious that in terms of structure, syntax or behavior, they do not have a strong similarity. If we measure the similarity between them in no consideration of granularity information, the return quantitative similarity will be very small, which means the retrieval result may not contain the target, but in fact they describe the same application procedure.

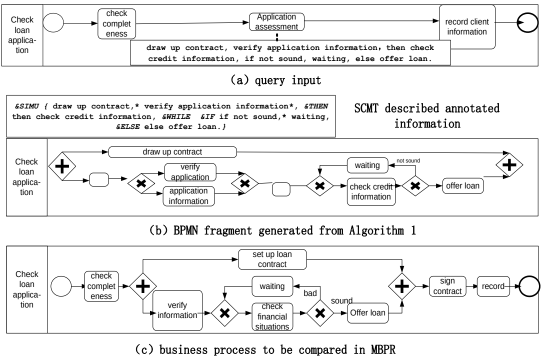
In the context of MBPR, using SCMT, the annotated information of activity “*application assessment*” in (a) can be marked and segmented as follows：

&SIMU { draw up contract ∗ verify application information ∗ ***&THEN*** then check credit information, ***&WHILE &IF*** if not sound ∗ waiting ***&ELSE*** else offer loan***}***

Then SCMT compiler of MBPR which implement Algorithm 1 will decompose coarse activity “*application assessment*” into the fragment described in part(b), after replace “*application assessment*” with this fragment, we can obviously see that, no matter from structure, execution path, or connection relations of these processes, the fine-grained business process are more similar to the target process, so it is more promising that the retrieval result contains the target part(c).

Next，we design and conduct experiments to study the performance of MBPR.

The multi-granularity retrieval procedure of MBPR includes two operations: decompose the coarse-granularity business process，calculate similarity using equation (2). The time complexity of equation (2) is O (| *V1* | x | *V2* | x | *E1* | x | *E2* |), where *V1* and *V2* are the number of nodes in *GBM*, *E1* and *E2* are the number of paths in *GBM*. For a coarse activity, the time complexity of decomposition is O(*S*), where S is the number of activity names contained in annotated information. If the coarse-granularity business process contains *n* coarse activities, the overall algorithm complexity is polynomials O(|*S*| x |*n*|+| *V1* | x | *V2* | x | *E1* | x | *E2* |).


Check loan applica- tion
check complet eness
Application assessment
record client information
draw up contract, verify application information, then check credit information, if not sound, waiting, else offer loan.

(a) query input

&SIMU { draw up contract,* verify application information*, &THEN then check credit information, &WHILE &IF if not sound,* waiting, &ELSE else offer loan.}
SCMT described annotated information
Check loan applica- tion
draw up contract
verify application
application information
waiting
not sound
check credit information
offer loan

(b) BPMN fragment generated from Algorithm 1

Check loan applica- tion
check complet eness
set up loan contract
waiting
bad
verify information
check financial situations
sound
Offer loan
sign contract
record

(c) business process to be compared in MBPR

Figure 6. Case study

We assess the effectiveness and efficiency of MBPR. We use 120 bioinformatics process workflows as experimental data set collected from biological research community www.myexperiment.org, which provides a platform for biotechnology researchers worldwide to publish, share and test their processes. In order to maintain consistency with the premise and the context of this paper, we first remodel the collected bio-computing processes using BPMN and post them to the database server of MBPR, each function of a bio-computing process is considered to be an activity in BPMN, and its note is attached to the correspond activity as annotated information, which we marked them with SCMT. The experiments were conducted on a Windows machine with a 2GHz Pentium IV CPU and 2G main memory.

The change of business process versions demonstrate the refinement procedure, so the original versions represent coarse-granularity business process and the latest ones represent fine-grained, experiment 1 simulate the scenario of Fig. 1 by comparing the similarity between the original version and latest version of a same business process. Fig. 9 shows the static results of these records. As we seen, making use of decomposition mechanism before matchmaking, the average similarity is 0.60 while without decomposition is 0.21. This means decomposition mechanism is promising to affiliate the MBPR retrieve more accurate result.

In the second experiment, we consider traditional precision and recall measures that have been extensively used in information retrieval. We randomly extract 10 business processes and set their original versions as “search query models”, then set their other versions as “relevant models”. We take “search query models” as the input of MBPR, Fig. 10 shows the average precision and recall scores across all the queries. We can see that if querying without decomposition, the precision become very low(only 0.2) when recall rate is equal to 0.55, but if add the decomposition step, precision dramatically decrease until recall rate exceed 0.8. This graph shows that on average, with the decomposition mechanism, MBPR can effectively affiliate multi-granularity retrieval. Using CPU time as metrics, experiment 3 evaluated the relation between retrieval time and the number of coarse activity contained in input business process, as shown in Figure 11；the result was that it is polynomial.

VII. CONCLUSION

In this paper, we proposed MBPR supporting multi-granularity business process retrieval based on a novel method for multi-granularity business process similarity measurement. The contributions of this paper include：A process matchmaking algorithm, Control flow segment and markup Tags (SCMT) is designed; A series of decomposition rules are proposed to refine coarse-granularity business process using annotated information; The effectiveness of MBPR is evaluated based on real data set.

Currently, MBPR has been used to query business process in different granularity. However, the coarse activity decomposition mechanism is highly dependent on the annotated information inputted by modeler. Our future work

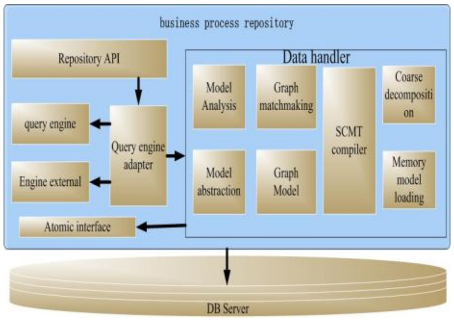
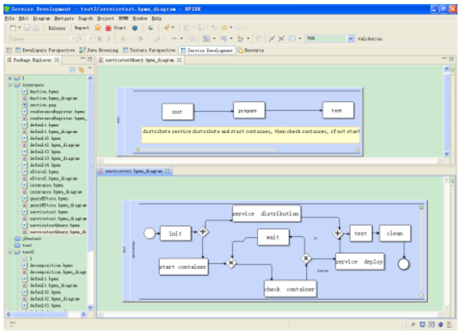


Figure 7. Architecture of MBPR



distribute service distribute and start container, then check container, if not start

Figure 8. Perspective of MBPR

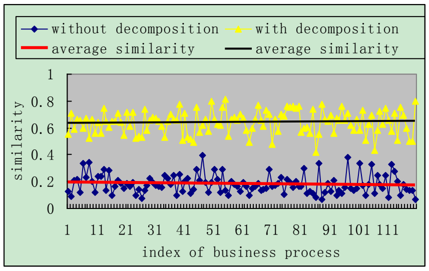


Figure 9. Effectiveness comparison

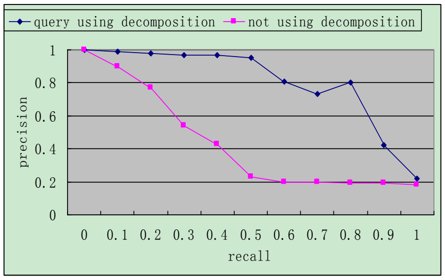


Figure 10. Precision-recall curve

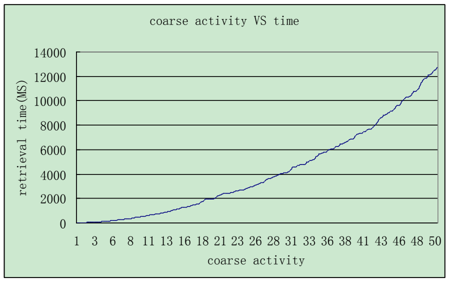


Figure 11. Coarse activity vs time

will be further enhancing the accuracy of decomposition by considering the more information about activity such as Interaction protocol information.

ACKNOWLEDGMENT

This work was partly supported by China 863 program under Grant No. 2007AA010301 and partly supported by China 973 program under Grant No. 2005CB321803.

REFERENCES

[1] K. Shahzad, B. Andersson, M. Bergholtz, A. Edirisuriya, T. Ilayperuma, P. Jayaweera, and P. Johannesson. Elicitation of Requirements for a Business Process Model Repository. BPM 2008 Workshops, pp. 42-53. Springer Heidelberg(2008).

[2] Ma, Z., Wetzstein, B. Anicic, D. Heymans, S. and Leymann, F.: Semantic Business Process Repository. In: Proceedings of the Workshop on Semantic Business Process and Produce Lifecycle Management, Innsbruck (2007)

[3] A. Awad, A. Polyvyanyy, and M. Weske. Semantic querying of business process models. In EDOC, pp. 85-94. IEEE Computer Society, 2008.

[4] B. F. van Dongen, R. M. Dijkman, and J. Mendling, "Measuring similarity between business process models," in Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE), ser. LNCS, vol. 5074. Springer, 2008, pp. 450–464.

[5] M. Ehrig, A. Koschmider, and A. Oberweis, "Measuring similarity between semantic business process models," in APCCM '07: Proceedings of the fourth Asia-Pacific conference on Comceptual modelling. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 71–80.

[6] R. Dijkman, M. Dumas, and L. Garc´ıa-Ba˜nuelos. Graph matching algorithms for business process model similarity search. In Proc. of BPM 2009, Ulm, Germany, September 2009.

[7] J. Mendling, B. Dongen, and W. Aalst. On the Degree of Behavioral Similarity between Business Process Models. In CEUR-Workshop, pp. 39–58, 2007.

[8] A. Wasser, M. Lincoln, and R. Karni: Accelerated Enterprise Process Modeling Through a Formalized Functional Typology. BPM 2005: LNCS 3649 pp. 446-451.

[9] Oracle Business Models (OBM) http://download.oracle.com/docs/cd/B53825_04/current/acrobat/121cacug.pdf 2009 2010-6-22

[10] SAP Business Map. http://help.sap.com/ 2010 2010-6-22

[11] Phios Process Repository for Supply Chain Management. http://repository.phios.com/SCOR/Default.asp 2010 2010-6-22

[12] White, S.A. (2008-01-17) Business Process Modeling Notation,V1.1. http://www.omg.org/spec/BPMN/1.1/ 2010-6-22

[13] Zicheng H, Jinpeng H, Hailong S, Xudong L, and Xiang L, "BestRec: A Behavior Similarity Based Approach to Services Recommendation," services, pp.46-53, 2009 Congress on Services - I, 2009

[14] G. Miller, "Wordnet: A lexical database for english," Commun.ACM, vol. 38, no. 11, pp. 39–41, 1995.

[15] T. Pedersen, S. Patwardhan, and J. Michelizzi, WordNet: Similarity–measuring the relatedness of concepts. In Proc. AAAI, pp. 1024-1025, 2004.

[16] Bunke, H. (2000), Graph matching: Theoretical foundations, algorithms, and applications, in 'Proc. Vision Interface 2000, Montreal', pp. 82-88.

[17] D. Shasha , J. Wang , and R. Giugno, Algorithmics and applications of tree and graph searching, Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, June 03-05, 2002, Madison, Wisconsin

[18] P. Bradley, J. Browne, S. Jackson and H. Jagdev, "Business process reengineering (BPR): A study of the software toolscurrently available", Computers in Industry 25(3) (1995) pp.309-330.

[19] D. Grigori, J.C. Corrales, and M. Bouzeghoub. Behavioral matchmaking for service retrieval: Applicationto conversation protocols. Inf. Syst., 33(7-8): pp.681–698, 2008.

Critical Information Infrastructures Management System and Security Issues

Focusing on the Public Administrative Sector

Jun Heo
Internet Service Protection Team
Korea Internet & Security Agency
Seoul, Korea
herjune@kisa.or.kr

Wan Suk Yi
Internet Service Protection Team
Korea Internet & Security Agency
Seoul, Korea
wsyi@kisa.or.kr

Abstract— **With Korea's transition to an advanced IT nation, its key infrastructures, including administration, transportation, healthcare, finance and communications, have become absolutely reliant on IT systems. This research examines systems in place in Korea for defense against cyber hacking and electronic invasions, and internal and external threats to the critical information infrastructure of the administrative sector, in which critical administrative service is provided to Koreans through e-government, and technical, physical and managerial measures to combat such threats are discussed.**

Keyword - Security; threats; administrative sector.

I. INTRODUCTION

The IT society is absolutely reliant on communications infrastructure. Social infrastructures, including administration, transportation, healthcare and education, rely on the communications infrastructure to function. A breakdown of the communications infrastructure due to destruction or malfunction will result in other social infrastructures malfunctioning. Thus it can be said that the information infrastructure is the core infrastructure of an IT society.

The Korean government implemented the revised Communications Infrastructure Protection Act in 2001 to systematically protect critical communications infrastructures in the finance, communications and energy sectors. To combat the increasing losses occurring from a rapid advancement of techniques used in electronic invasion activities, including the distribution of malicious programs, the Communications Infrastructure Protection Act stipulates preventative measures, countermeasures and recovery measures for systematic designation and protection of critical information infrastructures [1].

The communications infrastructures are designated as critical information infrastructures in accordance with 5 standards stipulated by the Communications Infrastructure Protection Act. Once designated as a critical information infrastructure, a risk assessment must be performed within 6 months. Subsequently, risks assessments are carried out once every 2 years. Based on the outcome of the risk assessment, the manager of a critical information infrastructure must implement short or long-term measures as required. Such measures are incorporated in the protection policies for the following year to achieve effective protection of critical information infrastructures. Also, protection policies are reviewed in the following year to verify that various protective measures have been taken as planned. Through such a process, critical information infrastructures are systematically protected [2].

As the social infrastructures in Korea, such as administration, transportation, healthcare and finance, are absolutely reliant on IT systems, efforts are being made to continuously expand the communications infrastructure. Since 2009, additional critical information infrastructures have been designated in the administrative sector.

A wide range of civil services are being provided in the administrative sector with ongoing e-government support. As the administrative sector is closely linked to the administrative bodies of cities, counties and districts throughout Korea, security breach in the administrative sector is highly likely to escalate to a national scale.

This research examines critical information infrastructure policies and their management systems. Also, threats to public service systems in the administrative sector being designated as critical information infrastructures, such as the e-government, and their protective measures are examined to provide reliable data for utilization in public administration.

II. CRITICAL INFORMATION INFRASTRUCTURE POLICIES AND MANAGEMENT SYSTEM

Communications infrastructures are managed under the jurisdiction of central administrative agencies. In order to provide systematic and comprehensive governmental protection of critical information infrastructures from electronic invasions, the Communications Infrastructure Protection Act stipulates the operation of the Communications Infrastructure Protection Committee to oversee the formation and execution of communications infrastructure protection policies to achieve cooperation on prevention and management of security breaches by various central administrative agencies.

Duties of the Communications Infrastructure Protection Committee include mediation in critical information infrastructure protection policies; review of formation and execution of protection plans; and review of systemic improvements and policies related to the protection of critical information infrastructures.

The Communications Infrastructure Protection Activities Committee was established to support the operation of the Communications Infrastructure Protection Committee by reviewing matters presented to the Communications Infrastructure Protection Committee or as directed by the chairman of the Communications Infrastructure Protection Committee.

In the event of a serious security breach in a critical information infrastructure, the Communications Infrastructure Protection Activities Committee operates a temporary security breach management headquarters under the supervision of the Communications Infrastructure Protection Committee to implement recovery measures and provide technical support.

The central administrative agencies responsible for the management of critical information infrastructures designate critical information infrastructures in their field of administration and implement protective measures. Protection guidelines are established by the central administrative agencies for critical information infrastructure managers to follow.

Critical information infrastructure managers hold the primary responsibility for protection and must assess vulnerabilities of facilities under their jurisdiction to implement protective measures. In the event of a breakdown of a critical information infrastructure due to disturbance or destruction, the relevant organizations must be notified and measures required for the recovery and protection of the communications infrastructure must be taken [3].

Organizations that support the protection of critical information infrastructures include the Ministry of Public Administration and Safety, National Intelligence Service, Ministry of National Defense, Public Prosecutor's Office, National Police Agency and Korea Internet Security Agency. These organizations also establish critical information infrastructure protective measures and provide technical support for prevention and management of security breaches.

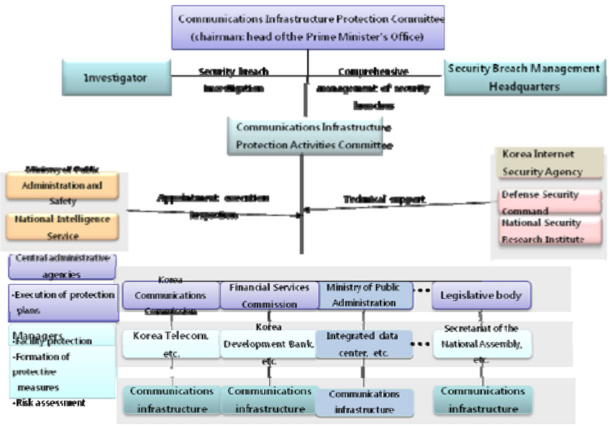

Communications Infrastructure Protection Committee
(chairman: head of the Prime Minister's Office)
Investigator
Security breach investigation
Comprehensive management of security breaches
Security Breach Management Headquarters
Communications Infrastructure Protection Activities Committee
Ministry of Public Administration and Safety
National Intelligence Service
Appointment execution inspection
Technical support
Korea Internet Security Agency
Defense Security Command
National Security Research Institute
Central administrative agencies
-Execution of protection plans
Managers
-Facility protection
-Formation of protective measures
-Risk assessment
Korea Communications Commission
Financial Services Commission
Ministry of Public Administration
Legislative body
Korea Telecom, etc.
Korea Development Bank, etc.
Integrated data center, etc.
Secretariat of the National Assembly, etc.
Communications infrastructure
Communications infrastructure
Communications infrastructure
Communications infrastructure

Figure 1. Critical Information Infrastructure Management System

III. SECURITY THREATS TO CRITICAL INFORMATION INFRASTRUCTURES IN THE ADMINISTRATIVE SECTOR

A. *Attacks on Public Administrative Service Websites*

Websites that service the public through critical information infrastructures of the administrative sector, such as the websites of local governments, are currently understood to be relatively safe from external infiltrations in their servers. However, there are hazardous elements found on other websites operated by local governments, which can jeopardize the security of such websites if such elements are analyzed and taken advantage of maliciously. In general, the vulnerabilities of websites with menus that contain sub-domains are cookie poisoning and cross website script vulnerabilities, which can be manipulated to gain access to some personal information of registered website users. File download vulnerabilities and factor modification can be incorporated into attacks to steal web application files and database access history and change posts made by other users.

Taking the online notice board for example, after session authentication, application factor modification can be used to read, modify or delete non-public posts made by other users; file download vulnerabilities can be taken advantage of to expose application source files; or insufficient script tag restriction settings can be taken advantage of to steal other users' session IDs to access their personal information. Because file extension verification for upload of image files on websites with inadequate security is insufficient, restricted extensions (ASPX, ASP) can be uploaded to generate certain files within the system. Such files then enable the upload of malicious programs, which are then used for remote execution of malicious commands to access databases, administrator accounts and personal information.

B. *Attacks on Internal Administrative Networks*

Development and management of critical information infrastructures of the administrative sector are frequently performed by a civilian business. In such a case, the critical information infrastructures may be insufficiently managed due to various reasons, such as the civilian business' heavy workload or budget shortage. This can result in vulnerabilities in servers, networks or PCs that can be taken advantage of by a third party to execute a malicious attack. Some common internal system vulnerabilities are as follows.

Server vulnerabilities can occur due to issues in account management, system/directory security settings, registry security settings and unnecessary services. Network vulnerabilities can occur due to issues in remote access control, log storage settings, anti-DDoS settings and unnecessary services. PCs can be vulnerable to unauthorized data access due to unnecessary services and non-use of a screen saver.

By taking advantage of such vulnerabilities, even closed networks with no external connection can be infiltrated using social engineering techniques and USB worms to steal information, which is then taken away from the premises on a USB or through a third party and disclosed on the internet, setting the network from which the information was stolen a target for attack.

Due to the nature of critical information infrastructures, system vulnerabilities must be removed by meticulously analyzing system stability and connection to other systems. Such security improvement may require substantial time and resources, so ongoing monitoring is required.

IV. PROTECTION OF CRITICAL INFORMATION INFRASTRUCTURE IN THE ADMINISTRATIVE SECTOR

For improved IT security in the operational environments of e-governments and local governments, various protective measures, such as the application of information protection processes, facility security upgrade and server/network security upgrade, can be implemented to achieve a consolidated information protection and reliability of the communications infrastructures. In particular, information protection for critical information infrastructures in the administrative sector is divided into technical, physical and managerial aspects and executed accordingly.

TABLE I. PROTECTION OF CRITICAL INFORMATION INFRASTRUCTURE OF THE ADMINISTRATIVE SECTOR

Security management		Details		
Technical	**Website**	Mock hacking	Performed on priority websites	
	Internal administrative network	Application system	Server network, etc.	o Security control settings, service access control, security system operation o Inspection of security policies, user authentication, etc.
		Security system	IDS, IPS Firewall, etc.	
		Personal user system	PC, etc.	
Physical	**Information protection assessment**	Physical aspect	Data room access control, power-supply control	o Physical access control, restricted area setting, installation of physical security facilities o Designation of a physical security facility manager, inspection of backup files o Power-supply control, etc.
Managerial		o Utilization of the KISA Infrastructure Information Protection Assessment Chart, KISA Infrastructure Protection Activities Inspection Chart and other methods o Information protection assessment based on local governments' activities and conditions		

A. Technical Protection

Based on an analysis of various security breaches that can affect the stable operation of critical information infrastructures and the confidentiality, integrity and availability of information stored therein, technical vulnerabilities of the infrastructures must be identified to gain an understanding of the consequences of a security breach and form preventative measures. As such, the servers, network equipment, security systems and PCs that are part of critical information infrastructures must be analyzed and a security breach test performed to analyze the ensuing web service stability and possible scenarios of external attacks. [4]

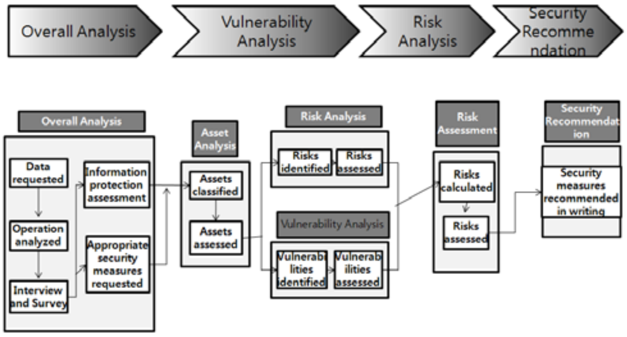


Figure 2. Vulnerability Assessment Methodology (KISA)

In order to remove the vulnerabilities identified through the analysis, short (6 months) and long-term (1 year) plans must immediately be established to continuously remove vulnerabilities. The technical measures to be taken to remove such vulnerabilities are as follows.

1) *Prevention of security breaches and loss of key information through consolidation of server/network security settings*

a) Consolidated account management, including consolidated authentication management and regular account policy reviews, for prevention of database access by unauthorized parties and loss of information

b) Consolidated access control, including consolidated remote access control and regular access control policy reviews, for prevention of access to key systems by unauthorized parties *c)* Consolidated operation management, including regular security patch application on servers/PCS and real-time anti-virus updates, for prevention of security breaches

2) Consolidated infrastructure security management, including the application of upgraded information protection solution, for prevention of exposure to potential threats

a) Implementation of integrated log management solutions that manage and monitor logs generated between different devices for systematic identification and analysis of threats

b) Implementation of anti-DDoS solutions and upgrade of existing information protection solutions for minimization of vulnerabilities in networking facilities used to service the public

B. Managerial/Physical Protection

Following a technical analysis of devices used in critical information infrastructures, the overall state of information protection by an infrastructure manager is analyzed to form customized information protection measures and plan their quantitative implementation. Information protection assessment methodologies established this way are presented to Korea Internet Security Agency annually. An information protection assessment assesses 12 areas of security control, which contain 89 sub-areas of security control. Each sub-area of security control is assessed according to the standards shown in Table II. The maturity of an area of security control is determined according to the outcome of an assessment of its sub-areas of security control.

TABLE II. VARIOUS LEVELS OF MATURITY OF INFORMATION PROTECTION ASSESSMENT

Level	Details of assessment
1	Sub-areas of security control are not being executed or are being executed without a set plan
2	An execution plan (procedures, schedules, budget) for some sub-areas of security control is set and documented
3	Sub-areas of security control are being executed according to a documented plan by the entire organization or have been executed
4	Assessment of sub-areas of security control continues to take place for a set period of time
5	Assessment of sub-areas of security control is completed and regular improvements are made according to the outcome of the assessment

Table III below shows the areas and sub-areas of security control assessed in an information protection assessment.

TABLE III. AREAS OF INFORMATION PROTECTION ASSESSMENT

Area	Sub-area
1. Information protection policies	1.1 Information protection policy organization
	1.2 Information protection plan
2. Risk assessment	2.1 Asset classification
	2.2 Asset allocation/management
	2.3 Security requirement review
	2.4 Risk assessment
	2.5 Vulnerability assessment
3.Structural management	3.1 Security structure modification control
4. Maintenance	4.1 Maintenance tools
	4.2 Remote maintenance
5. Media protection	5.1 Media display
	5.2 Media access management
	5.3 Media viewing/transportation methods
	5.4 Document management
	5.5 Media/record disposal
6. Security awareness/training	6.1 Security awareness/training
	6.2 Emergency training
7. Work continuity management	7.1 Work continuity management
	7.2 Information system backup/recovery
8. Physical/environmental protection	8.1 Physical access control/monitoring
	8.2 Electricity/communication cable protection
	8.3 Emergency electricity/lighting
	8.4 Environmental control
9. Personnel security	9.1 Personnel management
	9.2 Internal personnel management
	9.3 Third party security
10. Accident management	10.1 Accident management drill training
	10.2 Accident monitoring
	10.3 Security breach processing
11. Audit and attribution of responsibility	11.1 Auditable event generation
	11.2 Audit information management
	11.3 Audit monitoring/analysis/report
	11.4 Denial prevention
	11.5 Attribution of responsibility
12. System access control and communications protection	12.1 Account/password management
	12.2 Setting management
	12.3 Access control
	12.4 Failed access attempt management
	12.5 Notification of warnings during system operation
	12.6 Software faults/protection from malicious codes
	12.7 Service denial protection
	12.8 Confidentiality management

V. CONCLUSION

Korea is an advanced IT society and the majority of its key social infrastructures rely on communications infrastructures. Electronic invasions as well as natural disasters can result in the destruction and breakdown of such communications infrastructures, which can ultimately lead to the destruction or breakdown of social infrastructures that rely on the communications infrastructures.

The Korean government has established the Communications Infrastructure Protection Act which stipulates the designation of the communications infrastructures recognized as requiring protection from electronic invasions as critical information infrastructures. In the administrative sector, technical, managerial and physical measures must be taken to achieve ongoing security as attacks on websites of the administrative sector or internal administrative networks can stop the provision of administrative service to the public.

The number of designated critical information infrastructures is not high in comparison to the level of key social infrastructures' reliance on communications infrastructures. As such, an increase in designated communications infrastructures is in the best interest of the nation.

Therefore, critical information infrastructures must be continuously designated and accompanied by an efficient protection system in order to eliminate national and social vulnerabilities to acts of electronic invasion.

REFERENCES

[1] National Assembly: Critical Information Communication Infrastructure Protection Act (www.law.go.kr)

[2] Ministry of Public Administration & Safety, Korea Internet Security Agency: *Communications Infrastructure Protection Guide*.2009

[3] J. Chul-ki *"A Research on Cyber Security Threats to Critical information infrastructures and Their Countermeasures: With a Focus on the Broadcasting/Communications Sector."* vol. 9, pp. 34-36,　August 2009

[4] L. Bodin, L. Gordon, and M. Loeb, "Information security and risk management,Communications of the ACM", vol. 51(4), pp. 64‐68, 2008.

A Synergistic System of Institutional Repository and Researcher Database

Kensuke Baba
Research and Development Division
Kyushu University Library
Fukuoka, 812-8581, Japan
Email: baba@lib.kyushu-u.ac.jp

Masao Mori
Institutional Research Office
Kyushu University
Fukuoka, 812-8581, Japan
Email: mori@ir.kyushu-u.ac.jp

Eisuke Ito
Research Institute for IT
Kyushu University
Fukuoka, 812-8581, Japan
Email: itou@cc.kyushu-u.ac.jp

***Abstract*—The paper introduces a practical Web system which activates institutional repositories. Institutional repository is an important service of libraries in academic institutions. The authors of the paper developed a linking system between the institutional repository and the researcher database of their university. By the developed system papers registered in the institutional repository are linked from lists of papers in the researcher database, which improves the accessibility of the institutional repository. Additionally, the system reuses the metadata of the papers registered in the researcher database for registrations to the institutional repository, which reduces the efforts of researchers. As a result, this system is expected to encourage the registration of papers to the institutional repository. The paper describes the concepts and the details of the system. The essential idea can be applied to other academic institutions.**

***Keywords*-institutional repository; Web database; Open Access; scholarly paper; library**

I. INTRODUCTION

The subscriptions for electronic journals occupy a large proportion of the budget of university libraries, which forces researchers to make funds for the price or restrict the number of journals to read. This is a quite unacceptable situation for academic societies. A solution of the problem is the idea of "Open Access [3][4]", which is to open free-access information to the public. Publishing of scholarly papers on an institutional repository (IR) is one of activities based on the idea. An increase of the papers stored in IRs is expected to solve the problem of journal prices indirectly.

IR has been receiving increasing attention; actually 129 universities in Japan have their own IR (Aug. 12, 2010). However, the number of the papers in the IRs is small compared with the number of the papers actually written by the researchers in the universities. One of the obstacles to increase the number of papers is the fact that researchers are forced some efforts when they register their papers (and usually this operation is repeated when they upload the papers on their web-site, submit a list of the papers as a report to their institute, and so on!). Additionally, it is difficult to prove the effectiveness of IR, hence researchers do not think that registering papers to an IR is worth their time.

The aim of our research is to increase the number of the papers in IRs. Then, we set the following two purposes:

- To increase the number of accesses of papers on IRs,
- To reduce the efforts for registering papers.

The first purpose is to make clear an effectiveness of IRs. The number of citations to the registered papers from other papers is a measure of the effectiveness of an IR, and the number of the accesses of a paper can be considered as a rough upper-bound of the number of citations to the paper. The Ranking Web of World Repositories [11], which is provided by the Spanish Cybernetics Lab. in the Consejo Superior de Investigationes Científicas (CSIC), takes account of the "visibility" in addition to the number of the contents. The visibility is the total number of links from external sites. The number of accesses of papers is expected to be improved by links from a reasonable list such as a database or a result of a search. As to the second purpose, the practical efforts of registration cannot be zero, therefore it seems to be effective that the information of a registration is shared and reused with other systems.

For the previous two purposes, we developed a linking system between the IR and the database of researcher activities of Kyushu University and are operating the system from Apr, 2010. For the first purpose, by the linking system, some papers in the IR are linked from a list in the researcher database, which improves the accessibility of the IR. For the second purpose, the information of the registered papers are shared between the IR and the researcher database by the linking system. In Kyushu University, any researcher has a duty to input his/her research and education results into the researcher database, therefore some efforts of a researcher can be reduced. It is not special for our university that researchers are required to register their research activity to a researcher database and an IR. For example, the National Academic Research and Collaborations Information System (NARCIS) [2] in the Netherlands has been developed in 2009 in order to run a central search engine of academic information with linking function to IR. NARCIS can be the one-stop national service by collecting all kind of lump academic information in the Netherlands. Our linking system help researchers to register their articles accurately in researcher database and IR.

This paper introduces the linking system we developed. First, the basic information of the IR and the researcher database are described, which makes the problems clear. Next, the two subsystems of the linking system are explained. The subsystems correspond to the previous two purposes, respectively. By the developed system the accessibility of the IR is improved and the effort of registration to the IR is reduced, which yields an increase of the papers stored in the IR. The essential idea of our system can be applied to other institutions which have their IR and researcher database.

The rest of this paper is constructed as follows. Section II describes the basic information and the problems in the IR and the researcher database in Kyushu University. Section III explains the concepts and the concrete functions of the developed system. Section IV concludes this paper and introduces our future work.

II. Problems

This section describes the basic information of QIR (Kyu(Q)shu University Institutional Repository) [10] and DHJS (Academic Staff Educational and Research Activities Database in Kyushu University, “Daigaku Hyoka Joho System” in Japanese) [8], and then makes clear the problems we tackle.

A. QIR

QIR is an IR based on DSpace [9] and operated by the Kyushu University Library since Apr, 2006. The total number of the contents in QIR is 13,948 (Mar. 31, 2010), which is extremely large compared with the IRs of the other Universities in Japan. However, in the Ranking Web of World Repositories [11], QIR is ranked at the 88th position (Jul, 2010). Table I shows the number of the contents in QIR. The largest content is “Departmental Bulletin Paper” and its ratio is about 75%. Most of QIR contents are original (that is, unpublished). Therefore, links on Web are expected to be an effective factor to increase the number of accesses to contents.

Generally IR stores the full-text of a paper in addition to the metadata of the paper such as the title, the authors, the name of the proceedings, and so on. Figure 1 is a web image of QIR. The page is the profile page of a researcher and the list is the result of a search of the name in the author fields. The third column is the title of each paper and linked to the site of detailed information of the paper with the full-text. The rightmost column is the number of the accesses to each paper. In addition to the search of author, it is possible to search by general keywords in the fields of the title, the abstract, and so on.

For researchers (at least in Kyushu University), one of the obstacles to register their papers to IR is the fact that the effectiveness of the registration is not clear. Watson [7] and O’Leary [5] verified the validity of electronic journals by analyzing the relation between download history of papers and the citations in the papers. However, as to IRs in Japanese universities, no significant correlation is found as the relation [6]. The reason is considered that the number of the registered papers and the number of the accesses to the papers are not enough as samples to estimate some properties statistically.

Table I
The number of the contents in QIR (Mar. 31, 2010).

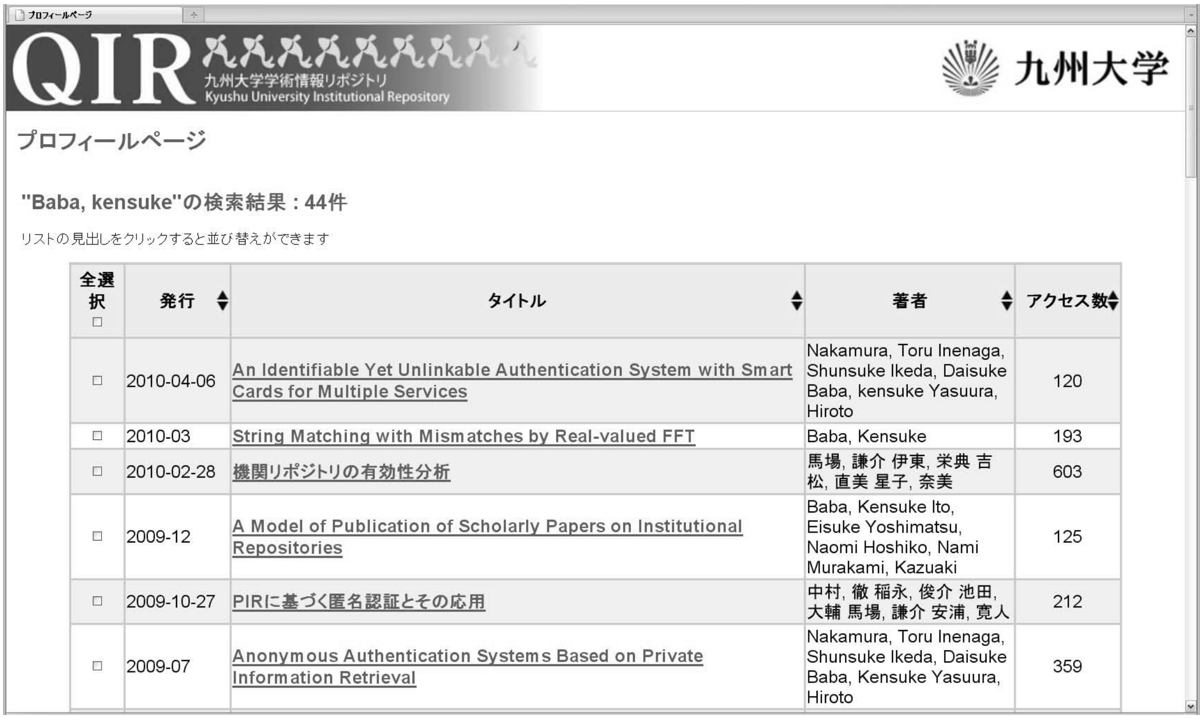
Type	Number of Contents
Journal Article	1,142
Thesis or Dissertation	95
Departmental Bulletin Paper	10,443
Conference Paper	731
Presentation	137
Book, Chapter	95
Technical Report	395
Research Paper	105
Article	164
Preprint	146
Learning Material	32
Others	463
Total	13,948

B. DHJS

Kyushu University developed a researcher database in 2005, and “DHJS” is the abbreviated name in Japanese of the database. DHJS has various kinds of data of any researcher in Kyushu University, for example, posts, research interests, and scholarly papers. The number of the researchers in Kyushu University is 2,197 at Oct. 1, 2009. DHJS consists of the two subsystems, the data-entry system and the viewer system. The data-entry system is developed from scratch by Kyushu University. The viewer system uses a commercial high-speed engine on XML as the backend search system. The main technology of the engine was developed in Kyushu University.

The data-entry system supports researchers to register their research activities to DHJS. In Kyushu University, any researcher has to register their research interests, research activities includes the metadata of scholarly papers, and so on. The viewer system shows the research activities registered in DHJS by the data-entry system. The data is separated with respect to each researcher and the registered metadata of scholarly papers of a researcher are described as a list.

Any researcher in Kyushu University has a duty to input his/her research and education results into DHJS. Therefore, DHJS has the list of (most of) the paper titles that were produced in Kyushu University in recent years. The number of the unique paper titles in DHJS is over 70,000. However, QIR has only 13,948 contents as mentioned in the previous subsection, that is, more than 56,000 papers are not uploaded


プロフィールページ
QIR
九州大学学術情報リポジトリ
Kyushu University Institutional Repository
九州大学

プロフィールページ

"Baba, kensuke"の検索結果：44件

リストの見出しをクリックすると並び替えができます

全選択 □	発行	タイトル	著者	アクセス数
□	2010-04-06	An Identifiable Yet Unlinkable Authentication System with Smart Cards for Multiple Services	Nakamura, Toru Inenaga, Shunsuke Ikeda, Daisuke Baba, kensuke Yasuura, Hiroto	120
□	2010-03	String Matching with Mismatches by Real-valued FFT	Baba, Kensuke	193
□	2010-02-28	機関リポジトリの有効性分析	馬場, 謙介 伊東, 栄典 吉松, 直美 星子, 奈美	603
□	2009-12	A Model of Publication of Scholarly Papers on Institutional Repositories	Baba, Kensuke Ito, Eisuke Yoshimatsu, Naomi Hoshiko, Nami Murakami, Kazuaki	125
□	2009-10-27	PIRに基づく匿名認証とその応用	中村, 徹 稲永, 俊介 池田, 大輔 馬場, 謙介 安浦, 寛人	212
□	2009-07	Anonymous Authentication Systems Based on Private Information Retrieval	Nakamura, Toru Inenaga, Shunsuke Ikeda, Daisuke Baba, Kensuke Yasuura, Hiroto	359

Figure 1. The web image of a list of papers stored in QIR. This example is the result of a search of "Kensuke Baba" in the author fields.

to QIR. Therefore, there is yet room for improvement of the number of the contents in QIR.

III. Synergy of QIR and DHJS

We developed a linking system between QIR and DHJS, and are operating the system from Apr, 2010. This section explains the concepts of the system and concrete functions of the system in the two points, that is, the connections with the data-entry system and the viewer system in DHJS.

A. Overview of System

We implemented two concepts to encourage researchers to register their papers to QIR. One is a support for registration to QIR by reducing the efforts of the input of the metadata of papers. The other is an improvement of the accessibility of the papers in QIR by linking from lists of papers in DHJS.

Figure 2 is the outline of the systems. For the former concept, researchers can register their paper to QIR by submitting the full-text instead of the metadata and the full-text. Additionally, researchers can know whether each paper is stored in QIR. As to the latter concept, users can search the full-text of a paper at the same time they search a researcher in DHJS in addition to search in QIR. This function should increase the number of the accesses to papers in QIR, which encourages researchers to register their paper.

B. Connection with Data-entry System

The linking system realizes two functions on the data-entry system. Figure 3 is a web image of the interface of

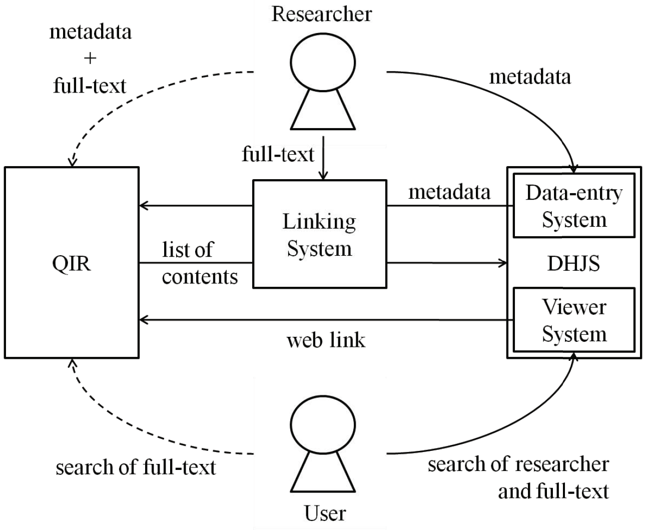
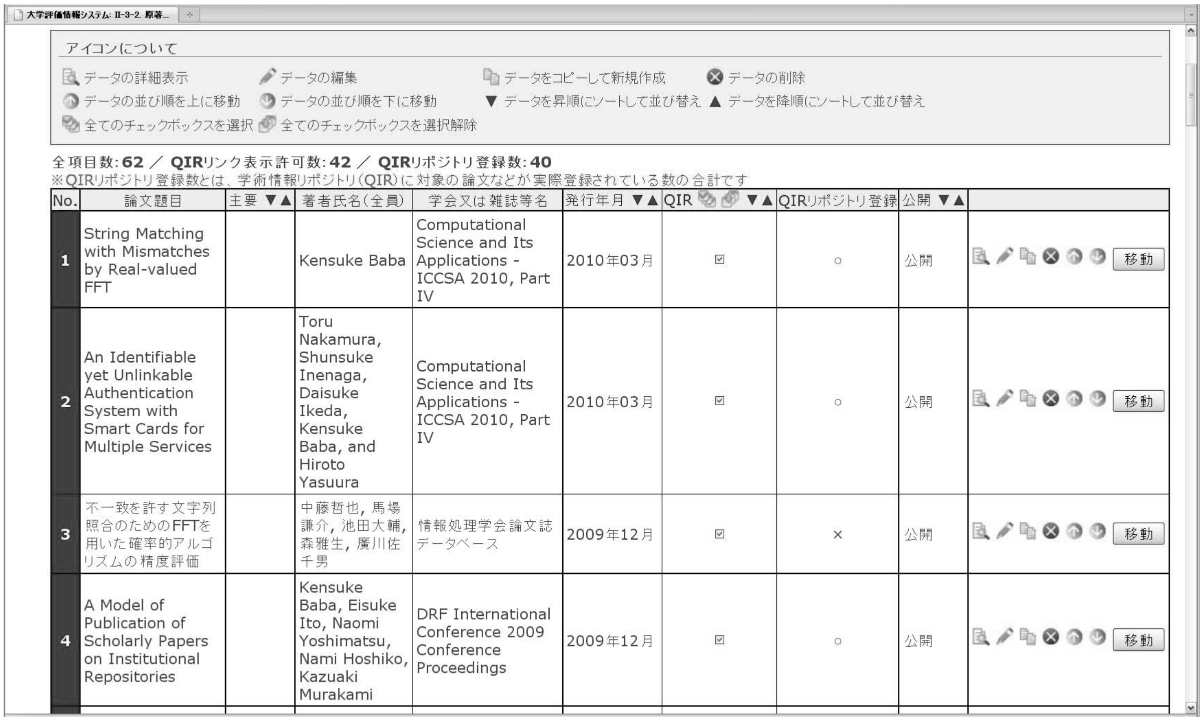


Figure 2. The outline of QIR, DHJS, and the linking system.

the data-entry system, especially for registration of scholarly papers.

Firstly, by a link to the registration form of QIR, researchers can register their papers on the data-entry system. Additionally, when a researcher wants to register a paper, the metadata of the paper is used to fill the registration form. In Figure 3, one of the icons in the rightmost row is the link to



アイコンについて

データの詳細表示　データの編集　データをコピーして新規作成　データの削除
データの並び順を上に移動　データの並び順を下に移動　▼ データを昇順にソートして並び替え　▲ データを降順にソートして並び替え
全てのチェックボックスを選択　全てのチェックボックスを選択解除

全項目数: **62** ／ **QIR**リンク表示許可数: **42** ／ **QIR**リポジトリ登録数: **40**
※QIRリポジトリ登録数とは、学術情報リポジトリ(QIR)に対象の論文などが実際登録されている数の合計です

No.	論文題目	主要 ▼▲	著者氏名(全員)	学会又は雑誌等名	発行年月 ▼▲	QIR ▼▲	QIRリポジトリ登録	公開 ▼▲	
1	String Matching with Mismatches by Real-valued FFT		Kensuke Baba	Computational Science and Its Applications - ICCSA 2010, Part IV	2010年03月	☑	○	公開	移動
2	An Identifiable yet Unlinkable Authentication System with Smart Cards for Multiple Services		Toru Nakamura, Shunsuke Inenaga, Daisuke Ikeda, Kensuke Baba, and Hiroto Yasuura	Computational Science and Its Applications - ICCSA 2010, Part IV	2010年03月	☑	○	公開	移動
3	不一致を許す文字列照合のためのFFTを用いた確率的アルゴリズムの精度評価		中藤哲也, 馬場謙介, 池田大輔, 森雅生, 廣川佐千男	情報処理学会論文誌データベース	2009年12月	☑	×	公開	移動
4	A Model of Publication of Scholarly Papers on Institutional Repositories		Kensuke Baba, Eisuke Ito, Naomi Yoshimatsu, Nami Hoshiko, Kazuaki Murakami	DRF International Conference 2009 Conference Proceedings	2009年12月	☑	○	公開	移動

Figure 3. The data-entry system of DHJS connected with QIR. This example is the interface for registration of (the metadata of) scholarly papers for the user "Kensuke Baba".

the registration form of QIR. The interface will be improved by Nov, 2010 so that users can upload a full-text of their paper directly in DHJS.

Secondly, the linking system obtains from QIR

- Whether the papers of the current user (researcher) are registered to QIR, respectively and
- The number of the registered papers of the user in QIR

and describes them on the data-entry system. In Figure 3, the column named "QIR" means whether the user want to link to QIR for each paper, and the next column means whether the paper is stored in QIR ("○" if stored). The three numbers (with some Japanese words) on the table are, respectively, the number of the papers registered in DHJS, the number of the papers checked at the "QIR" column, and the number of the papers stored in QIR. This function is expected to encourage researchers to register by showing the current state of their registration.

C. Connection with Viewer System

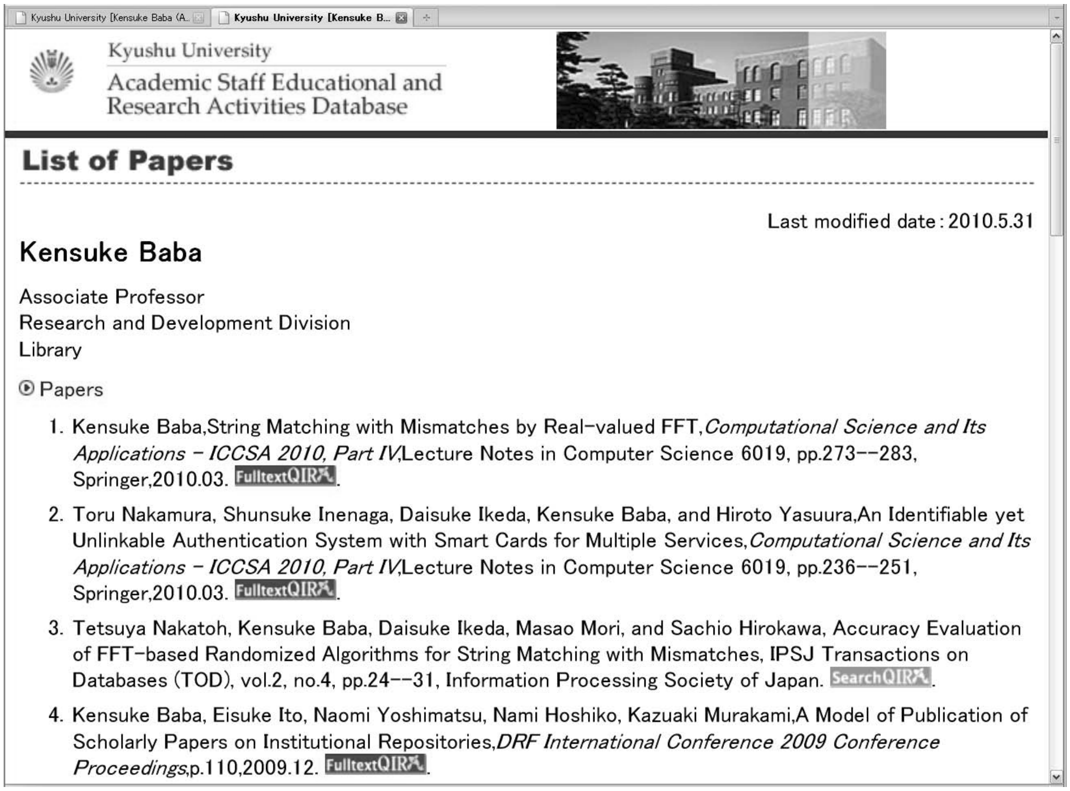
The papers which checked on the "QIR" column at the data-entry system are linked to the full-text of the corresponding paper in QIR by icons on the viewer system. There exist two kinds of icons which distinguishes whether the paper is registered in QIR. Figure 4 is the list of the papers registered in DHJS. For each paper in the list, a dark-colored icon "fulltextQIR" is added, a light-colored icon "searchQIR" is added, or there is no icon. The first case means that there is the full-text of the paper in QIR, the second that there is no full-text (although the researcher wants to register), and the other that the researcher does not want to link. In the second case, the linking system returns the result of the search by the author name in QIR.

The numbers indexing the papers in the table in Figure 4 are the numbers in Figure 3. (The paper 3 is written in Japanese.) You can see that the situation of the "QIR" column and the next column in the data-entry system is reflected to the icons in the viewer system.

IV. Conclusion and Future Work

In this paper, the linking system between the IR and the researcher database in Kyushu University were introduced. The system provides links from the metadata of papers in the researcher database to the corresponding full-text in the IR. Additionally, the system reuses the metadata of the papers in the researcher database for registrations to the IR. By the previous functions, the linking system is expected to enhance the registration of papers to the IR. The concepts of the developed system were explained generally, which leads the idea of the system to be applicable to other academic institutions.

It is our future work to analyze the number of the accesses to the papers in the IR. We already analyzed the access log of the IR from Apr, 2009 to Mar, 2010 [1], and are taking the access log from Apr, 2010. Also we are going to verify


Kyushu University [Kensuke Baba (A...
Kyushu University [Kensuke B...

Kyushu University
Academic Staff Educational and Research Activities Database

List of Papers

Last modified date : 2010.5.31

Kensuke Baba

Associate Professor
Research and Development Division
Library

Papers

1. Kensuke Baba,String Matching with Mismatches by Real-valued FFT,*Computational Science and Its Applications - ICCSA 2010, Part IV*,Lecture Notes in Computer Science 6019, pp.273--283, Springer,2010.03. FulltextQIR.
2. Toru Nakamura, Shunsuke Inenaga, Daisuke Ikeda, Kensuke Baba, and Hiroto Yasuura,An Identifiable yet Unlinkable Authentication System with Smart Cards for Multiple Services,*Computational Science and Its Applications - ICCSA 2010, Part IV*,Lecture Notes in Computer Science 6019, pp.236--251, Springer,2010.03. FulltextQIR.
3. Tetsuya Nakatoh, Kensuke Baba, Daisuke Ikeda, Masao Mori, and Sachio Hirokawa, Accuracy Evaluation of FFT-based Randomized Algorithms for String Matching with Mismatches, IPSJ Transactions on Databases (TOD), vol.2, no.4, pp.24--31, Information Processing Society of Japan. SearchQIR.
4. Kensuke Baba, Eisuke Ito, Naomi Yoshimatsu, Nami Hoshiko, Kazuaki Murakami,A Model of Publication of Scholarly Papers on Institutional Repositories,*DRF International Conference 2009 Conference Proceedings*,p.110,2009.12. FulltextQIR.

Figure 4. The viewer system of DHJS connected with QIR. This example is the list of the papers of "Kensuke Baba" in DHJS.

an effectiveness of the developed system by analyzing the number of the registrations and the ratio of the registrations by the system. Additionally, we are planning to develop an "embargo" system in the IR, that is, a system to manage the exhibition of the papers on the basis of the copyrights and the policies of publishers. The embargo system can make more efficient the operations for registrations to IRs.

ACKNOWLEDGMENTS

We thank the staff of the Kyushu University Library, Naomi Yoshimatsu, Nami Hoshiko, and Eriko Kudo, for their comments about QIR and general institutional repositories. We also thank the anonymous referees for their helpful comments to improve this work.

REFERENCES

[1] Baba, K., Ito, E., Yoshimatsu, N., and Hoshiko, N.: "An Analysis of Validity of Institutional Repository" (in Japanese), Proc. The Second Forum on Data Engineering and Information Management (DEIM 2010), F7-3, 2010.

[2] Dijk, E., Baars, C., Hogenaar, A., and van Meel, M.: "NARCIS: The Gateway to Dutch Scientific Information", Proc. ELPUB2006 Conference on Electronic Publishing, 2006.

[3] Harnad, S.: "The Green Road to Open Access: A Leveraged Transition", The Culture of Periodicals from the Perspective of the Electronic Age, L'Harmattan, pp. 99–105, 2007.

[4] Harnad, S., Brody, T., Vallieres, F., Carr, L., Hitchcock, S., Gingras, Y, Oppenheim, C., Stamerjohanns, H., and Hilf, E.: "The Access/Impact Problem and the Green and Gold Roads to Open Access", Serials Review 30, 2004.

[5] O'Leary, D. E.: "The relationship between citations and number of downloads in Decision Support Systems", Decision Support Systems, vol. 45, no. 4, pp. 972–980, 2008.

[6] Sato, S., Tomimoto, H., and Itsumura, H.: "The Relationship between Citations and Number of Downloads in Institutional Repositories" (in Japanese), http://www.tulips.tsukuba.ac.jp/dspace/handle/2241/104229 (Aug. 12, 2010).

[7] Watson, B. A.: "Comparing citations and downloads for individual articles", A Journal of scientific research on biological vision, vol. 9, no. 4, pp. 1–4, 2009.

[8] DHJS: Academic Staff Educational and Research Activities Database ("Daigaku Hyoka Joho System" in Japanese) in Kyushu University, http://hyoka.ofc.kyushu-u.ac.jp/search/index_e.html (Aug. 12, 2010).

[9] DSpace, http://www.dspace.org/ (Aug. 12, 2010).

[10] QIR: Kyu(Q)shu University Institutional Repository, https://qir.kyushu-u.ac.jp/dspace/ (Aug. 12, 2010).

[11] Ranking Web of World Repositories, http://repositories.webometrics.info/ (Aug. 12, 2010).

Estimation of Telecommunication Technologies, Services and Costs to Support Public Transport Information System Requirements

Tapio Väärämäki (tapiov@jyu.fi)[1], Timo Korhonen (timo.korhonen@tkk.fi)[2], Vesa Riihimäki (vesa.riihimaki@tkk.fi), Reijo Salminen (reijo.salminen@m-oas.fi), and Arto Karila (arto.karila@karila.com)
[1]University of Jyväskylä, Jyväskylä, Finland [2] Aalto University, Helsinki, Finland

***Abstract*—Unified vision of Helsinki Metropolitan Area Council (YTV) aims systematically develop ITS services and associated networking technologies to follow the very latest global ITS trends. In this study, Intelligent Transport Systems (ITS) telecommunication technology, service and economical scenarios for YTV area are inspected up to 2014. Our goal set by YTV is to research and validate YTV suggested ITS services and to investigate viable telecommunication networking alternatives in an evolution timeline. At the moment, most ITS services of YTV do not require broadband connection or real-time operation, as traffic light priority and RFID-based ticking. On the other hand, there are timely broadband networking needs as supplying Internet connection to passengers. Generally, ITS service data rates are increasing and real-time operation is getting important as indicated by efforts of car manufactures to develop WLAN solutions. Also, integrated real-time ITS operation and management systems require high data rates. Generally, Quality of Service (QoS) requirements of ITS services form a base in our evaluation. Our results, compressed to technology-service scenarios, indicate that YTV has the following paths for successful ITS development: (1) They can buy the networking services/technologies from a telecommunication operator, (2) They can build own network or (3) They can realize a hybrid solution. Our results in telecommunication networking alternatives are compressed into investment sensitivity estimations that can be used to support decision making.**

Keywords – Intelligent Transport System, Networking services, Networking technologies, Investment estimation, Networking Scenarios

I. Overview

In this study, ITS telecommunication scenarios for Helsinki Municipality area are inspected up to 2014. Our study analyzes various networking technology scenarios intended to support existing and planned ITS services for busses and trams. Also, we comment relating business aspects and competition environment.

Generally, QoS requirements of user services form a base in this evaluation. Typical technical QoS parameters include data rate, delay, error rate, packet loss and coverage. Service associated performance qualifiers come from pricing models, purchasing, operation and maintenance as well as service arrangements and user interface design and operation.

Currently, city of Helsinki applies Radio Frequency Identification (RFID) – based ticketing solution, traffic light priority switching for some of the most important crossing and the city has tested broadband communications in busses using Flash-OFDM technology [1]. Internet, mobile phone and real-time displays at busses, trams and commuter train stops, terminals and other central locations facilitate convenient travelling and assist in travelling planning. Tickets can be paid also by travel cards or purchased even by mobile phones. Unified vision of YTV is to systematically develop ITS services and associated networking technologies to follow the very latest global ITS trends.

At the moment, most of the ITS services do not require broadband connection or real-time operation with few exceptions as supplying Internet connection directly to passengers that is also part of the ITS development strategy of Finland [2]. However, the services will inevitably develop further and respective capacity and delay requirements will stringent with services such as passenger internet and video surveillance (Table I); [3]. YTV's vision is that networking solutions should have open interfaces whenever possible for scalability and device manufacturer independence. Also, they should be modular and cost effective thus enabling easy system development.

Our results to be discussed further in this paper indicate that based on inspected networking technology alternatives and respective evolution pathways, YTV has in principle three strategies for ITS development: (1) YTV can buy the networking services/technologies from a telecommunication operator, (2) build own network or (3) to realize a hybrid solution.

The paper is organized as follows. In section I the overview of the study was introduced. Section II presents the backgrounds of the search in more detail. Section III discusses about the service development in the study environment. Section IV introduces available networking development scenarios for YTV. Section V presents the cost analysis of chosen scenarios. Finally, Section VI concludes the paper.

II. Background

ITS services set technical, geographical and economical requirements for networking technology as illustrated in

ISBN: 978-1-61208-105-2

Figure 1. We have divided ITS services into real-time/buffered and basic/supplementary service in order to support flexible and modular networking technology development in the time span of the system evaluation up to 2014. Division to real-time/buffered services is linked to telecommunication networking QoS parameters. This means that the networking technology must satisfy technical service requirements. Division to basic/supplementary services is linked to economical constrains and flexible system realization. We assume that the basic services following definitions of Table I, carry a greater degree of importance in economical sense than the supplementary services. Also, the supplementary services can be realized without a joint telecommunication networking solution though their common management (requiring transmission of a larger amount of data) can also bring up some significant benefits. For example, remotely controlled uploading of advertisements could allow them to be updated several times per day if required, and on-time realized system diagnostics and software updates could increase system operation/maintenance efficiency.

Geographical quality requirements affect especially overall system costs constrained to service quality. For instance, it is not necessarily required for the networking solutions to cover geographical areas that the busses do not run. However, if this is realized, passengers can be offered end-to-end telecommunication services leading to a greater degree on service engagement and potentially to some novel, more usable and/or profitable services. This can be realized especially by using heterogeneous networking concepts linking ITS networking structures to the existing networks, as GSM/UMTS, Wi-Fi or femtocells. The role of networking alternatives of Figure 1 is elementary because they form the bases to inspect the respective ITS service solutions for YTV. In summary, in Networking we research applicable technologies and interfaces. This inspects partly overlapping solutions. For instance, WiMAX and 802.11 mesh- networks can support about the same technical service quality though their costs and technological maturity/coverage differs. In Finland, WiMAX is operated in licensed bands and these bands are interference and congestion free. In Economics we strive to inspect network costs and suggest realization alternatives. In Service requirements we introduce classification of ITS services suitable to YTV, discuss the respective service requirements set to the telecommunication infra, and inspect future development.

III. SERVICE DEVELOPMENT

The project of YTV [14, 15] that we refer in this study, strives to develop cost efficient and user-friendly ITS for the expanding Helsinki Metropolitan commuting area (note that YTV has changed its name in 2010 to HSL [16]). The system includes payment of fares, real time passenger information and online data communication from and to the vehicles (Table 1). The system enables to collect fares from passengers based on agreed tariffs. The first stage of the project is realization of updated RFID-ticketing system in 2009-2011 that is already now practically completed. The system is planned to serve over its 15-year life cycle cost-effectively and YTV expects it to be designed such that it can be easily updated to follow technology and customer needs [4; 5].

Table I summarizes service QoS requirements based on device manufacturer's data sheets and estimated service statistics. Service profiles are constrained to the assumed, required service quality. For instance, data storage requirement for video surveillance depends on video quality and channel delay. Top priority services, as listed by YTV, are as follows:

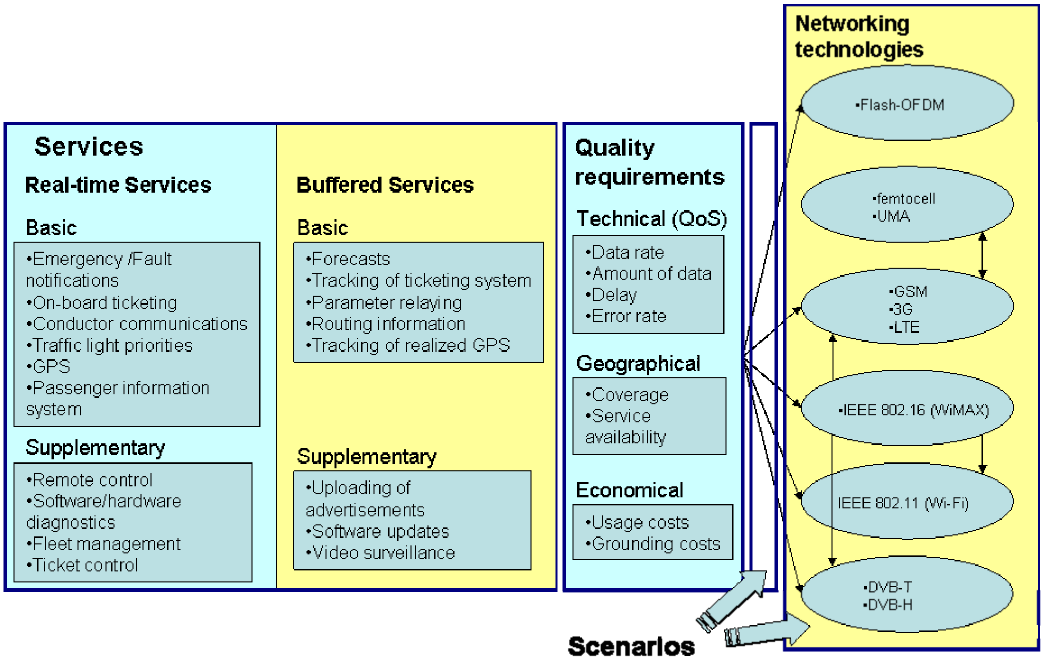


Figure 1. Study Framework

Ticketing system is intended for selling tickets and for ticketing system follow-ups. It enables paying fares by travel card, in cash or by mobile phones.

Traveller information system produces location, route, and buss stop information for passengers. In addition, this information is applied for reporting and for route analysis (congestion follow-ups and real-time timetable updates).

Traffic light priority targets to make crossings faster for public transportation in rush hours.

System upgrades should be modular to follow technology / service development. Also, open interfaces to other fare, ticketing and information systems are required. System components are tracked in the duration of the project for applicable updates from device manufacturers and outsourced service providers. Open interfaces are therefore important.

TABLE I
OVERVIEW OF QOS REQUIREMENTS FOR THE EXPECTED ITS SERVICES IN THE FRAMEWORK OF THIS STUDY.

Service	Basic	Supp.	Real-time	Buff.	Note
Emergency /Fault notifications	▨		▨		few kbit/s
On-board ticketing	▨		▨		slow, real-time traffic
Passenger information	▨		▨		100 kb/s / vehicle
Conductor (in a train) communications	▨		▨		GSM/UMTS, slow, real-time traffic
Traffic light priorities	▨		▨		slow, real-time traffic
Passenger Internet	▨		▨		1 Mb/s / vehicle
GPS	▨		▨		downlink GPS, uplink slow, real-time traffic
Equipment diagnostics		▨	▨		slow, real-time traffic
Remote control		▨	▨		10-100 kb/s / vehicle
Fleet management		▨	▨		downlink GPS, uplink slow, real-time traffic
Ticket control		▨	▨		slow, real-time traffic
Forecasts	▨			▨	below 100 kb / vehicle / 24h
Ticketing records	▨			▨	app. 1Mb / vehicle / 24h
Parameters	▨			▨	below 100 kb / vehicle / 24h
Route information	▨			▨	few Mb / vehicle / 24h
Realized location information	▨			▨	below 100 kb / vehicle / 24h
Advertisement		▨		▨	few
download		▨		▨	10b/vehicle/24 h
Software updates		▨		▨	max 10 Mb / vehicle/24 h
Video surveillance		▨		▨	max 1 Gb / vehicle / 24h

IV. NETWORKING DEVELOPMENT SCENARIOS FOR YTV

Let us now consider various networking alternatives by following Figure 2. The thick lines represent main evolution pathways and the dashed lines supplementary pathways that relate to networking technologies that can be used in some special circumstances as for instance to support geographic high telecommunication traffic density areas. Universal Mobile Access (UMA) of 3GPP refers to development of seamless handover in multimode 2G+/3G/Wi-Fi handsets. The femtocell technology can increase short range (up to 50 meters) 3G coverage and provide new operator based service and pricing models. Digital Video Broadcasting (DVB) technologies (terrestrial (T) and handheld (H)) support video reception especially for special receivers. For further information regarding these scenarios, see [6], where they are explained and argued in more detail.

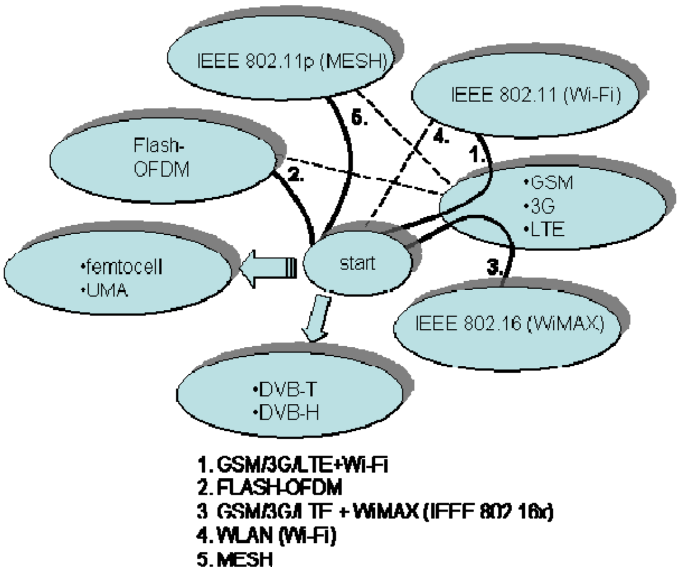


Figure 2. Networking technology development alternatives for YTV- area

Evaluation of network technologies and service environment leads us to inspect two alternative implementation scenarios. The first scenario relays on traditional subscriber-operator business model, where YTV as a transport operator orders all telecommunication services from a mobile network operator. Our second model suggests building of own network. Both options can be supported by Wi-Fi and Flash-OFDM technology.

A. *Being a subscriber in operator's network*

Currently a technically simple solution is to acquire telecommunication services from mobile network operators.

Their network coverage for 2G is country-wide, subscriptions are cheap and there is a wide variety of terminals available. 3G network capacity and performance seems to improve rapidly and recent HSPA upgrades increase download rates up to 3.6 and 5 Mbps. Also, femtocell solutions can enable new interesting business models for both YTV and public transportation operators [7].

YTV presents a big customer for operators with approximately 1700 vehicles. Number of subscriptions is not, however, so large that YTV could necessarily control development of operator's networks to a desired direction. Anyhow, the current technology level should be adequate enough to start implementing the planned real-time services by the operator based model.

Hardware investments should be quite straightforward until 2014. To start with, HSPA-compatible 3G modems could be purchased and later upgraded to LTE-compatible modems. In this scenario, we estimate that operators would start to use LTE widely at latest in 2014, and thus also the migration to LTE-compatible modems would happen at this time. This is in-line with a rule-of-thumb that typical life span for a telecommunication devices, such as core network switches, is about 5-8 years. (We do note that life span for some telecommunication devices such as mobile phones and other related consumer electronics can be substantially shorter.) We assume that in 2014 LTE technology should be well-established.

WLANs and femtocells can be used together with mobile networks. This is due to the fact that operation of mobile phone networks as such may not be cost-effective or convenient for large data transfers such as video surveillance data. WLAN access points can be used in smaller areas such as depots to enable buffered data transfers and relating WLAN based services. Costs for building the WLAN coverage to these small areas should also be relatively modest. Co-operation in femtocells with telecommunication operators can open up new business models [8]. Flash-OFDM networking could be used in parallel with the mobile phone subscriptions for reliability. Mobile subscriptions and terminals can be purchased from multiple operators that should compensate operator dependency. Data streams from different subscriptions could also be combined by multi-homing to enhance networking performance and reliability.

In overall, subscribing services from mobile operators should be easy and reliable. Current service fees are also very reasonable in Finland due to healthy competition environment and there is no reason to expect them to rise significantly. However, YTV should get itself operator based guarantees of the planned QoS before making the final investment decision. This is important due to the development of overall networking loading that is affected by other network users too. An important feature of the operator based model is that the network would be basically owned by the operator potentially excluding the WLAN hot spots. This restricts the way how major part of the network would be developed from YTV point of view.

B. *Building own network*

Operating area of YTV reaches 12 municipalities (in traffic and waste services) in Helsinki capital area which all could utilize the same network. If an own network would be build, it could support a wide variety of municipal services, also other actors than just YTV. This strengthens financial bases of this scenario.

A base of large capacity wireless network is a fast and reliable fibre core network. In YTV case we have estimated that the fibre network would consists of 1000 route kilometres, 500 node points, and 12 000 device ports costing 28 k€ / month (fibre) + 12,5 k€ / month (active devices) + 22,5 k€ / month (network) maintenance, yielding 63 k€ / month or 756 k€ / year. In this estimation we assume the devices would be placed in existing server rooms and power consumption would not be a significant cost factor in the overall budget. Prices are based on the manufacturers' data. In the network, own fibres and/or Ethernet-layer virtual networks (VLANs) could be separated for served parties without trading their QoS requirements. If the own network would be build and marketed wisely, YTV could charge other users as referred earlier (cities, hospital districts, fire and rescue services, operators etc.) so that YTV's own service cost could be subsidizes. If compared to annual costs municipalities need to pay for operators, the 756 k€ annual cost feels a relatively small amount.

Implementing own network would require building a wireless access network between the vehicles and the core fibre network. The access network could be implemented by various technologies: YTV could build cheap Wi-Fi coverage by placing access points especially to those places where capacity demand is high. Wi-Fi is now and probably also at least until year 2014 the most cost-efficient ITS related hot-spot wireless technology.

In the city of Helsinki, there are 450 traffic light controlled crossroads and 3000 bus or tram stops. This makes in total 3500 areas where access points should be placed. Some 7000 WLAN access points would give fairly adequate coverage to bus traffic roads in Helsinki. The forthcoming 802.11p technology supports especially well ITS needs and will spread along Wi-Fi. On the other hand, WiMAX technology can be used to provide larger area coverage for sparsely populated areas and to supply Wi-Fi access points by fixed wireless access (RF-links). By using YTV's fibre core network telecommunication operators could setup high-speed wireless coverage areas to support other network technologies too. For example, Flash-OFDM and 3G-LTE networks could thus be extended. Relating cost saving would benefit both the network operators and YTV.

Own network can be developed more independently and starting from own needs. Established fibre core network can be expected to have lifespan extending up to 2014 and even later. After the payback period, the network would be totally owned and controlled by YTV. In this point, the most significant network expenditures would be up keeping and maintenance fees. Own network could be easily tailored to serve specific service areas and needs. Cooperation with telecom operators would be mutually scaled as own network and services develop.

V. COSTS ANALYSIS FOR THE NETWORKING OPTIONS

To be able to compare the two networking options in terms of telecommunication network investment costs and especially investment sensitivities, we now discuss cost structures.

Being a subscriber in operator's network is more straightforward to analyze, because operators list publicly subscription fees. We can even expect that some volume discounts could be negotiated by YTV.

Option of building own network carries more insecurities. Especially, estimation of route kilometres, the number of required Wi-Fi access points and costs required for cable digging carries insecurities. In rural areas, building a fibre network would cost some 5-6 €/meter including work and the cable [9]. Expenses can be divided approximately fifty-fifty to work and the cable. In city areas cable digging is significantly more expensive especially due to opening and restoring asphalt and revetments. Also, existing cables and pipelines make work demanding and difficult. Thus an opportunistic network building utilizing existing cables, tunnels, and other infrastructure should be applied always when possible. Let us estimate, based on comparable projects as referred in [10], that setup expenses would be 5 million Euros for the fibre network covering the assumed 1000 route kilometres. Thus, for the network, it would cost 5000 €/km to dig the cable. (This relatively low level cost estimate is based on using existing underground cable pits and channels. Without them the cost would be substantially higher, eg in the order of $40/ meter [11].) If the investment would be funded with a 5 M€ bank loan, the total expenses for 7 years annuity loan with 4.5% interest rate would be 5,8 M€ with a 25 years payback period yielding 8,3 M€. (Expenditures for the active devices were listed in the previous chapter.) Also in the case of own network, wireless access network is required to connect the vehicles. YTV estimates that at least 7000 access points are needed to provide adequate coverage.

If the price for an access point, antenna and outdoor-box would be 200€, the resulting cost would be 7000 x 200 € = 1,4 M€. Network maintenance and operation could be outsourced as with the fibre network. However, in this case the price per access point should be much lower, e.g., 5 €/access point making 7000 x 5 € = 35 000 € in total per month.

Wi-Fi is not the only technology which can be utilized to build the access network, and e.g., WiMAX could be an alluring option in the future. Currently, however, due to equipment prices, frequency allocation issues, and terminal availability, Wi-Fi might still be the most viable option.

Own wireless access network can also be supported by 3G subscription. For example, one 3G subscription in each vehicle would increase service reliability especially at the beginning when wireless coverage is not optimized and there is no longer term experience how it works. Costs of various wireless networking options are listed in Table 2.

TABLE II
OPERATOR SUBSCRIPTION FEES

Access technology	Monthly Fee	Terminal Price
GPRS / 3G	10€	100€
Wi-Fi Access Point	5€	200€
Wi-Fi Terminal	0€	50€
FLASH-OFDM	40€	200€
In-Vehicle Equipment	0€	4000€

Total costs for fixed and wireless networking are shown in Figures 3 and 4. Calculations are done for 7 year bank loan and the monthly costs are shown on y-axis. Calculation parameter sensitivities are shown on the x-axis. The parameter change refers to a price change of a single acquisition unit. For example, in the case of the fibre, an increase of 10% could mean either an increase in the digging costs or fibre price so that it results 10% increase in the total fibre costs. The price of the fibre itself is, however, a relatively small factor in the overall costs (and also easier to predict), and it would therefore be useful to estimate especially the digging costs. However, this requires precise understanding of the current installation infrastructure in the YTV area such as access to a municipal tunnelling, piping, and electricity maps and plans. At the time of this study, the actual implementation alternatives were not yet available that results great cost sensitivities in this respect.

In Figure 3, sensitivity analysis indicates that the costs of fibre network are somewhat critical. Costs of other components are easier to estimate due to existing market information. If the fibre network would be paid in the first 7 years, it is "free" to use as long it has enough capacity to serve the users. We may estimate that the capacity of a metropolitan wide, high speed fibre core network would be enough even for the next 30 years demand [12].

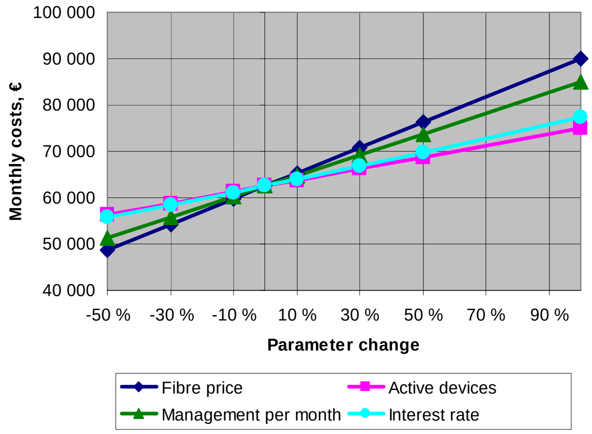


Figure 3. Sensitivities for monthly costs of fixed network

In Figure 4, we note that Flash-OFDM subscriptions are quite expensive. On the other hand, they can support high overall QoS provided that the number of subscribers remains in system limits set by the applied 450 MHz bandwidth in Finland. In-vehicle equipment for 1700 vehicles including on-board servers, Wi-Fi terminals and connections to peripherals are the most sensitive for price fluctuations. For example, if the price of in-vehicle equipment increases by 20%, it becomes more expensive than the Flash-OFDM offered by current price. In-vehicle equipment CAPEX depends on device and installation costs (including network installations) that depend on vehicle type, contractor etc. Based on earlier studies, the installation environment can vary greatly in vehicles and carriages [13]. Building and maintaining the Wi-Fi network of 7000 access points seems currently reasonable priced. However, some cost margin can be obtained using mobile subscriptions as a backup to start with.

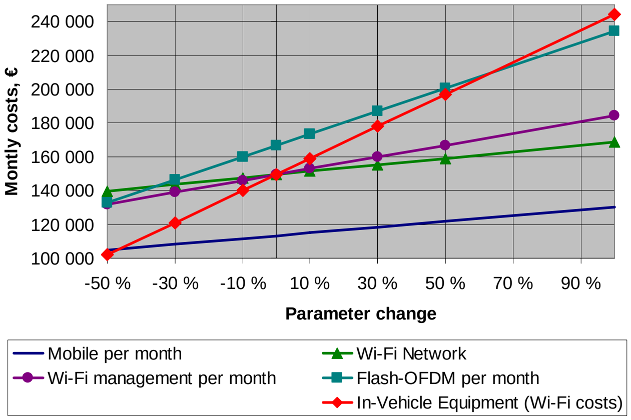


Figure 4. Sensitivities for monthly costs of wireless network

Sensitivity analysis indicates that there are certain cost parameters, where price fluctuations result major changes in overall costs. In the case of building own network, the fibre price has a significant role in the total costs. Especially the in-vehicle equipment costs and the Flash-OFDM subscriptions contain significant uncertainties. Fibre and in-vehicle equipment price uncertainty could be reduced substantially with a detailed implementation plan and by carrying out more detailed studies. By introducing new revenue components into the case of building own network, the cost structure could be improved and make the business case more solid. Especially, use of femtocells to supply common services for operators and YTV should be further investigated. In summary, a separate service mapping and related investment analysis is recommended for rigorous project risk management.

VI. Conclusions

We have inspected telecommunication networking, services and economics for YTV ITS solutions. In summary, we started by listing the required and optional buffered and real-time ITS services. When telecommunication networks get faster, need for buffered services will reduce. Core networking can be basically based on buying 3G/LTE service from telecommunication operators or to build own optical network. 3G/LTE can naturally support access networking too. If the optical network solution would be selected it carries significant initial costs and greatest investment sensitivities relate to fibre network setup costs. In this scenario, YTV could even rent extra optical network capacity to telecommunication operators. The second option, buying capacity from a telecommunication operator, would carry costs more in OPEX. Common services should be decided and negotiated with the network operator. Core networking solutions should be supplemented by Wi-Fi base stations. Usage of femtocells could support new business models with the network operators. ITS solutions require investments also to vehicles where Wi-Fi equipment (routers) and installation costs are important. If prices of Wi-Fi equipment develop positively, as it seems now, all applications of Wi-Fi technology will come more favourable. Installation costs will reduce due to vehicle manufacturers incorporating telecommunication infra to new vehicles.

In summary, cost structure and sensitivity analysis should always be used to support ITS investment decision making and to find appropriate pricing models for services. Application of ITS in any city area faces similar problem framework as we have discussed in this paper and therefore scenarios of YTV should also be applicable in a more general level. We have provided examples of various aspects that should be considered when establishing city-wide ITS networking solutions. If establishment of own optical core network is seriously considered even in some extent by YTV, investment economics can be improved if part of the network capacity is rented to third parties, as telecom operators, companies or officials in the YTV area of municipalities. Generally, cooperation with telecommunication operators can be useful. For instance, there are services that interest both YTV and network operators as femtocell technology. Therefore, searching and supporting common services should be used to support mutually beneficial business solutions.

ACKNOWLEDGMENT

We would like to thank Helsinki Metropolitan Area Council YTV for funding the research that resulted among others, this article. Also, the audition of the original research done by external opponents, professors Olli Martikainen (The Research Institute of Finnish Economy) and Hannu H. Kari (National Defence University) is highly appreciated.

REFERENCES

[1] Digita Ltd.: @ 450 Wireless Broadband: Available at: http://www.450laajakaista.fi/9019/9039/9847/9081, Accessed August 2010

[2] Antti Rainio: “Development strategy for the intelligent transport in year 2005-2010”, ITS Finland publication 5/2005.

[3] K. Sharma, H. Sharma and A. K. Ramani: “Towards In-Car Ad Hoc Network for Pervasive Multimedia Services”, International Journal of Computer and Network Security, Vol. 2, No. 3, March 2010.

[4] Anderson, A.: Ticketing and Information System 2014. Helsinki: YTV, 2008

[5] Seminar by Tallinn Training and Development Center: “Tallinn-Turku-Helsinki-Riga interoperable ticketing cooperation”, Available at: http://www.takk.ee/index.php?id=14 . Särgava allee 4, Tallinn (Pirita), October 22nd -23rd 2009

[6] Arto Karila, Timo Korhonen, Reijo Salminen, Tapio Väärämäki: “Laajakaista joukkoliikennevälineissä verkkoteknologiaselvitys”. ÄLLI publications 1/2008, The Ministry of Transport and Communications of Finland, 2008, Available at: http://alliohjelma.info/?action=file&id=19&file=19.pdf, Accessed June 2010

[7] D. N. Knisely, T. Yoshizawa, and F. Favichia: “Standardization of Femtocells in 3GPP”, IEEE Communications Magazine, September 2009

[8] Motorola: “Navigating Operator Business Model Concerns for Femtocells”, White Paper, Available at: http://www.motorola.com/staticfiles/Business/Solutions/Technologies/Femtocells/_Documents/Static%20files/Navigating_Operator_Bus_Model_Femtocell_558186-001-a.pdf, Accessed June 2010

[9] H. Äyväri and M. Kääriäinen: ”Fiber to Home (FTTH)”, Publications of the Ministry of Transport and Communications 33/2005.

[10] Raffaele Bruno, Marco Conti, and Enrico Gregori: “Mesh Networks: Commodity Multihop Ad Hoc Networks”, IEEE Communications Magazine , March 2005

[11] Ars Technica: Fiber-to-the-X: the economics of last-mile fiber. Available at http://arstechnica.com/telecom/news/2010/03/fiber-its-not-all-created-equal.ars, Accessed June 2010.

[12] Bigo, S.: "Multiterabit/s DWDM terrestrial transmission with bandwidth-limiting optical filtering," Selected Topics in Quantum Electronics, IEEE Journal of , vol.10, no.2, pp. 329-340, March-April 2004

[13] Väärämäki, T.; Hämäläinen, T.: On-board broadband in trains: Proof-of-concept phase and New business models, Publication 41/2006 by Ministry of Transport and Communications of Finland, 2006

[14] HSL: “Ticketing and Information System 2014”, Available at: http://www.hsl.fi/EN/planning/TIS2014/Pages/default.aspx, Accessed 5.8.2010

[15] HSL: “Mobility management aims at influencing people’s choice of mode of transport”, Available at: http://www.hsl.fi/EN/planning/mobilitymanagement/Pages/default.asp , Accessed 5.8.2010

[16] HSL: “Helsinki Region Transport”, Available at: http://www.hsl.fi/EN/Pages/default.aspx, Accessed 5.8.2010

Streaming Cloud Service Concept by Peer-to-Peer Distributed Technology

Takeshi Tsuchiya
Faculty of Business Administration and Information
Tokyo University of Science, Suwa
Nagano, Japan
t-tsuchi@rs.suwa.tus.ac.jp

Hirokazu Yoshinaga, and Keiichi Koayanagi
Faculty of Science and Engineering
Waseda University
Kitakyushu, Japan
{yoshinaga@akane, keiichi.koyanagi}@waseda.jp

Abstract—**In this paper, we discuss management manner for streaming services provided on complex network environment, and propose streaming management platform for realizing Streaming Cloud Services, which doesn't depend peer environments. These streaming service-based applications would be expected to diffuse more greatly as the next generational services, and have been used for security surveillance, vehicle-to-vehicle communication, and route guidance service. Therefore, it is required to achieve our proposal platform that overcome several difficulties of streaming systems, network environment, and functions for stable services continuously by distributed computing technology. We focus on the generating of the streaming session instead of each peers allocated environment, and the issues for continuous the service such unstable environment. The experimental results showing the streaming service with best effort using several hundred sessions are described in this paper.**

Keywords-** **Streaming Service; Peer-to-Peer; Cloud Computing.

I. INTRODUCTION

Demands of streaming services collaborated by distributed camera and devices allocating in ubiquitous services have been increased as for the applications such as security surveillance, vehicle-to-vehicle communication, and route guidance service [1]. Streaming data generated from cameras and devices are also used to improve service accuracy and robustness, and attachment devices are also utilized as for service accuracy such as a GPS (Global Positioning System) and acceleration sensor.

But the condition (status) of network changes dynamically on overlay networks and streaming data, although it is expected to use them whenever. And it is not ensured the quality and conditions of network to use specific streaming applications as the codec and control by the each application because the environment of peers would be complicated by location of connections. Therefore, we discuss the manner, which realizes streaming communications in these complicated environments, and propose the streaming management platform adapted several streaming and network environments using P2P (Peer-to-Peer) overlay networks [4]. The P2P overlay networks is to construct logical space based on peers, and allocate services by each peers on that space [5]. These are provided on the overlay network are divided to two types by their characteristics, synchronous and asynchronous type services. In the synchronous type services, the streaming service, which means the interaction service by sharing same time, such as the Skype [6], is very used for communications. Face tracking and recognition systems have been developed to use CV (Computer Vision) technologies with real time video streams. The asynchronous service is added based on the file synchronous one. However, the communication band for real time sharing should be required between distributed peers. In addition, the peer fluctuates in the overlay networks because of the move of a user from the network, and its network and system environments. Therefore, our goal is that a user does not need to use a specific application, and can use services except the knowledge of the network.

Java Spaces [7] has been proposed for the framework of distributed components. It manages connections and allocations; the unstable network environment such as large distribution of component is not provided in this framework. In Open Call Media Platform [8] proposed by HP Inc., it is possible to make sessions among nodes utilizing different streaming service. A static server is set as for codec transformation service in the border of closed streaming network. In this case, some problems should be solved to accept several codec and stream control methods, and scalability for the fixed a server caused by the concentration of traffic. This proposal model only solves managed and planed services like telephone networks.

In our considerations, streaming service can be used among distributed peers on overlay networks, and named it streaming cloud service. It enables to generate streaming sessions in best-effort network route on that time.

In the following part of this paper, we propose streaming platforms including adaptation manner on the route of session, one is the CODECs of streaming data and another is streaming control are adapted to peer environment. At the same time, our proposal provides priority control of streams by applying the session condition for unstable environment. As the result, it is available to make best-effort route for

ISBN: 978-1-61208-105-2

streaming services, and evaluate that functions and service scalability as for streaming platforms.

This paper is written as follows. The modeling of the proposed streaming platform in the Section II, and the proposal and implementation method are written in the Section III. The evaluations are described in the Section IV. Finally, the paper is concluded in the Section V.

II. FUNCTIONS OF STREAMING PLATFORM

In this section, we clarify the definitions and assumptions of service environments in this paper.

A. *Service Environments*

Streaming services in this paper are defined as follows: streaming services provide following functions either or both, generating the sessions and sending streaming data. Currently, the streaming services have own stream control methods and use of codec in the stream data described above. Therefore, controlling streams for starting communication among distributed objects, each system need to use same control methods and same data CODEC for service inspective each peer environments. Since peers provide most of the streaming services on the overlay networks, these services have the possibility to causes instability and loss of the services.

B. *Functions*

Some functions are needed in the network between different peers, as follows.

1) *Detection of Services*

Peers in overlay networks by normal users, and the time pass must change its number of services, status, and performance dynamically. The information of these peers is notified to other peers in the process of joining and separation to the overlay network.

2) *Retrieval of Services*

The retrieval service would be important functions, and used in overlay networks by query indicating network ID from detected service, and service type of peers. However, the reliability of overlay network is not so high caused by each peer without servers. Therefore, all information shared among peer is updated periodically, and normal peers without fixed peer provide all services.

3) *Allocation of Components in Streaming Session*

The components and allocation of route on streaming session is decided by the function on platform when the peer generates streaming communications. The “trans-code” (Trans-code means transferring function of CODECs such as from Mpeg 2 to DV) service is selected suitably from overlay network to transfer data codec, and it is allocated on session between peers.

Relay peer that relays all communications on overlay networks is allocated for through NAT, and firewalls on the session between peers because of user’s environments. A CODEC is transferred to another through trans-code service, and it transfers the data continuously while the session existing. Functions for controlling the streams are independent from CODEC transferring, and defined “basic control interface” such as “request” and “stop” common methods. Therefore, it happens asynchronously in streaming session. The orders of controlling method make status changing. Relay peer also provide to adapt environment for transport protocol.

C. *Stable Session Provision*

Created session has issues of buffer overflow and the debasement of its quality. The size of buffer and the quality are adjusted to recover one.

III. PLATFORM FOR IMPLEMENTATIONS

In this section, we propose streaming platform that include efficient functions demanded previous sections. This proposal platform is allocated as for middleware between user and each streaming application shown in Figure 1. The role and provided functions of each layer shown in Figure 1 is as follows.

A. *Pipe Management Layer*

This layer manages all communications and connections to other platforms implemented other peers. In these communications, there are not only but collaborative data, all communication data included management data of overlay network, query for retrieval service, administration information of shared information and so on utilize this layer.

This layer only manages all communications, therefore, peer and service information notified to overlay network are controlled in upper layers such as retrieval service, service selection and so on. The update of peer information is distributed and managed periodically, and statistic information is also performed derived from peer by the demands. In the retrieval service, the query for retrievals and its result are sent and received in the layer by the demands. In the streaming services, its data are treated between platforms as shown in Figure 2. Each session is identified as an added pipe with 16 bit ID, and is managed by a table based on the ID.

1) *Session Environment Support*

The transport protocols (UDP, TCP, Http over TCP) [9] which each streaming service adapt in peer environment are chosen and use to create streaming session for best effort, while user generate the session by streams. The transport protocols are corresponding to each peer environment based on received peer and service information, and transport protocols are decided by demand from the result of analysis in upper layer.

2) *Stable Streaming Service*

The sending and receiving data in each session is kept continuously while the session exists. It needs to handle plural streams depending on the service condition at the same time. Some peers are assumed to provide trans-code services that are able to transform CODECs, and to generate new sessions. Each session is controlled to keep service quality, and it is available to stably treat plural sessions. The peer separation can be notified as the problem occurs because RTP and RTCP [10] are utilized between streaming platforms.

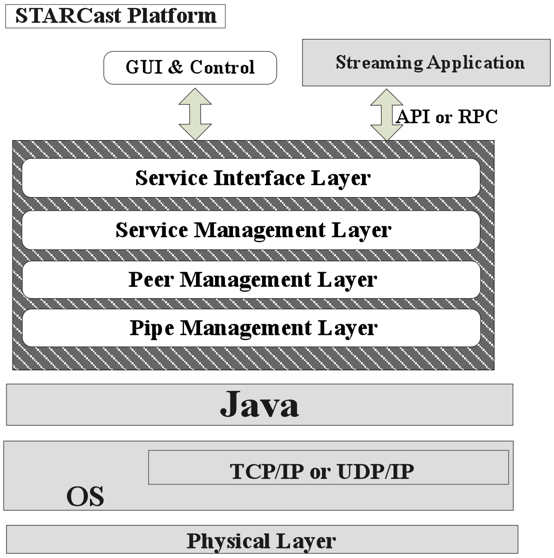


Figure 1. Platform Stack

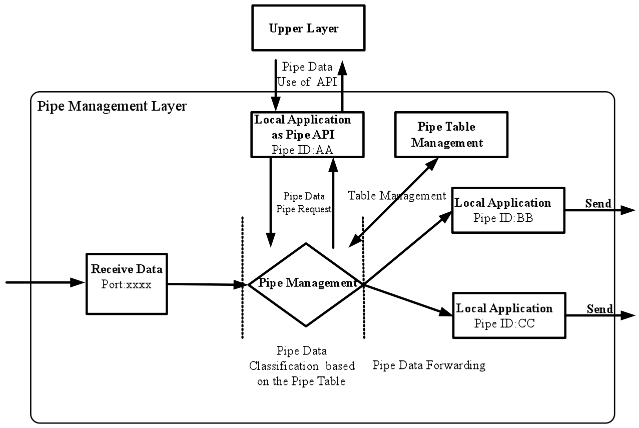


Figure 2. Pipe Processing

B. *Peer Management Layer*

This layer manages taken peer information, service information, and management information for generating session.

1) Distributed Information Management

Peer and service information notified and shared to overlay to network are written as following elements. A 128-bit ID (IDentification) is used for identification of each peer, and the ID and unique name are for user one. Peer and service information are identified with this ID and name in the upper layer of platform.

2) Service Retrieval

Our proposal platform provides function of retrieval service for peer and service information notified to overlay networks. Writing detail, the information for codec of streaming data and stream control protocol in each streaming service, peer ID and so on is used as query for retrievals by the platforms. In this function, the retrieval manner is used our proposal distributed information retrieval manner [5], which realizes almost same availability comparing current information retrieval services like google among distributed peers without any servers. Therefore, all users can retrieve information same as current normal services on overlay networks.

3) Adaptation for Streaming Control Environment

Transition of service status can be controlled by each stream control method such as session generation and disappearing, start and stop of stream sending and receiving. It is required to control stream that does not depend on the environments of session environment. Therefore, this layer adapts and absorbs various streaming control methods. We define common stream control method based on the conceptualistic transition of service status because specific stream control method would not unify any services in the near feature. The control command and a list of its order are shared in the control method. As for example, this control method corresponds to RTSP [11], SIP [12], and DVTS [13] in present implementation in TABLE 1.

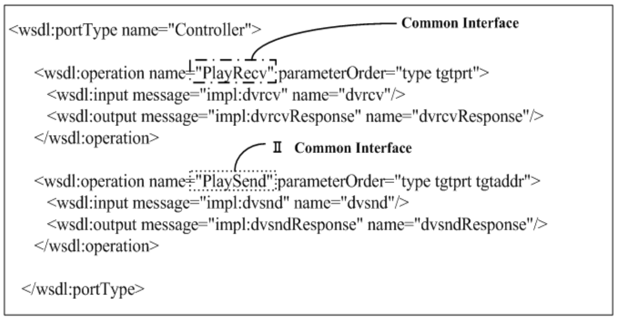
Some system except a stream control function exists. The system can be control from outside, and some scripts are prepared to solve this problem.

TABLE 1. Mapping of Common Interfaces

Common Interface	RTSP	SIP	DVTS
Init	SETUP	BYE	etup.sh
Play-Snd	PLAY	INVITE	dvsnd
Play-Rcv	Rcv	INVITE	dvrcv
Stop	TEARDOWN	BYE	stop.sh

4) Collaboration with Applications

A streaming application is connected to the platform, and it can be controlled with the stream control method written in TABLE 1. Common Interface is defined Common Stream Control Method as for utilizing streaming services seamlessly, and implemented by Web Service Description Language (WSDL) shown in Figure 3. The defined method by mapping enables to change status for services.


Common Interface
Ⅱ Common Interface

```
<wsdl:portType name="Controller">

  <wsdl:operation name="PlayRecv" parameterOrder="type tgtprt">
    <wsdl:input message="impl:dvrcv" name="dvrcv"/>
    <wsdl:output message="impl:dvrcvResponse" name="dvrcvResponse"/>
  </wsdl:operation>

  <wsdl:operation name="PlaySend" parameterOrder="type tgtprt tgtaddr">
    <wsdl:input message="impl:dvsnd" name="dvsnd"/>
    <wsdl:output message="impl:dvsndResponse" name="dvsndResponse"/>
  </wsdl:operation>

</wsdl:portType>
```

Figure 3. Implementation by WSDL

C. *Service Management Layer*

The services allocated in session are chosen from all peers who have same functions by peer selection manner, and the construction of the session is determined and generated by chosen members.

1) *Generating the Session*

In the case that peer environments are different for generation of session, codec transferring should be performed in the session, and allocates trans-code service for transferring stream data. The service for transferring is determined by the total load and codec demand on that service. We could assume that there are many services existing on overlay network, and a peer that creates session can provide codec transferring. The service is located on the session between constructor peers, and starts the transferring. If relay of peers is required in the network environment, peers who compose sessions are tandem located, and they have some effects to the service processing, quality of streaming service of peer.

2) *Selection of Peer and Service*

Selection of service, which chooses services provided, peers by their condition of status as best effort, and these chosen peers compose streaming session. The authors have proposed the adaptive coordinator election (ACE) platform [14] to select peer as best effort on overlay network.

First, the candidate service for session are chosen by their conditions, transferring data codec and transport protocol, and taken by retrieval service written in section B. Where, the algorithm is used in the Java Bench [15] for derivation of peer processing. After this process, the selections of services are executed and chosen the service that composes the session by our proposed algorism based on derivation result. After all, each user could create a streaming session by best efforts. The result of selection is listed, and shared this information as for back up of streaming session. However, status of listed peer would be change by time pass, and execute selection again. Therefore, the list of peer that the order is from creation can be updated to reconstruct session for best effort. The period of update makes load big in the network and the platform, and that is ten minutes in our system.

D. *Service Management Interface Layer*

This layer provides interface for GUI and control to a user, and the streaming application has the function of API and proxy in the platform.

1) *API and Proxy Function*

TABLE 2 shows an example of API (Application Programmable Interface) for streaming service developer and upper streaming services as functions of streaming platform. Each API can execute each function mentioned above. In the following, APIs are sample that adapt to basic functions for streaming sessions and managements. The proposed platform is allocated and affected as local proximal node for existing streaming services and applications, not only for news streaming service. Therefore, it effectively uses the existing resources. Therefore, it makes to use existing resources effectively.

TABLE 2. Examples of API

Module	Method (Object type)	Explanation
BSC	notify (Network ID)	Notifying join and separation
BSC	set, get (Peer ID)	peer information management
SRV	retrieval (Query)	retrieval the query
ARV	selection (Peer Array)	peer selection
CTL	sndCtrl (Cmd)	control request
CTL	getStatus (Peer Array)	session peer information

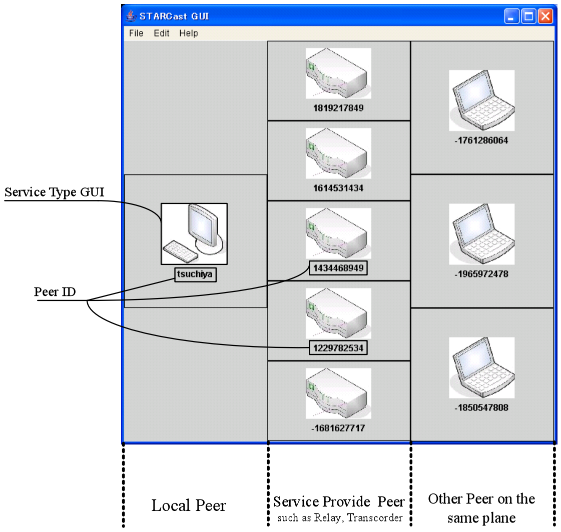


Figure 4. GUI for peers

Figure 4 shows GUI provided for users. Each user can detect peers and provided services on the overlay network. User can control, and generate streaming session.

E. *Transcode Service*

Trans-code service is one of functions provided by a proposal platform and a peer. These are allocated on the streaming session by the demands on peer environment that means data codec of stream and connecting network environment. It performs as a process, and the codec of streams would be transferred adaptively, while transferring streaming data on session.

In the current implementation, each peer allocates this function as a service. When the peer generates the streaming session, most suitable trans-code service are chosen adaptively from all same services provided overlay network by above mentioned manner.

IV. Evaluation

In this section, our proposal platforms are evaluated adaptation functions for each environment by the simulation and implementations.

A. Service Scalability

In this section, a service on overlay network by simulation is evaluated, and adaptability is also done with an implemented platform. The environment of simulation is Pentium 4 2.8 [GHz], RAM 2,048M [Bytes], Linux 2.4.28, and Java J2SE 1.5, and constructs streaming environments assumed distribution of several services on the same overlay network. Each peer in overlay networks composed overlay networks, and generates streaming session among these peers under their environment.

The "Network Link" in Figure 5 means link of the Internet connection, and the ratio of network link in each peer changes dynamically by the traffic under Poisson distribution (λ=1). In this time, the loss of data as the throughput of transit ranges in $0 \leq$ Link ≤ 30 [%]. The trans-cord services which transfer the CODEC of streaming data, are set in the point A or B, and execute the service for the sessions of peers.

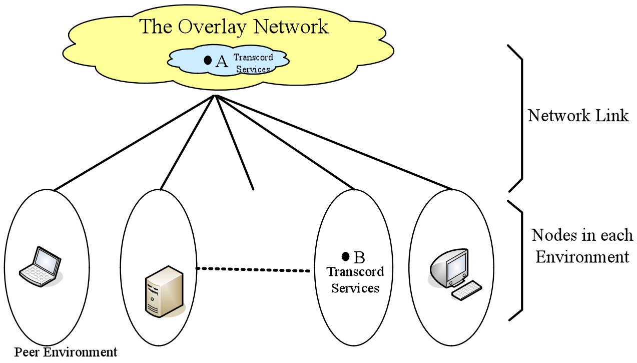


Figure 5. Environment of simulations

The result of simulation is shown in Figure 6. The x axis shows the number of streaming sessions among peers on the overlay network, and the y axis shows increment ratio of session time comparison to normal one. At this time, the trans-code services are disappearing randomly. Each lines in Figure 6 show the difference of number of services for trans-code and the selection manner. These lines are affected by other streaming sessions, and affect each other. "The line xx%" in Figure 6 mean the ratio of trans-code service for all streaming sessions. And, "Static Server" means a trans-code service provided by a static server on the point A. "Normal-50%" means the generated streaming sessions without above mentioned peer selection, and the peers who need the service use nearest service.

In the range of x-axis between 0 and 50, these lines are looks similar. But the "Static Server" line would be occurring increment rapidly caused by concentration of streaming traffic to server. But, in the case of changing the number of peers dynamically such as P2P overlay networks; there is problem in the viewpoint of the scalability of service. This system needs to provide load-balancing manner of this service.

The effect of peer selection manner enables to compare "Proposal-50%" with "Normal-50%" in the same number of sessions, and these lines have almost same characteristic in the range of less 100 sessions. In the more large range of x-axis, the increment of "Normal-50%" value grows large rapidly. This increment of line have caused by the selection of specific services for trans-coding with bias in the streaming session. Therefore, the concentration of traffic to specific services makes the line of increment ratio larger. Comparison of lines using our proposal ("Proposal-xx%"), these lines are ranges increment inversely proportional to the number of service allocation such as 50%, 25%, and 20%. However, all the transmission time is almost fixed by increasing the session.

Therefore, the streaming session for best effort is created from trans-code service distributed ion overlay networks in the proposed platform.

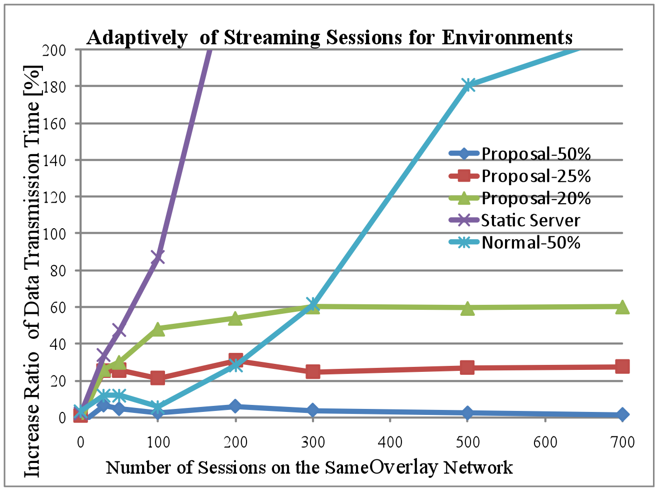


Figure 6. The number of session and the increase ratio of data transmission time

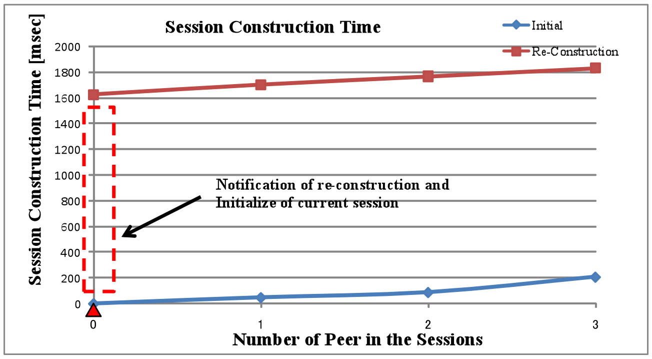


Figure 7. Construction and Reconstruction of sessions

ISBN: 978-1-61208-105-2

Figure 7 indicates the time sequence to generate new streaming session between peers, and re-construction time from session broken. x and y are the hop number and the construction time in this Figure. *Initial* indicates time to notify session requirement to all peer for construction, and to create session based on the response from peers. *Re-Construction* is the time to stop a session in the time 0. The peer in the session detects the changing and disappearing the peer and trans-code service that reconstructs a session. The peer notifies the other peer constructing the session. In each receiving peer, a creating session is initialized, and the session is reconstructed. The time is 1,600 m[sec] from notifying session stop to reconstruction beginning in this graph.

V. Conclusion

We proposed streaming management platform for streaming cloud service on overlay network. The streaming service can be provided on unstable overlay networks without considering environments of each peer, network environment, and services. The stable service is shown using proposal platforms including some functions to avoid the delay and data loss between platforms.

However, scalability of service should be improved, and need to consider the manner for continuous service in an unstable session. In this time, the session should be re-generated to others when a user does not notice it. An idea is to use seamless service, and the changing is available that it does not need to stop the session for future works.

Acknowledgment

This work is partly supported by Grant-in-Aid for Research Activity Start-up (21800063).

References

[1] H. Sawano and M. Okada, "Next-generational Route Guidance Method by Using Multimodal Information Integration", Proc. of Information and Computer Elements, pp.273-278, Sept.2007

[2] A. Barton-Sweeney, D. Lymberopoulos and A. Sawides, "Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks",Proc. of the 3rd Int'l Conf. on Broadband Communications, Networks and Systems, 2006, pp.1-10, Oct.2006

[3] R. Tron, R. Vidal and A. Terzis, "Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks", Proc. of ACM/IEEE Distributed Smart Cameras, 2008, pp.1-10, Sept.2008

[4] E. Lua, J.Crowcroft and M.Pias, "A Survey and Comparison of Peer-to-peer Overlay Network Schemes", The Electronic Magazine of Original Peer-Reviewed Survey Articles, vol.7,pp.72-93, Mar.2006

[5] T. Tsuchiya, M. Lihan, H. Yoshinaga, and K. Koyanagi. "Distributed Information Retrieval Service for Ubiquitous Services" Proc. of Availability, Reliability and Security, pp. 842-850, Barcelona, March 2008

[6] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", technical reports 2004, Columbia University

[7] Sun Microsystems Inc. JavaSpaces Service Specification, version 1.1, October 2000, http://www.sun.com/jini/specs/js11.pdf, [Retrieved; August 24, 2010]

[8] Y. Makino, Y. TAN, "Integrated Visual Communication Network Architecture", Proc. Of Towards Peta-Bit Ultra-Networks, Ishikawa Japan, pp. 156-161, September, 2003

[9] T. Tsuchiya, H. Yoshinaga,and K. Koyanagi "STARCast: Streaming Collaboration Architecture on Heterogeneous Environment Everywhere", Proc. Of ACM Multi Media, pp. 57-62, October 2004

[10] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Request for Comments: 1889, January 1996

[11] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol ", RFC 2326, April 1998

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol", RFC 3261, IETF, June 2002.

[13] A. Ogawa, "DVTS:http://www.sfc.wide.ad.jp/DVTS/" [Retrieved; August 24, 2010]

[14] H. Yoshinaga, T. Tsuchiya, and K. Koyanagi, "Coordinator Election Using the Object Model in P2P Networks", Springer Berlin / Heidelberg, Volume 3601/2005, pp.161-172,

[15] Java Benchmark, http://mathsrv.ku-eichstaett.de /MGF/homes/grothmann/java/bench /Bench.html, [Retrieved; August 24, 2010]

[16] Fine Packet Builder,http://www.fineqt.com/, [Retrieved; August 24, 2010]

Evaluation of the Applicability of the OSGi Service Platform to Future In-Vehicle Embedded Systems

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

Arne Koschel, Roman Roelofsen
Faculty IV, Dept. for Computer Science
Applied University of Sciences and Arts
Hannover, Germany
akoschel@acm.org, roman.roelofsen@googlemail.com

***Abstract*—One promising market for embedded systems is the automotive industry. For example, engines, dashboards, wipers, lights, doors, windows, seats, mirrors, radios, CDs, DVDs, and hand-free phones are being controlled by in-vehicle embedded systems. This paper evaluates the applicability of the OSGi Service Platform to future in-vehicle embedded systems. One specific application area, which can greatly benefit from this evaluation, is Web services, where a number of future scenarios have not been supported by the OSGi Service Platform yet. This is exemplified by two example scenarios, viz., a car tracking service and an advertising service. Based on these example scenarios, a number of requirements that should be met by the OSGi Service Platform are identified, viz., dynamic availability, versioning, persistence, composition, remote management, platform independence, security, and distribution. In addition to the identification of the requirements, another contribution of this paper is the evaluation of the OSGi Service Platform against these requirements. This evaluation will help to extend the OSGi Service Platform to be applicable to future in-vehicle embedded systems.**

Keywords—OSGi Service Platform, OSGi Framework, in-vehicle embedded systems, Web services, car tracking service, advertising service

I. INTRODUCTION

"Nowadays, for any activity in our everyday life, we are likely to use products and services, whose behavior is governed by computer-based systems also called *embedded systems*" [4]. Embedded systems constitute the biggest sector in the market today. "Of the 9 billion processors manufactured in 2005, less than 2% became the brains off new PCs, Macs, and Unix workstations. The other 8.8 billion went into embedded systems" [2].

This market trend also affects the automotive industry, one of the largest economies in the world. Over the last two decades, there has been an exponential increase in the number of computer-based systems embedded in vehicles also called *in-vehicle embedded systems* (or *automotive embedded systems*) [4].

In-vehicle embedded systems are currently used for navigation, climate control, adaptive control, traction control, stabilization control and active safety. In the future, they will also be used for remote diagnostics of a vehicle, access to the Internet and audio/video entertainment. The cost of in-vehicle embedded systems constitutes more than 25% of the total cost of a vehicle today [11]. In 2010, in-vehicle embedded systems will account for 40% of a vehicle's content [3].

The remainder of this paper is organized as follows. At first, the paper describes example scenarios of using Web services in future in-vehicle embedded systems, viz., a car tracking service and an advertising service. Next, the paper lists requirements driven by the example scenarios, viz., dynamic availability, versioning, persistence, composition, remote management, platform independence, security, isolation, communication, and distribution. Finally, the paper evaluates the OSGi Service Platform to see if this platform can meet the requirements of the example scenarios.

II. MOTIVATION

The motivation that leads us to evaluate the applicability of the OSGi (Open Services Gateway initiative) Service Platform [8] to future in-vehicle embedded systems stems from the following facts:

1. The OSGi Service Platform originally targeted gateways (as can be deduced from the platform name). However, the platform has been adapted to many other domains, including vehicles.
2. Vehicle manufactures (producing around 70 million cars per year [20]) have showed interest in the OSGi Service Platform. Simple evidence of this fact is that Automotive Multimedia Interface Collaboration (representing major vehicle manufactures) has joined the OSGi Alliance [18].
3. Vehicles are a promising market for embedded systems. For example, engines, dashboards, wipers, lights, doors, windows, seats, mirrors, radios, CDs, DVDs, and hand-free phones are controlled by embedded systems.
4. More than 80% of all new innovations in vehicles are based on embedded systems [19]. For example, when an accident causes an airbag to inflate in a Cadillac, an embedded system in the car emits a signal for the global positioning system (GPS) service to get the car's position and then communicates with the driver's cell phone to send the car's position to the rescue service [4].

III. EMBEDDED SYSTEMS

An *embedded system* is "a micro-processed device, thus programmable, which uses its processing power for a specific purpose" [1]. It typically consists of memory (such as RAM, EPROM, ROM or flash memory), a processor (such as Intel x86, PowerPC or ARM), a clock and an input/output device (see Figure 1).

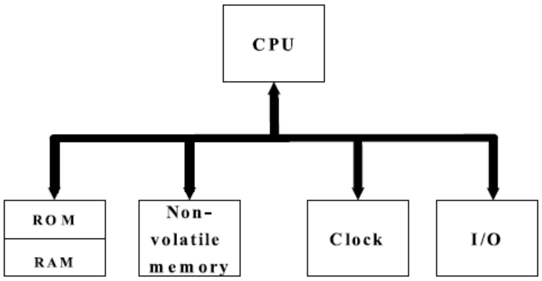


Figure 1. Embedded system [1]

Because of small memory footprints, embedded systems are also called *small-memory devices*. However, the size of software embedded in vehicles has been growing. For example, in 1980 and 2000, a Peugeot contained 1.1 KB and 2 MB, respectively [4]. In 2010, some vehicles may contain 100 million lines of code [3].

The growing size of embedded software also raises a question of its quality. For example, in 2003, 49.2% of all car breakdowns in Germany were due to bugs in embedded software [4].

IV. WEB SERVICES

A *Web service* is "a software system designed to support interoperable computer-to-computer interaction over a network" [10].

The market trend towards embedded systems gives rise to the idea of using Web services in embedded systems. However, this is a challenging task because "embedded systems rarely have enough memory and processing power to run Web services" [6].

V. OSGI SERVICE PLATFORM

The OSGi Service Platform [8] is a Java platform that has arisen in the context of embedded systems. The platform is freely available and constantly developed by the OSGi Alliance [8].

There are several commercial and non-commercial implementations of the OSGi Service Platform, including Eclipse Equinox [30], Apache Felix [31], Knopflerfish [32], and ProSyst's mBedded server [33]. Well-known applications that are based on the platform include the Eclipse IDE and Apache Service Mix [34].

The core of the OSGi Service Platform is the OSGi Framework. This framework simplifies the development and deployment of extensible applications also called *bundles*, by decoupling the bundle's specification from its implementation. This means that a bundle is accessed by the framework through an interface, which is by definition separate from the bundle's implementation. This separation enables changing the bundle's implementation without changing the environment and other bundles.

The OSGi Framework makes it possible to run multiple applications simultaneously within a single Java Virtual Machine (JVM), by dividing applications into bundles that can be loaded at runtime and also removed. For communication within the JVM, the framework provides a service registry to register services, so that services can be found and used by other bundles.

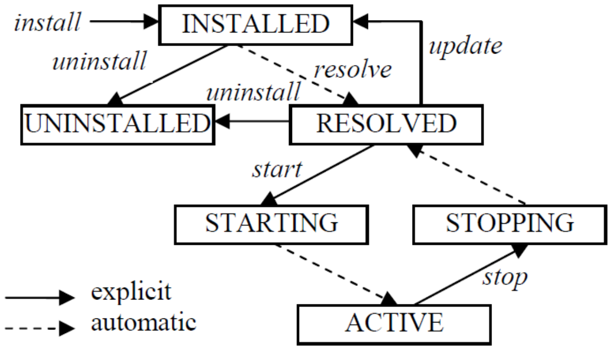


Figure 2. Life cycle of bundles in the OSGi Framework [9]

A bundle has a well-defined life cycle (see Figure 2) and its own context also called a class loader. It can be in one of the following states:

1. *Installed.* The bundle has been installed. After this, the bundle will be moved into the resolved state.
2. *Resolved.* All classes that the bundle needs have been loaded. This state indicates that the bundle is either ready to be started or has stopped.
3. *Starting.* The bundle is being started. After this, the bundle will be moved into the active state.
4. *Active.* The bundle has been activated and is running. The bundle's functionality is provided and its services are exposed to other bundles registered in the service registry.
5. *Stopping.* The bundle is being stopped. After this, the bundle will be moved into the resolved state.
6. *Uninstalled.* The bundle has been uninstalled.

VI. EXAMPLE SCENARIOS

To identify the requirements for the OSGi Service Platform in the vehicle domain, we considered the following examples of using Web services in embedded systems:

1. Car tracking service (see Figure 3a).
2. Advertising service (see Figure 3b).

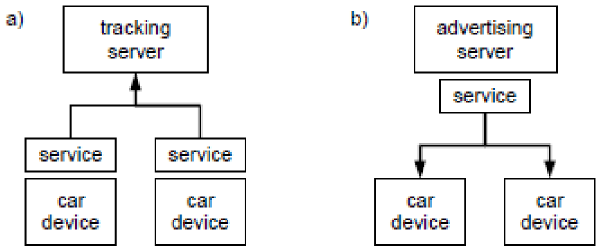


Figure 3. Example scenarios: (a) car tracking service; and (b) advertising service [5]

These examples were derived from vehicle manufacture opinions about the future direction of using Web services in embedded systems.

A. Car Tracking Service

The car tracking service will be used by the car rental company to get the car's position. Installation of this service will be initiated either by the driver or by the car rental company.

B. Advertising Service

The advertising service will be used by the driver (e.g., waiting in a traffic jam) to get advertising information. This service will be installed by the local advertising server just in time as the car enters a local hot-spot network.

VII. Requirements of Example Scenarios

We identified the following requirements for the OSGi Service Platform in the vehicle domain:

1. Dynamic availability.
2. Versioning.
3. Persistence.
4. Composition.
5. Remote management.
6. Platform independence.
7. Security.
8. Isolation.
9. Communication.
10. Distribution.

These requirements were driven by the example scenarios.

A. Dynamic Availability

Since embedded systems have small memory footprints, it is important to keep the memory footprint at runtime as low as possible. Therefore, it should be possible to install services on a running system only when they are really needed and uninstall them afterwards (e.g., when they are no longer needed) without requiring the system to be restarted, as this would also affect other services and temporarily stop them from running. In addition, since services may change over time, it should be possible to update them at runtime. However, only the smallest possible set of services should be affected by that update.

B. Versioning

Since services can be updated at runtime, it should be possible to reflect that update as new versions of services (e.g., by assigning version numbers to services).

C. Persistence

It should be possible to make services persistent. For example, once installed, the car tracking service can be saved for later reuse. This is by contrast to the advertising service that will usually be removed just in time as the car leaves the local hot-spot network.

D. Composition

An application can be composed of multiple services in order for the running system to install services only when they are really needed. Again, this helps to keep the memory footprint at runtime as low as possible. Therefore, it should be possible to deploy a composite application. However, the order in which services will be installed (e.g., to add new functionality to the running system) should not be fixed.

E. Remote Management

Since services can be installed, uninstalled and updated by external systems, it should be possible to manage the life cycle of services from the outside world.

F. Platform Independence

Since services can be installed by external systems, they cannot know in advance all platforms on which they will run. Therefore, it should be possible to run services unchanged on multiple platforms (including different hardware platforms and operating systems).

G. Security

Security is important because networks represent a potential avenue of attacks to any embedded system connected to them. Since networks allow embedded systems to communicate with external systems, networks can potentially serve as a way to break into embedded systems, enabling malicious (or buggy) services to steal resources or data. As a consequence, communication over a network raises many security issues. These issues become especially important in distributed environments, where code is downloaded across networks but executed locally as is done, e.g., with the advertising service. Because the advertising service is automatically downloaded when the car enters a local hot-spot network, it is likely that the driver will encounter code from untrusted sources.

Without security, communication of embedded systems with external systems would be a convenient way to distribute malicious services. Therefore, this communication will require the use of security techniques and technologies, from firewalls to cryptography and identity certification that are necessary to prevent services to be downloaded from untrusted sources and to prevent unauthorized remote life cycle management as well as unauthorized access to resources and data. For example, an embedded system can store the phone book from the driver's cell phone. This information should be protected from access by the advertising service (e.g., to ensure that the information will not be accidentally or maliciously modified).

H. *Isolation*

Different services such as the car tracking service and the advertising service can be deployed on the same embedded system. Therefore, it should be possible to prevent services from interfering each other (e.g., by isolating them).

I. *Communication*

While some services can live in isolation, others may need to work together to combine their functionality. Therefore, it should be possible to enable communication between services. For example, the car tracking service can communicate with the GPS service to get the car's position.

The main challenge here is to preserve security of information from, to, and within embedded systems. For example, the car's position can be transmitted over an insecure network such as the Internet. Therefore, it should be possible to encrypt this information.

J. *Distribution*

Services can be distributed across a network for the purpose of high availability, performance, or just due to their popularity. Therefore, it should be possible to deploy services in distributed environments, which usually involve versioning and communication.

VIII. EVALUATING THE OSGi SERVICE PLATFORM

We evaluated the OSGi Service Platform against the requirements of the example scenarios. Table I summarizes the results of our evaluation.

TABLE I. SUMMARY OF EVALUATION RESULTS. 'YES' – REQUIREMENT IS FULLY MET. 'YES/NO' – REQUIREMENT IS PARTIALLY MET. 'NO' – REQUIREMENT IS NOT MET AT ALL

Requirement	**Is Requirement Met?**
Dynamic availability	Yes
Versioning	Yes
Persistence	Yes
Composition	Yes
Remote management	Yes
Platform independence	Yes
Security	Yes/No
Isolation	Yes
Communication	Yes/No
Distribution	Yes/No

A. *Dynamic Availability*

Applications represented as bundles for deployment can be installed, uninstalled and updated dynamically (i.e., at runtime) without requiring a restart of the OSGi Framework.

B. *Versioning*

Bundles export services by registering them in the service registry. During this registration, additional information (including version numbers) can be assigned to the services.

The OSGi Framework goes even further in supporting versioning semantics. Versions can be assigned to export packages as well.

C. *Persistence*

Bundles are stored in the persistent storage of the OSGi Framework and remain there until they are uninstalled. Whenever the framework is restarted, bundles will be set to the same state they had just before the framework shut down.

D. *Composition*

An application can be represented as a single bundle. But it can also be composed of multiple bundles.

E. *Remote Management*

Bundles can be installed, uninstalled and updated remotely (i.e., from the outside world) without requiring a restart of the OSGi Framework.

F. *Platform independence*

The OSGi Framework is built on top of a JVM. Therefore, bundles can run on any platform that hosts the JVM.

G. *Security*

The OSGi Framework is focused on protecting embedded systems from code downloaded across a network from untrusted sources. When a bundle requests access to a particular resource, the framework grants the bundle access to that resource if and only if such access is a privilege associated with that bundle. Access control is based on digital signing, which authenticates the signer and ensures that a bundle's content is not modified after it has been digitally signed by the principal. Digital signing is based on a public key cryptography.

Not only are privileges granted to code and signers, but they are also granted to principals on whose behalf code is being executed. For example, a bundle can be granted the permission to manage the life cycle of other bundles that are digitally signed by the principal.

Although the OSGi Framework imposes strict access control on what code can and cannot do, the framework is not completely protected against damages from interference of malicious bundles. For example, malicious bundles can modify shared objects such as static variables, interned strings and `java.lang.Class` instances. This modification can affect other bundles running in the same JVM. Malicious bundles can even perform denial of service attacks against resources such as memory and a processor [12].

H. *Isolation*

Bundles are isolated from each other by class loaders.

I. *Communication*

Bundles can communicate with each other if they are loaded by the same class loader. When they are loaded by different class loaders, their communication is possible using a shared parent class loader only. However, this class loader is not part of the standard JVM and therefore requires an additional custom solution.

J. *Distribution*

To reduce the memory footprint at runtime (which is important for embedded systems) and to help avoid library redundancies, all bundles run in the same JVM. Thus, they can communicate with each other within a local framework only.

The OSGi Framework supports “rudimentary” distribution through Universal Plug and Play (UPnP) [17]. UPnP is a collection of networking protocols – e.g., TCP, UDP, IP, and HTTP – that allows networked devices to automatically communicate with each another. In particular, once a device is plugged into a network, it can access other devices connected to the network whereas other devices can access it.

IX. Related Work

In our previous work [21], we evaluated the Java Platform against the requirements of the example scenarios. The OSGi Service Platform is based on the Java Platform. But it addresses some of the weaknesses of the Java Platform; e.g., dynamic availability [7] (i.e., the ability of services to come and go at any point in time), versioning, persistence, and security. Thus, the OSGi Service Platform is more advanced than the Java Platform.

The OSGi Alliance has the Vehicle Expert Group, which aims to gather vehicle-specific requirements. These requirements are then used to tailor and extend the OSGi Service Platform. The OSGi Framework remains unchanged to provide upward compatibility.

X. Conclusion and Future Work

The OSGi Service Platform is a promising platform for future in-vehicle embedded systems and its concepts (such as bundles, class loaders and a JVM) help to meet many of the requirements of the example scenarios, viz., the car tracking service and the advertising service. However, limited support of security and distribution can be a severe hindrance for the use of the OSGi Service Platform in future in-vehicle embedded systems.

A. *Security*

The OSGi Framework has 25 security holes [23]. While 17 of them are due to weaknesses of the framework itself, other 8 are due to weaknesses of the JVM and the isolation mechanism provided by class loaders. However, this problem is being vanished over the years as a number of efforts – e.g., KaffeOS [24], Luna [25], and I-JVM [26] – have been made to patch the security holes that the JVM and class loaders leave open.

Since more and more in-vehicle embedded systems will communicate with external systems through an insecure network such as the Internet, security becomes important. However, proven solutions such as SSL/TLS for transport security and even recently issued standard such as WS-Security [22] for Web services require a lot of processing power and can disrupt the operation of embedded systems. Moreover, security increases the delay, jitter and deviation time [13].

B. *Distribution*

Limited support of distribution also becomes a less severe problem as a number of efforts – e.g., R-OSGi [14], Distributed OSGi [17], IBM Lotus Expeditor [27], Eclipse Communication Framework [15], Newton Framework [28], and Apache CXF [16] – have been made to add the distributed capability to the OSGi Framework, thus enabling bundles running in one framework to communicate with bundles running in another, potentially remote, framework.

C. *Future Work*

In the future, we will evaluate another Java platform such as JAIN SLEE [29] against the requirements of the example scenarios. This evaluation will help to extend JAIN SLEE to be applicable to future in-vehicle embedded systems.

Acknowledgment

Irina Astrova’s and Ahto Kalja’s work was supported by the Estonian Centre of Excellence in Computer Science (EXCS) funded mainly by the European Regional Development Fund (ERDF).

References

[1] G. Machado, F. Siqueira, R. Mittmann, and C. Vieira e Vieira. Embedded Systems Integration Using Web Services, Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, April 2006.

[2] M. Barr. Embedded Systems Glossary, last accessed: June 2010, http://www.netrino.com/Publications/Glossary

[3] Automotive Industry, last accessed: June 2010, www.windriver.com/solutions/automotive

[4] Automotive Embedded Systems Handbook, eds. N. Navet and F. Simonot-Lion, CRC Press, 2009

[5] R. Roelofsen, D. Bosschaert, V. Ahlers, A. Koschel, and I. Astrova. Think Large, Act Small: An Approach to Web Services for Embedded Systems Based on the OSGi Framework, Lect. Notes in Business Inform. Process. 53, eds. by J.-H. Morin, J. Ralyté, and M. Snene, Springer, Berlin, 2010, pp. 239-253

[6] M. Barr and A. Massa. Programming Embedded Systems, 2nd edition. O’Reilly, CA, USA, 2007

[7] H. Cervantes and R. Hall. Automating Service Dependency Management in a Service-Oriented Component Model, Proceedings of the 6th Workshop on Component-Based Software Engineering, May 2003

[8] OSGi Alliance: OSGi – The Dynamic Module System for Java, last accessed: June 2010, http://www.osgi.org

[9] M. Persson. Resource and Service Registration and Lookup in Cooperating Embedded Systems, acc. 6/2010, http://www2.hh.se/staff/tola/cooperating_embedded_systems/papers/magnus_persson_final.pdf

[10] World Wide Web Consortium: Web Services Activity, last accessed: June 2010, http://www.w3.org/2002/ws

[11] H. Kopetz. The time-triggered architecture, Proceedings of the 1st International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC), Kyoto, Japan, April 1998, pp. 22-31

[12] N. Geoffray, G. Thomas, B. Folliot, and C. Clement. Towards a New Isolation Abstraction for OSGi, Proceedings of the 1st Workshop on Isolation and Integration in Embedded Systems, 2008

[13] M. Shopov, H. Matev, and G. Spasov. Evaluation of Web Services Implementation for ARM-based Embedded System, Proceedings of ELECTRONICS'07, Sozopol, Bulgaria, September 2007, pp. 79-84, ISBN: 1313-1842

[14] Swiss Federal Institute of Technology (ETH) Zurich: R-OSGi pages, last accessed: June 2010, http://r-osgi.sourceforge.net

[15] Eclipse Foundation. Eclipse Communication Framework, last accessed: June 2010, http://www.eclipse.org/ecf

[16] Apache Software Foundation: CXF pages, last accessed: June 2010, http://cxf.apache.org

[17] R. Santoso. Initial Idea: Distributed OSGi Through Web Services, last accessed: June 2010, http://www.dosgi.com , http://www.dosgi.com/index/39-distributed-osgi-webservices-articles-category/55-initial-idea-distributed-osgi-through-web-services.pdf

[18] T. Honkanen. OSGi — Open Services Gateway initiative, last accessed: June 2010, http://www.automationit.hut.fi/julkaisut/documents/seminars/sem-s01/honkanen.pdf

[19] K. Hackbarth. OSGi — Service-Delivery-Platform for Car Telematics and Infotainment Systems, Advanced Microsystems for Automotive Applications, Springer, pp. 497 – 507, 2003

[20] H. Kopetz. The time-triggered approach to real-time system design. In Predictably Dependable Computing Systems, eds. B. Randell, J.-C. Laprie,H.Kopetz, B. Littlewood. Springer-Verlag, New York, 1995

[21] R. Roelofsen, A. Koschel, and I. Astrova. Evaluation of Life Cycle Functionality of Java Platform, Proceedings of the 14th WSEAS International Conference on COMPUTERS, Corfu, Greece, July 2010, pp. 69-74

[22] Web Services Interoperability Organization. Basic Security Profile Version 1.0. last accessed: June 2010, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

[23] P. Parrend and S. Frenot. Security benchmarks of OSGi platforms: Toward hardened OSGi. Software: Practice and Experience, 39(5):471-499, April 2009

[24] G. Back, W. Hsieh, and J. Lepreau. Processes in KaffeOS: Isolation, resource management and sharing in Java. Proceedings of the Syposium on Operating Systems Design and Implementation, San Diego, USA, October 2000

[25] C. Hawblitzel and T. Eicken. Luna: A flexible Java protection system, SIGOPS Operating Systems Review, 36(SI):391-403, 2002

[26] N. Geoffray, G. Thomas, G. Muller, P. Parrend, S. Frénot and B. Folliot. I-JVM: a Java Virtual Machine for Component Isolation in OSGi, Proceedings of 39th IEEE/IFIP Conference on Dependable Systems and Networks (DSN), Lisbon, Portugal, 2009

[27] IBM, Lotus Expeditor – Product and DeveloperWorks pages, last accessed: June 2010, http://www-01.ibm.com/software/lotus/products/expeditor/, http://www.ibm.com/developerworks/lotus/products/expeditor/

[28] Paremus Limited. Newton 1.5 Developer Guide, last accessed: June 2010, http://newton.codecauldron.org, http://newton.codecauldron.org/site/ newton-1.5-DEV-developer-guide.pdf

[29] JCP-JSR204, JSR 240: JAIN SLEE (JSLEE) v1.1 Spec., last accessed: June 2010, http://www.jcp.org/en/jsr/detail?id=240

[30] J. McAffer, P. VanderLei, and S. Archer: OSGi and Equinox: Creating Highly Modular Java Systems, Addison-Wesley, 2010

[31] Apache Community, Felix – OSGi framework, last accessed: June 2010, http://felix.apache.org/site/index.html

[32] MakeWave, Knopflerfish OSGi, last accessed: June 2010, http://www.knopflerfish.org/

[33] ProSyst, mBS Mobile SDK, last accessed: June 2010, http://www.prosyst.com, http://www.prosyst.com/index.php/de/html/content/64/mBS-Mobile-SDK/

[34] Apache Community, ServiceMix 4.2, last accessed: June 2010, http://servicemix.apache.org/2010/04/27/servicemix-420-released.html, http://servicemix.apache.org/home.html

Reliable Authentication and Anti-replay Security Protocol for Wireless Sensor Networks

Laura Gheorghe, Răzvan Rughiniş, Răzvan Deaconescu, Nicolae Ţăpuş
Politehnica University of Bucharest,
Bucharest, Romania
{laura.gheorghe, razvan.rughinis, razvan.deaconescu, ntapus}@cs.pub.ro

Abstract—**Wireless Sensor Network provide monitoring services such as environmental, military and medical monitoring. Sensor networks are often deployed in hostile environments and are vulnerable to attacks and failures. Security need to be implemented in order to prevent unauthorized access to the network and malicious attacks. The Authentication and Anti-replay Security Protocol is a combination of two lightweight mechanisms that ensure authentication, anti-replay and intrusion detection: the "Last Hash" method, and the authentication handshake. This paper introduces three reliability enhancements to the first version of the protocol: acknowledgements, re-authentication and a current hash computed with a different key to ensure integrity. Reliable AASP was implemented in TinyOS and tested using TOSSIM. Simulations indicate that Reliable AASP is able to provide a reliable authentication connection between any two communicating nodes, and it meets the critical security requirements: integrity, authentication and freshness.**

Keywords-wireless sensor networks, security, reliability, integrity

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of a large number of sensor devices characterized by reduced dimension, low cost and low power, which are able to organize themselves into a network by communicating through a wireless medium, collaborating in order to accomplish a common task [1].

WSNs provide monitoring services in different areas, such as industrial, military, public safety, automotive, agriculture, localization, seismic, medical, commercial and emergency situations. Some of the most interesting applications are detecting the enemy units during military monitoring, person locator, disaster detection, and health condition monitoring [2].

Because WSNs have the advantage of being deployable in inhospitable fields, such as battlefields, outer space and deep waters, they are highly recommended in military applications, environmental monitoring, security and surveillance, industrial process control and health care applications [3].

The network design objectives and requirements include: reduced dimension, low cost and low power, scalability, adaptability, reliability, fault-tolerance, security, self-configurability and QoS. Fault tolerance includes capacities for self-testing, self-calibrating, self-repairing and self-recovering [4].

Securing WSNs is essential when they are used in critical applications such as battlefield surveillance and homeland security. This is a challenging task because of several limitations deriving from the wireless channels, resource constraints, and hostile environments.

Because the wireless medium is open, anyone can intercept traffic and inject fake data packets if they have the radio interface configured on the same frequency band.

Traditional security mechanisms cannot be applied to sensor networks because of their severe resource constrains. Sensor nodes do not have the computational capacity to manage public key cryptography or other complex protocols. For this reason, the best choice for WSNs is to use symmetric keys, though they must be used with precaution in order to avoid performance degradation.

Another challenge regarding security in WSNs is their deployment in hostile environments and the fact that they must work unattended. They are vulnerable to physical attacks, such as tampering and node capturing.

The rest of the paper is structured as follows: Section II presents the related work, Section III contains the basic protocol design, Section IV describes the basic protocol issues, Section V presents the reliability improvements brought to the basic protocol, Section VI describes the implementation of the protocol, Section VII presents the experimental results, in Section VIII we discuss the potential problems and solutions, and Section IX presents the conclusions of the paper and some of the future work.

II. RELATED WORKS

Various security solutions were developed for WSNs, and the most important are SPINS, LEAP, TinySec, and SM [5].

SPINS is a set of security protocols that consist of two building blocks: SNEP and µTESLA [6]. Both were implemented to run on top of TinyOS [7].

SNEP is used to provide authentication, integrity, confidentiality and freshness, and µTESLA provides authenticated broadcasts [5]. Authentication and integrity is provided by Messsage Authentication Code (MAC), confidentiality through encryption, and freshness through nonce. µTESLA emulates asymmetry through the delayed disclosure of symmetric keys.

ISBN: 978-1-61208-105-2

LEAP (Localized Encryption and Authentication Protocol) was designed by Zhu et al. in 2003 and is a key management protocol for WSNs [8].

LEAP was implemented as LEAP+ in TinyOS and first used on Berkley Mica2 motes [9]. LEAP uses four different types of keys in order to provide an adequate level of security to the various messages exchanged in the WSN [5].

TinySec was designed by Karlof et al. in 2004 and is the replacement for SNEP [10]. TinySec is a link layer security architecture that was included in the TinyOS release. It provides authentication and integrity, confidentiality and semantic security. Semantic security is achieved through Initialization Vector (IV) [5].

Security Manager (SM) was proposed by Heo and Hong in 2006 and is a new security method of authenticated key agreement [11]. It uses Public Key Infrastructure (PKI) and Elliptic Curve Cryptography in order to assure security [5].

The existent security solutions are very complex because they aim to meet all the major security requirements: authentication, integrity, confidentiality and freshness. We consider that confidentiality cannot be obtained without a major computational overhead that is not feasible for Wireless Sensor Networks. However, if the application is critical and requires confidentiality, a complex security solution such as SNEP must be used.

Energy consumption is the most critical problem in Wireless Sensor Networks. We aim at developing a lightweight security protocol that is focused only on authentication, integrity and anti-replay, which mitigate the most important threats to sensor networks: packet injection and packet altering. Our protocol introduces a small overhead because it only computes a hash function; therefore it is a lightweight security protocol designed to minimize the energy consumption.

III. AASP

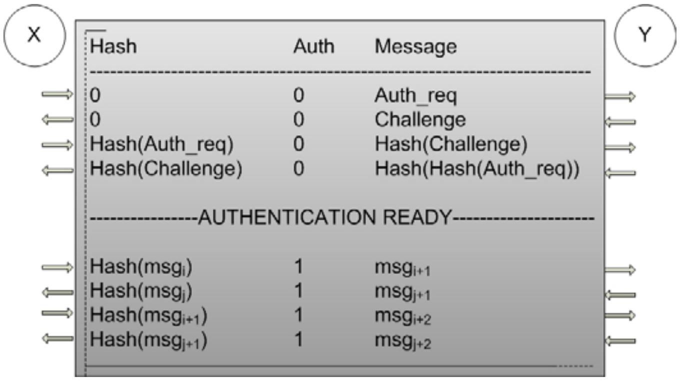
We developed a lightweight security protocol, called Authentication and Anti-replay Security Protocol (AASP), which is able to provide authentication, anti-replay and intrusion prevention [12].

The protocol uses a globally shared key to compute the Message Authentication Code (MAC) in order to provide authentication between nodes.

Anti-replay requirement is assured by the "Last MAC Method", in which the MAC of the last sent message with the same source and destination is sent along with the current message.

An authentication connection must be established between two nodes that want to communicate, using an authentication handshake, as represented in Figure 1.

AASP provides intrusion prevention because it prevents malicious nodes from communicating with the nodes inside the network. Intrusion prevention is achieved using the handshake authentication and the shared key.



Hash	Auth	Message
0	0	Auth_req
0	0	Challenge
Hash(Auth_req)	0	Hash(Challenge)
Hash(Challenge)	0	Hash(Hash(Auth_req))
---------------AUTHENTICATION READY---------------------		
Hash(msg_i)	1	msg_{i+1}
Hash(msg_j)	1	msg_{j+1}
Hash(msg_{i+1})	1	msg_{i+2}
Hash(msg_{j+1})	1	msg_{j+2}

Figure 1. AASP authentication handshake

The security protocol was implemented in TinyOS, an operating system developed especially for Wireless Sensor Networks [7]. AASP was tested using TOSSIM, which is a discrete event simulator for TinyOS sensor networks [13].

IV. AASP ISSUES

As we have stated in [12], several problems can appear in a real-life deployment of this security protocol. This paper aims at finding the most efficient solutions to those problems.

A. *Altered packets in multi-hop networks*

In a multi-hop network, the packet must be routed in order to reach the destination. However, an intermediate node could maliciously alter the payload of the packet, leaving the Hash field unmodified. This would lead to the destination accepting an altered message, which is a serious vulnerability. Our solution is to include another field in the header that will include the MAC of the current message. Therefore, an intermediate node is not able to modify the payload without knowing the secret key, and if it does, the message is dropped at the destination.

B. *Packet loss*

The problem that has the highest probability to appear is the loss of packets. If a single packet is lost, the hash becomes de-synchronized and the subsequent packets are dropped at the destination with an "Incorrect Hash" alert. A similar problem appears in the SPINS suite of security protocols, where the anti-replay protection is provided by incrementing a counter for every sent and received packet. Packet loss causes the disruption of the counter synchronization. A solution would be for the destination to acknowledge packets as they are received and the source to wait for an acknowledgement before sending the next packet. We will evaluate the impact of such a solution as regards energy consumption.

C. *Re-authentication after reboot*

Another problem may appear in the eventuality that one node, designated as X, reboots during an authentication connection between two nodes (X and Y). After rebooting, X could try to re-establish the connection, but Y would reject all packets with the message: "Node already authenticated tries to re-authenticate". This would happen because Y was not announced in any way that node X closed the connection, and continued to act as if it had an authentication connection with it. In addition, we also need a method of terminating an idle connection. The solution for both problems would be to permit the connection to expire after a period of time in which no messages are sent. After the old connection is closed by both sides, a new connection can be established. This method would introduce a certain delay in communication, and we must evaluate the impact of this solution taking into account the probability for a sensor to reboot unexpectedly.

Another solution would be to permit the operation of re-authentication, in which one node can initiate an authentication handshake even if the other node knows it has a connection already established. However, this practice is dangerous because any attacker can interpose between two nodes and tear down their connection just by sending an “Authentication Request”.

V. RELIABLE AASP DESIGN

Four main levels of fault tolerance can be implemented for WSNs that correspond to the following layers: hardware and software, network communication and application layers [1]. We are interested in the network communication layer fault tolerance.

In the following sections, we will present three methods that have been integrated into AASP in order to enhance the reliability of the security protocol and the network communication.

A. Packet loss

As we have observed in simulations, packet loss is a serious problem. It desynchronizes the anti-replay mechanism because the destination expects another value of the last hash. Subsequent packets are dropped by the destination when a single packet is lost on the way.

The standard procedure when dealing with packet loss relies on the use of acknowledgements. When a packet is received, the destination node is responsible for sending back a packet containing the sequence number of the acknowledged message, and an acknowledgement flag. This way, the source node knows when it is safe to send the next packet.

The source node waits for the acknowledgement for a predefined period of time, after which it resends the packet that was not acknowledged. Not receiving the acknowledgement before timeout, could indicate that either the connection with the destination was closed, or the acknowledgement packet was lost on the way. The connection could be closed when the destination is dead or when it had restarted itself. The source node is not able to differentiate between these two situations, and will treat them in the same way, by resending the lost packet.

During the handshake process, the packets are not acknowledged by separate packets, but because an actual conversation is taking place, the acknowledgements are piggybacked in the reply messages. This cannot be done after the authentication connection is established because, in most cases, the conversation consists in packets sent from the child to the parent in the network hierarchy. A hierarchy is formed so that the collected information would reach the base station. In this case, the parent would have to reply with separate acknowledgement packets.

This method provides reliability to AASP, because it handles the loss of packets through the traditional method of acknowledgements.

B. Re-authentication

Re-authentication is the solution for both desynchronized and closed connections. Desynchronization could occur when acknowledgements are not used and packets are lost, or if they are used and the information about the acknowledged packets is corrupted on the source or destination node. As we have previously stated, a connection is considered closed when either node is dead or has restarted.

The connection must be terminated after a specific amount of time in which no message is received from the node on the other side of the connection.

After each received message, a timer is set to fire once after a specific amount of time that is to be determined experimentally. If another message is received during this period of time, the timer is reset. Otherwise, the authentication connection will be terminated by erasing all authentication information regarding the connection with that specific node for which the timer has fired. In addition, a connection termination message will be sent to that node, in order to announce it of the connection tear down. The neighbor could be alive, in the situation of non synchronization or restarted node. In the first situation, the neighbor node will receive the termination message and erase all information about that connection. In the case of a restarted node, all information is already to its default settings and nothing should be done.

After authentication data has been erased at both nodes, a new authentication connection can be established if either node initiates an authentication handshake.

C. Altered packets

We can prevent messages from being altered by sending the Message Authentication Code (MAC) of the current message. The destination will compute the hash value of the received message and compare it with the MAC in the header. If the values are different, the packet will be dropped, and an “Altered packet received” message will be generated.

If we want to be sure that the attacker cannot replay older packets, we must have two secret keys in the network: one for the last MAC, and one for the current MAC. This way, the attacker cannot use the current MAC from an intercepted packet as the last MAC in a malicious packet.

We evaluate that the overhead introduced by keeping two secret keys instead of one is insignificant comparable to the advantage of messages being protected from alteration in multi-hop networks.

VI. IMPLEMENTING RELIABLE AASP

A. *Packet loss*

Acknowledgements are implemented in AASP using a new field in the authentication header, which contains the value "1" if the packet is an acknowledgement and the value "0" if it is a data packet. We call this field the ack flag.

During the authentication handshake, all packets except for the "Authentication Request" have the ack flag set, because each reply acknowledges the previous message exchanged between the two nodes.

After the authentication connection has been established, because a two-way conversation is not expected, acknowledgements must be sent as stand alone packets. These packets contain the sequence number of the packet that is acknowledged, placed in the payload field, and the ack flag that is set. The sequence number will be represented as a new field in the AASP packets. The sequence values start from value "1" after the authentication connection is established.

Each node stores two values for every established connection: the last received acknowledge (last_recv_ack) and the last sent acknowledge (last_sent_ack). After sending an ack, the value of last_sent_ack is updated to the value of the sequence number of the acknowledged packet. When receiving an ack, the value of last_recv_ack is updated to the sequence number received in the payload.

In a hierarchical network topology, when a child sends a message to its parent with a sequence number of x, it starts a timer that expires after a configured period of time. During this period, if the ack is received, the timer is stopped, and the packet with sequence x+1 is sent. However, if no ack is received, the packet with sequence x is resent.

This method has been implemented in the Authentication layer, by maintaining two arrays of integers, representing the last_sent_ack and last_recv_ack for each established connection. For example, last_recv_ack[i] is the sequence number of the last acknowledged packet sent to the neighbor with the identifier i.

An array of timer interfaces is used through parameterized interfaces, an important feature of nesC language. The timer is set to fire once by using *Timer.startOneShot* command and by specifying the period of time as an argument. In the *Timer.fired* event, the packet must be resent to the destination, therefore the packet must be stored locally until an acknowledgement is received. However, if an acknowledgement is received using *Receive.receive* event, the timer for that specific destination is stopped. The next message can now be sent by the main application. The main application execution must be delayed until the next message can be sent.

B. *Re-authentication*

Re-authentication is possible only after a previous connection is terminated, or both sensor nodes are restarted. The state of a connection is maintained at the Authentication layer by three arrays: auth, req and rd. For a neighbor i, auth[i] is "1" if an authentication connection has been established with that neighbor and "0" if not, req[i] is "1" if an "Authentication Request" has been sent to that neighbor and "0" if not, and rd[i] keeps the value of the challenge used for that neighbor.

At the MAC (Message Authentication Code) layer, the state is kept using two arrays of hashes, both called last_hash, where last_hash[i] is the hash of the last message sent to or received from that neighbor. All this authentication data should be erased during connection termination.

An array of timer interfaces is maintained through parameterized interfaces, as with acknowledge timers. When a packet is received at the Authentication layer, the timer associated with the source node i, is set to fire once after a configured period of time. In the *Timer.fired* event, the information in Authentication layer (auth[i], req[i] and rd[i]) is set to "0". A message is sent to the sensor node i in which auth field is set to a specific termination code called TERM. The MAC layer recognizes the code and erases all authentication data from its level. If the destination is alive and has not restarted, it will reset all authentication data from MAC and Authentication layer. If the destination has restarted, it will ignore the message.

The authentication handshake will be further initiated by the node that transmits data, in most situations, the child in the hierarchy.

C. *Altered packets*

The method was implemented by adding a new field in the authentication header containing the MAC of the current message that is computed using a different key.

The hash of the payload is computed and put in the current MAC field of the packet, in MacLayerSenderP component, before sending the packet.

In MacLayerReceiverP component, the first operation when receiving a packet is to check if the current MAC contained in the received packet is equal to the hash of the payload. If not, the packet is dropped at this level, or sent to the Authentication Layer with an error code in Auth field. We reserved the code 3 for "Altered packet". In AuthenticationLayerC component, if the Auth field is equal to 3, the packet is dropped, and an "Altered packet received" message is generated.

VII. TESTING RELIABLE AASP

The protocol has been implemented in TinyOS; therefore TOSSIM is the best solution for simulating AASP [13].

We use a section of a real topology to test the first scenario, in which node 3 sends messages to node 1. This scenario is used to test the functionality of acknowledgements, connection termination and re-authentication. Therefore, we did not include the hash of the current message in the simulation output.

Node 3 and 1 communicate using Reliable AASP; the first one sends data packets and the later one replies with acknowledgements. When the packets are lost, no acknowledgement is received and the packet is retransmitted.

When no packet is received by node 3 in the idle period of time, the connection between the nodes must be terminated.

We consider that the authentication handshake takes place and ends successfully with an established authentication connection. After that, in Figure 2 we can observe that node 3 starts sending packets to node 1 periodically. Packets are sent with auth field equal to "1" and ack field equal to "0". The seq field contains the sequence number of the packet. Node 3 responds with an acknowledge packet that has the ack field set to "1" and the payload containing the sequence number of the acknowledged packet. The figure presents two packets that are sent, received and acknowledged.

```
(3): AuthLayer: Packet sent [seq=1 hash=4442 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet received [seq=1 hash=4442 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet sent [msg=1 hash=24000 auth=1 ack=1 (1->3)]
(3): AuthLayer: Packet received [msg=1 hash=24000 auth=1 ack=1 (1->3)]
(3): AuthLayer: Packet sent [seq=2 hash=123 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet received [seq=2 hash=123 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet sent [msg=2 hash=123 auth=1 ack=1 (1->3)]
(3): AuthLayer: Packet received [msg=2 hash=123 auth=1 ack=1 (1->3)]
```

Figure 2. Normal flow of packets

In Figure 3, the message with sequence number equal to "7" does not reach its destination. The timer set on node 3 expires after the ack period and the node resends the message. After the idle period, the connection timer at the destination expires and the connection is terminated. Node 1 also sends a reset message containing a connection termination code, announcing the destination about the connection tear down. The code used in this implementation is "4" and is placed in the auth field. In this moment both nodes have the authentication data erased from both AASP layers.

```
(3): AuthLayer: Packet sent [seq=7 hash=97 auth=1 ack=0 (3->1)]
(3): AuthLayer: Timeout ack node 1
(3): AuthLayer: Packet sent [seq=7 hash=97 auth=1 ack=0 (3->1)]
(3): AuthLayer: Timeout ack node 1
(3): AuthLayer: Packet sent [seq=7 hash=97 auth=1 ack=0 (3->1)]
(3): AuthLayer: Timeout ack node 1
(1): AuthLayer: Timeout connection node 3
(1): AuthLayer: Reset packet sent [seq=0 hash=0 auth=4 ack=0 (1->3)]
(3): AuthLayer: Packet received [seq=0 hash=0 auth=4 ack=0 (1->3)]
```

Figure 3. Packet loss causing connection timeout

In Figure 4, we can observe the authentication handshake being re-initiated by node 3 and terminating with another authentication connection established.

```
(3): AuthLayer: Packet sent seq=124 hash=0 auth=0.
[..]
(1): AuthLayer: Managed to authenticate myself to node 3
(3): AuthLayer: Managed to authenticate myself to node 1
(3): AuthLayer: Packet sent [seq=7 hash=4442 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet received [seq=7 hash=4442 auth=1 ack=0 (3->1)]
(1): AuthLayer: Packet sent [seq=7 hash=24000 auth=1 ack=1 (1->3)]
(3): AuthLayer: Packet received [seq=7 hash=24000 auth=1 ack=1 (1->3)]
```

Figure 4. Re-authentication

The packet with sequence number "7" is resent and received successfully at node 1 and acknowledged. The subsequent packets are treated in the same way.

In the second scenario, we test the ability of Reliable AASP to provide integrity, and to detect that the packet has been altered during the routing process by a malicious node, or by errors during transmission.

We use a section of a multi-hop network topology: node 3 wants to send a data packet to node 1, but cannot reach it directly, only through node 2. Therefore it will use node 2 to route data packets to the destination node. In this scenario, node 2 is a malicious node programmed by an attacker to modify the payload of the packets that it must route to destination.

The previous version of the protocol did not provide packet integrity. Therefore packets could have been modified by an intermediate node, while the destination would not have noticed. However, it would have dropped packets starting with the next packet even if they would not have been altered, only because the altered packet would desynchronize the connection.

In the current version of Reliable AASP, the packet includes the hash of the current message, called current hash, which is computed using a secret key different than the one that is used to compute the hash of the last message. This way, even if the intermediate node changes the payload, the packet will be dropped at the destination because the current hash contained in the packet would not match the hash computed at the destination. This does not desynchronize the connection, because the source node waits for an acknowledgement before sending the next message. If the message is dropped by the destination, the acknowledgement is not received, and the message is retransmitted.

In Figure 5, we consider that node 2 does not alter the message. Therefore, we can observe the normal flow of packets. We omitted the authentication handshake process, and the information about the last hash and auth fields.

```
(3): AuthLayer: Packet sent [msg=222 chash=64327  seq=1 ack=0 (3->1)]
(2): RoutingLayer: Packet received [msg=222 chash=64327  seq=1 ack=0 (3->1)]
(2): RoutingLayer: Packet sent [msg=222 chash=64327  seq=1 ack=0 (3->1)]
(1): AuthLayer: Packet received [msg=222 chash=64327  seq=1 ack=0 (3->1)]
```

Figure 5. Non-malicious routing node

The data packet is routed by node 2 and reaches node 1 without being altered. Node 1 computes the hash of the payload using the second secret key, and it verifies if the value is equal to the current hash found in the packet. The values are equal, therefore node 1 generates an acknowledgement packet and sends it to node 3 through node 2.

In Figure 6, we consider that node 2 alters the message received from node 3, and then routes it to node 1. Node 2 cannot re-compute the current hash, because it does not have the secret key. Therefore, it sends the packet with the altered payload and the old current hash. The altered data packet reaches node 1, and is inspected by the MAC layer, which detects that the computed hash is different from the one contained in the packet send it to the Authentication layer with an error code of "3" in the auth field of the packet. The Authentication layer receives the packet, generates an "Altered packet" message and drops the packet. It does not send any acknowledgements, and after the ack period has passed on node 3, it resends the data packet.

```
(3): AuthLayer: Packet sent [msg=222 chash=64327  seq=1 ack=0 (3->1)]
(2): RoutingLayer: Packet received [msg=222 chash=64327  seq=1 ack=0 (3->1)]
(2): RoutingLayer: Packet sent [msg=999 chash=64327  seq=1 ack=0 (3->1)]
(1): MacLayer: Packet received [msg=999 chash=64327  seq=1 auth=1 ack=0  (3-
>1)]
(1): AuthLayer: Packet received [msg=999 chash=64327  seq=1 auth=3 ack=0  (3-
>1)]
(1): AuthLayer: Altered packet received. Packet dropped.
(3): AuthLayer: Packet sent [msg=222 chash=64327  seq=1 ack=0 (3->1)]
[…]
```

Figure 6. Malicious routing node

However, if the attacker continues to alter the data packets, it will cause the connection to terminate, because the correct data packet would not be received by the destination. This is considered to be a Denial of Service attack.

VIII. DISCUSSION

As we have stated in the previous section, a node that acts as a router between source and destination nodes can cause a Denial of Service attack by altering all messages that must be routed. The source and destination node would wait the idle time period and close the connection. The best solution would be to modify the routing protocol in order to choose another intermediate node to route the data packets to destination, if the current one does not deliver the packets correctly. Another solution would be to send negative acknowledgements to the source of the data packets in order to reduce the waiting time of the source node before choosing another route to destination. A serious problem would appear if the malicious node is the only way to the destination, meaning that it is the only node that can reach the destination. Therefore, even if we consider that the source and destination are at least two hops away, if the last hop is the only one that can reach the destination, changing the route would not exclude that node, therefore, the communication with that specific destination will be irremediably lost.

Adequate values must be determined experimentally for acknowledge and idle periods. Both values must be computed for the specific deployed network and application. We consider that some requirements must be satisfied in order to find the appropriate values.

In WSNs, we have two types of data messages: periodic and triggered. The periodic messages contain data that is periodically collected from the environment and sent to the base station. The triggered messages are generated in emergency situations and should reach the base station as soon as possible. The idle period of AASP should be greater than the period of data collecting. Otherwise, before every data collecting moment, the connection would expire and another connection must be established introducing overhead because of the authentication handshake. Because triggered messages can be generated anytime, it is impossible to configure the idle period using predictions about these messages.

The acknowledge time period should be greater than the two-way message exchange time between two nodes multiplied by the maximum number of hops between two nodes in the sensor network. This way, if an authentication connection is established between two nodes that have the maximum number of hops between them, the acknowledgement should have time to reach the source of the data packets. Otherwise, the source would send the data packet, and the acknowledgement time period would expire before the acknowledgement would arrive and the data message will be unnecessary retransmitted. The two-way message exchange time period depends on the network traffic. This value can only be determined experimentally after the deployment of the network and application.

Therefore, idle time period depends on the application and its period of collecting and transmitting data. The acknowledge time period depends on the actual topology and the network traffic.

A problem is introduced by the high quantity of acknowledgements required in order to detect packet loss. In the actual implementation, the number of acknowledgements is equal to the number of data packets. However, other methods exist in which an ack packet is used to acknowledge a set of data packets instead of one single packet, but they would require a large number of messages to be stored at the source node before they are acknowledged.

Another problem regarding acknowledgements is that ack packets can also be lost. This would determine the source node to send the packet again. The duplicate would reach the destination and would be dropped because it would have the same sequence number. Duplicate packets are undesirable because they waste bandwidth and consume energy.

Nodes are still vulnerable to flooding with “Authentication requests”, but this attack can be detected using Storm Control Mechanism [14].

IX. CONCLUSION

Wireless Sensor Networks provide monitoring services in critical domains such as military, security and medical, a lightweight security protocol has to be used in order to secure the communication within the sensor network.

AASP is a lightweight security protocol that was designed to provide security features such as authentication, freshness and intrusion detection. In this paper, we present three reliability enhancements to AASP: acknowledgements, re-authentication and integrity hash.

Acknowledgements are used to detect the loss of packets. When a node sends a data packet, it sets a timer and waits for an acknowledgement for that data packet. If the ack does not arrive until the timer has expired, the packet is resent.

Re-authentication is needed in various situations, from desynchronized connections to restarted nodes. To proceed with the re-authentication, the previous connection must be terminated, otherwise all packets are blocked. The connection is terminated after a specific period of time in which no packet is received by the node.

The initial protocol version did not provide the integrity of the current message. The enhanced protocol version contains an integrity hash placed in the packets. This hash is a Message Authentication Code computed using a second secret key. The hash is recomputed at the destination and compared with the one received in the packet. This way, altered packets can be dropped.

Reliable AASP has been implemented in TinyOS by modifying the previous version of AASP. The protocol was tested in TOSSIM, a simulator for TinyOS applications.

Further work will consist in testing the performance of the protocol in terms of energy and bandwidth consumption. The number of messages exchanged by sensor nodes has doubled because of the acknowledgements. However, acknowledgement packets are relatively small compared to the data packets, and are necessary in order to detect the loss of packets.

Also as a future work, we wish to test our protocol on a real Wireless Sensor Network and compare our simulation results to the ones obtained on the physically deployed sensor network.

REFERENCES

[1] J. Zheng and A. Jamalipour, “Wireless Sensor Networks A Networking Perspective”, John Wiley & Sons, 2009.

[2] C.F. García-Hernández, P.H. Ibargüengoytia-González, J. García-Hernández, and J.A. Pérez-Díaz, “Wireless Sensor Networks and Applications: a Survey”, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.3, March 2007, pp. 264-273.

[3] D. Trossen and D. Pavel, “Sensor networks, wearable computing, and healthcare Applications”, IEEE Pervasive Computing, vol. 6, no. 2, Apr-June 2007, pp.58-61.

[4] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, “Fault-tolerance techniques for ad hoc sensor networks”, Proceedings of IEEE Sensors, vol. 2, June 2002, pp. 1491-1496.

[5] D. Boyle and T. Newe, “Securing Wireless Sensor Networks: Security Arhitectures”, Journal of Networks, Vol. 3, No. 1, January 208, pp. 65-77.

[6] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D. Culler, “SPINS: Security Protocols for Sensor Networks”, Wireless Networks, 2002, 8(5), pp. 521-534.

[7] P Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks”, Ambient Intelligence, 2005, pp. 115-148.

[8] S. Zhu, S. Setia, and S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks”, CCS ’03, Washington D.C., USA, 27 – 31 October 2003, New York, USA: ACM Press, pp. 62-72.

[9] S. Zhu, S. Setia, and S. Jajodia, “LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks”, ACM Transactions on Sensor Networks TOSN, 2006, 2(4), pp. 500-528.

[10] C. Karlof, N. Sastry, and D. Wagner, “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks”, Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 03 – 05 November 2004, New York, NY, USA: ACM Press, pp. 162 – 175.

[11] J. Heo and C.S. Hong, “Efficient and Authenticated Key Agreement Mechanism in Low-Rate WPAN Environment”, International Symposium on Wireless Pervasive Computing 2006, Phuket, Thailand 16 – 18 January 2006, IEEE 2006, pp. 1-5.

[12] L. Gheorghe, R. Rughiniş, R. Deaconescu, and N. Ţăpuş, “Authentication and Anti-replay Security Protocol for Wireless Sensor Networks”, The Fifth International Conference on Systems and Networks Communications, August 22-27, 2010, pp 7-13.

[13] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, In SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, 2003, pp. 126-137.

[14] R. Rughiniş and L. Gheorghe, “Storm Control Mechanism for Wireless Sensor Networks”, 9th RoEduNet IEEE International Conference, June 24-26, 2010, pp. 430-435.

Adaptive Trust Management Protocol Based on Fault Detection for Wireless Sensor Networks

Laura Gheorghe, Răzvan Rughiniş, Răzvan Deaconescu, Nicolae Ţăpuş
Politehnica University of Bucharest
Bucharest, Romania
{laura.gheorghe, razvan.rughinis, razvan.deaconescu, ntapus}@cs.pub.ro

***Abstract*—Trust management is an important issue in self-configurable and autonomous networks such as Wireless Sensor Networks. Sensor nodes need to determine if other nodes are trustworthy, in order to decide whether to cooperate with them in completing the sensing and communication tasks. Therefore, adaptive trust management assures an appropriate level of security to the critical services provided by Wireless Sensor Networks. In this paper, we present the Adaptive Trust Management Protocol for Wireless Sensor Networks, which is able to compute data trust based on fault detection. The adaptive trust management protocol operates cycles in which reputation values are computed and penalty values are exchanged periodically. A spanning tree is generated for the sensor network, after which nodes evaluate their children using the fault detection mechanism and then exchange penalty values with their neighbors. The protocol has been implemented in TinyOS and evaluated in a test scenario using TOSSIM.**

Keywords-reputation, trust, fault detection, wireless sensor networks

I. INTRODUCTION

Wireless Sensor Networks are an emerging technology, becoming a fundamental method in monitoring various environments [1]. A sensor network consists of a large number of sensor nodes that are able to perform sensing, processing and communicating tasks in a collaborative manner, in order to detect specific events that take place in the monitored environment [2].

The sensor network can be seen as a service provider for user applications. The services provided by the WSN are data collection and data delivery. A service-oriented approach fills the semantic gap between high level application requirements and the low level operations provided by the sensor network [3] [4].

Because sensor networks are used in critical applications such as battlefield surveillance, homeland security and medical monitoring, a critical task when designing a Wireless Sensor Network is to ensure security against malicious attacks and faulty nodes [5].

A sensor network can be protected against external malicious nodes through the use of authentication methods [6]. However, for internal faulty nodes, another method must be used to ensure protection against false reports. A solution against false reports relies on using a trust management scheme and enforcing a trust policy that sensor nodes must respect [7].

While the aspects presented in this paper are general and can be applied to any kind of wireless network, the special aspect specific to Wireless Sensor Network considered is the need to minimize the consumption of energy. Therefore, we aim to reduce the number of packets being sent and received and we use simple algorithms to compute the trust values.

We propose an Adaptive Trust Management Protocol (ATMP) that determines trust by computing reputation values based on fault detection techniques. The protocol operates in three phases: setup, learning and exchange phase. In the setup phase, a spanning tree is built, while in the learning phase, the local penalty value is modified on the basis of the fault detection techniques. In the exchange phase, nodes exchange reputation values, re-compute them and determine trust. The protocol is adaptive because the reputation values are modified on each cycle, according to the detected faults and to the penalty values received from neighboring nodes. The protocol is collaborative because sensor nodes interact on every cycle in order to update reputation values.

The rest of this paper is organized as follows: Section II presents the problem of false reports and the proposed solution. Section III contains definitions of “trust” and “reputation” and introduces “trust management”. Section IV describes related work. Section V introduces the Adaptive Trust Management Protocol. Section VI presents implementation details of the protocol in TinyOS. Section VII describes the test scenario and simulation results. Section VIII discusses advantages and potential problems. Section IX concludes this paper.

II. FALSE REPORTS

The main application of Wireless Sensor Networks consists in environment monitoring and event detection. However, malicious or faulty nodes can generate and send incorrect data to the base station. Incorrect data can disrupt normal data fusion, and event detection. It can trigger false alerts by generating false alert data, or it can hide important events by generating false normal data instead of alert data [8].

Attackers could physically capture and compromise a sensor node. They could maliciously inject invalid data into the network in order to disrupt normal functionality, especially event detection. Authentication and cryptographic methods are not a solution to this problem, because once they have captured a sensor node, the attackers can have access to the cryptographic information stored on the sensor and they can successfully authenticate themselves [7].

Besides malicious attacks, hardware and software faults can also cause incorrect data to be generated and sent to the base station. For example, the sensing unit or the radio can malfunction, altering packets and generating inaccurate sensor readings. This problem, also, cannot be solved by using authentication and cryptographic mechanisms.

A possible solution relies on the fact that, when an event takes place in a specific area of the sensor network, sensor nodes that are in the proximity should have similar collected data [9]. However, if a node is malicious or faulty, it can generate data sets that do not match its neighbors' data. Therefore, an incorrect data reading can be detected by comparing the data collected by sensor nodes from the same area.

In the data aggregation process, the values received from the children nodes are combined and one single value is forwarded toward the base station [7]. In order to prevent the transmission of faulty information, before the aggregation operation takes place, the node waits for all the children to send data, and after that, it checks whether proximal nodes send similar data. The incorrect values will be detected and they will not be forwarded, while the source nodes will be penalized.

III. Trust management

Trust can be defined as the level of confidence a decident has in the performance of a person or object. Trust has always played an important role in social environments, and recently it started to be considered in various kinds of networks, such as peer-to-peer, ad-hoc and sensor networks.

Trust is associated with the ability to provide the expected service. In sensor networks, trust is associated with the accuracy of event detection and undisturbed network and protocol functionality.

From a networking perspective, a node can evaluate and use trust in order to decide whether another node is uncooperative, malicious or faulty.

Trust is especially critical in networks that rely on collaborative event detection and environmental monitoring, where nodes cooperate permanently in order to provide accurate data collection that characterizes the monitored environment.

Trust management was first defined by Blaze et al. in [9]. They propose a framework for security policy, credentials and trust relationships.

In Wireless Sensor Networks, trust management is a challenging task because they are autonomous and self-configurable, without any central point of management. In such networks, trust management is a cooperative process, rather than a local node oriented process.

Three types of trust evaluation have been defined in Wireless Sensor Networks: communication trust, data trust and energy trust [10]. Communication trust consists in computing reputation values based on successful and failed transactions. Data trust is the assurance of fault tolerance and data consistency. Energy trust depends on the level of existing energy and on the threshold level needed in order to perform sensing and communication tasks.

In this paper, we focus on data trust, and we present a data management protocol that is able to enforce trust on the basis of fault detection methods, and to provide data consistency for Wireless Sensor Networks.

Adaptive Trust Management Protocol (ATMP) uses cooperative trust management and has a hierarchical view over the network. The parent nodes obtain information about their children and then exchange penalty values with their neighbors in order to compute reputation values.

IV. Related work

The most important trust management schemes suited for Wireless Sensor Networks are: Reputation-based Framework for Sensor Networks (RFSN), Agent-based Trust and Reputation Management (ATRM), Parameterized and Localized trUst management Scheme (PLUS), Group-based Trust Management Scheme (GTMS) and Trust-aware Query Processing.

A. RFSN

Reputation-based Framework for Sensor Networks (RFSN) is a trust management framework in which each node maintains a trust value for each neighbor node [12]. RFSN uses statistical and decision theory methods in order to predict the future behavior of nodes and to identify misbehaving nodes.

Trust values are computed based on reputation. Bayesian formulation is used to represent the reputation of a node, but also for updates, integration and trust evolution. A level of confidence is computed for each data reading, through consensus-based outlier detection schemes.

RFSN is not suited for sensor networks with high mobility, because in this case, the reputation values will not converge. A node must have constant neighbor interactions in order for the reputation to stabilize.

B. ATRM

Agent-based Trust and Reputation Management (ATRM) is based on mobile agents that are generated by a single trusted authority [13]. It assumes that the information carried by the agents will not be accessed or modified by the malicious nodes present in the sensor network.

The major advantage of this trust management scheme is that it takes into consideration the power and bandwidth constrains and tries to reduce communication overhead and delay.

C. PLUS

Parameterized and Localized trUst management Scheme (PLUS) is built on top of the PLUS_R routing scheme. It uses a localized distributed algorithm in which trust is computed using direct and indirect observations [14].

In PLUS, the control messages generated by the BS contains a hashed sequence number (HSN). When a judge node receives a packet from another node, it uses the HSN to check the integrity of the received packet. If the integrity has been compromised, the trust in node i is decreased. However, if node i had just forwarded the packet and it is not a malicious node, it is penalized without being guilty.

D. T-RGR

Trust management scheme for Resilient Geographic Routing (T-RGR) is a non-adaptive scheme in which sensor nodes observe the behavior of their one-hop neighbors [15]. T-RGR is vulnerable to collaborative attacks because it uses direct observations in order to compute trust values.

E. Group-based Trust Management Scheme (GTMS)

Group-based Trust Management Scheme (GTMS) is a method for clustered Wireless Sensor Networks that evaluates the trust of a group of sensor nodes [16]. This approach reduces the memory used to store trust values for each observed entity.

F. Trust-aware Query Processing

Trust-aware Query Processing is a new approach to efficient trust-aware routing in data intensive WSNs [17]. The trust metric is based on subjective logic that includes properties of deployment area, sensor design, and properties of the transmission channels. The approach optimizes energy consumption and provides reliability to data intensive sensor networks.

V. ADAPTIVE TRUST MANAGEMENT PROTOCOL

Wireless Sensor Networks are used to collect data about the environment in which they are placed. This data may refer to temperature, humidity, pressure, light, sound, and advanced properties such as air or water quality, or other specific object attributes.

We assume that nodes in the same range will gather similar measurement data regarding a given environmental property. The optimal range will be determined experimentally because it depends on the deployed network and application.

We also assume that nodes have the capability to determine the distance between them by using ranging techniques such as TOA-based or RSS-based ranging. This topic goes beyond the purpose of this paper. We assume that data packets contain the localization of the source packet.

We define the reputation of a node as a measure of confidence in ability of that node to correctly collect and transmit sensor readings.

Every node computes the penalty values for neighbor nodes on the basis of the packets it has to forward to the base station. After that, the nodes exchange penalty values, and the final reputation for a specific node is computed using the local penalty values and the received ones.

Each sensor node uses reputation values to determine whether it can trust a certain node or not. The trust is represented as a binary value. The trust values are used in order to select which messages will not be forwarded or aggregated.

The Adaptive Trust Management Protocol for Wireless Sensor Networks consists in the following three phases:

A. The setup phase

1) The base station broadcasts a Hello packet.

2) The nodes that receive a Hello packet re-broadcast it in order to reach the whole network.

3) Every node stores the address of the node from which it has received the first Hello packet. This node will be called the parent. This way, a spanning tree overlay will be contructed.

B. The learning phase

1) The learned trust for each neighbor is set to the default value.

2) The nodes start collecting data and sending it to the base station. Every node will forward data towards the base station by using the parent node.

3) The nodes perform error detection using the following algorithm:

a) Leaf nodes just transmit the raw collected data

b) Every other node within the spanning tree waits to receive data from children nodes for a specific period of time. The packets are stored in a list.

c) After the waiting period, based on the location of each source of data, the nodes are grouped in clusters, so that the distance between nodes within a cluster is less that a constant ε.

d) Each cluster of nodes is represented by a list of measurement values generated by member nodes. Each list of nodes is sorted in an ascending manner.

e) For each list of values, the median value is computed. For error detection purposes, the median is a better measure of the central tendency than the average, because it is less sensitive to outliers.

f) For each list, the values are compared with the median value. If the difference between the considered value and the median is greater than a constant deviation γ, the value will be considered erroneous. The constant deviation γ is defined as a percent of the median value. The actual value depends on the application.

4) For each node that is the source of an erroneous data value, the local penalty value will be increased with a specific value. This values depends directly on the diference between the analized and the median values.

5) Each non-leaf node will have a list of associations between nodes and penalty values, called penalty associations.

C. The exchange phase

1) Each node sends the list of penalty associations using a broadcast message.

2) Each node waits to receive the lists of associations from their neighbors for a predefined period of time.

3) After the period of time has expired, the reputation value is recomputed using the current local penalty obtained through the learning phase, the previous reputation value and the penalty values received from the neighbours using the following formula:
$Reputation(X)=Previous\_reputation(X)-Local\_penalty(X)-\Sigma_Y(W_Y*Received\_penalty_Y(X))$. The received penalty from node Y is weighted with W_Y, which represents the trust value that the current node has in node Y. The trust value is either 1 for trusted nodes or 0 for un-trusted nodes.

4) The trust value is recomputed using the following conditions:

Trust(X)=1 if Reputation(X)>=TRUST_LIMIT
Trust(X)=0 if Reputation(X)<TRUST_LIMIT

This computed trust value can be used by parent nodes in order to forward or aggregate data packets received from children that are trustworthy, and ignore packets from children in which they do not trust.

A complete trust management cycle consists in a learning phase and an exchange phase. At the end of a cycle, each node has updated their trust in other nodes, even if they are not reachable within one single hop.

The setup phase is repeated after a specific number of trust management cycles. This phase must be repeated because the topology may change and nodes could lose their parents, and therefore they would not be able to send data to the base station. The number of cycles after which a setup phase must take place depends on the duration of a cycle and on the dynamic of the network. The dynamic of the network depends on the frequency of the topology changes that may be caused by energy depletion and node mobility.

After a topology change, two nodes that were not neighbors in the previous cycle can become parent and child after a setup phase. The parent is now able to use the information it has previously obtained about the new child node.

VI. PROTOCOL IMPLEMENTATION

The protocol has been implemented in TinyOS, an open-source, component-oriented operating system designed especially for Wireless Sensor Networks [18].

A single component was used to implement the protocol and a wiring component is used to place the protocol component on top of the TinyOS Active Message Stack.

The messages used in our implementation of the protocol rely on the following Layer 2 header which corresponds to the TinyOS Active Message header. The real AM header contains other additional fields that are not relevant for the understanding of the Adaptive Trust Management Protocol. The Layer 2 header is represented in Figure 1. The source and destination addressed are AM addresses used for the hop-by-hop communication between nodes.

Hop_src	Hop_dest	Upper Layer data

Figure 1. Layer 2 header

In Figure 2, we present the Layer 3 header, which is specific to our protocol. The source and destination addresses are AM addresses that are used for the end-to-end communication between nodes. The Type field represents the type of message being sent: hello, data or penalty exchange message. The fields X and Y represent the coordinates of the source node used to compute the distance between the nodes in order to form clusters.

End_src	End_dest	Type	X	Y	Upper Layer Data

Figure 2. Layer 3 header

The component contains nine events implemented, from which the most important are the receive event that is used to manage received messages and the fired events for each of the four timers that are used to perform specific actions.

The component uses four timers in order to assure the proper functionality of the protocol: Hello timer, Collect timer, TrustAnalyse timer and TrustExchange timer.

The Hello timer is used only by the base station in order to periodically broadcast Hello messages that are used to build the spanning tree overlay, which corresponds to the Setup phase, step 1.

The *Hello_timer.fired* event is used to periodically send messages containing: the Hop_src and End_src equal to the base station identifier, the Hop_dest and End_dest equal to AM_BROADCAST_ADDR, the broadcast address, the type equal to 1 which represents Hello messages. Fields X and Y are not filled. The Application data contains a sequence number in order to keep track of the Setup phases.

The Collect timer is used by the nodes in the network to periodically collect data from the environment and send it towards the base station, which is the implementation of Learning phase, step 2.

The Collect_timer.fired event sends messages with the following fields: Hop_src and End_src equal to the node identifier that is generating the message, End_dest equal to the identifier of the base station, Hop_dest equal to the parent node identified in the Setup phase, Type equal to 2 which represents Data messages, and fields X and Y containing the coordinates of the source node. The data packet is sent to the parent of the source node.

The TrustAnalyse timer is used in Learning phase, step 3, to model the waiting period in which non-leaf nodes receive data packets from their children and forward them towards the base station.

The TrustAnalyse_timer.fired event implements the algorithm presented in Learning phase, step 3, in which clusters are formed, messages are sorted in lists for each cluster and data errors are detected using the median method. The local penalty values are modified according to the data errors detected and broadcasted to the neighbor messages. The packets used to broadcast the penalty associations contain the following fields: Hop_src and End_src is equal to the node identifier, Hop_dest and End_dest equal to AM_BROADCAST_ADDR, type is set to 3, representing a penalty exchange packet. Fields X and Y will not be filled. The payload contains only the penalty associations modified in the Learning phase.

The TrustExchange timer is used in the Exchange phase, step3, to represent the waiting period in which nodes receive penalty associations from their neighbors.

The TrustExchange_timer.fired event is used to re-compute reputation lists according to the local penalty values and the penalty associations received from the neighbor nodes. The trust binary values are determined by comparing the reputation values obtained with the threshold limit of the accepted reputation.

The Receive.receive event is used to react to every message received by the current node:

1. If the node receiving the message is the base station and the message has type equal to 2, the message contains collected data that reached destination.

2. If the message type is 1 and the node receiving the message has no parent, the Hop_src node becomes its parent. The message is re-broadcasted in order to reach other nodes from the network.

3. If the message type is 2 but the current node does not have a parent yet, the following message is generated: "No route to base station, packet from X with value Y is dropped".

4. If the message type is 2 and the current node has a parent, the Hop-by-hop addresses are changed to reflect the current Hop source (Hop_src) and destination (Hop_dest) and the message is forwarded towards the base station, through the Hop_dest. The message is stored until analyzed in the TrustAnalyse_timer.fired event.

5. If the message type is 3, the received penalty associations are stored locally.

VII. TEST SCENARIO

The Adaptive Trust Management Protocol has been tested using TOSSIM, a simulator for TinyOS applications [19], which is particularly adequate for testing WSN protocols [20], [21].

We use a test scenario based on a simple topology in order to prove the functionality of the protocol. The topology is represented in Figure 3, and it contains ten nodes placed at the coordinates specified in the figure. The line between two nodes indicates that they are in the broadcast range of each other, and therefore they can directly communicate with each other. The three circles observed in the figure represent the clusters identified by the nodes using a specialized algorithm.

We present the output of TOSSIM for every step in the protocol. We choose to display only receive events in order to eliminate redundant data from the figure. Even for broadcast messages, the Hop_dest field in the received packet is the unicast AM address of the receiving node. This behavior is specific to TinyOS implementation. The broadcast address is equal to 65535, as the AM address is represented on 16 bits.

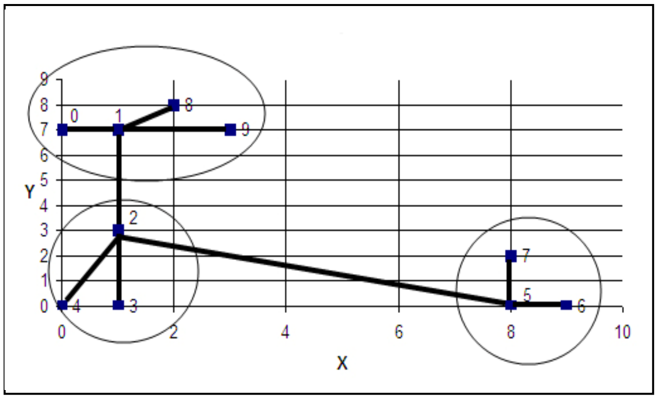


Figure 3. Simple topology

The Setup phase consists in flooding the network with Hello messages and building the spanning tree overlay. As it can be seen in Figure 4, each node learns its parent node when it first receives a Hello packet. For example, node 1 receives Hello messages from node 0, 8, 9 and 2, but it stores as parent, the node from which it has received the first Hello message.

After all nodes have learned their parent, the spanning tree is ready and the sensor nodes can start collecting data and send it towards the base station.

```
(1): Packet received (Hop_src=0 Hop_dest=1 End_src=0 End_dest=65535 type=1)
(1): Parent 0
(9): Packet received (Hop_src=1 Hop_dest=9 End_src=0 End_dest=65535 type=1)
(9): Parent 1
[…]
(3): Packet received (Hop_src=2 Hop_dest=3 End_src=0 End_dest=65535 type=1)
(3): Parent 2
(1): Packet received (Hop_src=2 Hop_dest=1 End_src=0 End_dest=65535 type=1)
(7): Packet received (Hop_src=5 Hop_dest=7 End_src=0 End_dest=65535 type=1)
(7): Parent 5
(6): Packet received (Hop_src=5 Hop_dest=6 End_src=0 End_dest=65535 type=1)
(6): Parent 5
```

Figure 4. Setup phase

The nodes receiving a data packet will forward it using the parent node. A four-hop data routing process is presented in Figure 5.

```
 (5): Packet received (Hop_src=6 Hop_dest=5 End_src=6 End_dest=0 type=2
msg=14)
 (2): Packet received (Hop_src=5 Hop_dest=2 End_src=6 End_dest=0 type=2
msg=14)
 (1): Packet received (Hop_src=2 Hop_dest=1 End_src=6 End_dest=0 type=2
msg=14)
 (0): Packet received (Hop_src=1 Hop_dest=0 End_src=6 End_dest=0 type=2
msg=14)
(0): Received from End_src=6 collected data=14
```

Figure 5. Collect and route data packets

The data packet is generated by node 6, which has the parent node 5. The Layer 2 AM addresses are changes at every hop as it can be observed in Figure 5. Each node on the path forwards the packet to its parent. Node 0 displays a message with the data received and the source node.

```
(0): Received from End_src=2 collected data=8
(0): Received from End_src=3 collected data=9
(0): Received from End_src=4 collected data=12
(0): Received from End_src=7 collected data=15
(0): Received from End_src=1 collected data=4
(0): Received from End_src=2 collected data=9
(0): Received from End_src=3 collected data=10
(0): Received from End_src=5 collected data=14
(0): Received from End_src=6 collected data=15
(0): Received from End_src=9 collected data=6
```

Figure 6. Collected data

Figure 6 presents all data received by the base station in a round. Node 2 has received and stored the messages from nodes 3, 4, 5, 6, 7. Two clusters are identified, cluster1 containing nodes 2, 3, 4 and cluster 2 containing the nodes 5, 6, 7, as they are represented in Figure 3. Two sorted lists are built containing the measurement data from nodes within the two clusters: [8, 9, 12] and [14, 15, 15]. The median values for the two lists are 9 and 15. The value γ was set to 20%, therefore the value 12 collected and sent by node 4 is found to be erroneous.

Initially, the local reputation is set to a default value, for example 100, and local_penalty[4] is set to 0. After the error is detected, local_penalty[4]=3, the difference between the value sent by node 4 and the median value. Node 2 sends a broadcast message announcing that it has detected an error. The message contains the accused node identifier and the error found, as it can be seen in Figure 7. The broadcast message is received by nodes 5, 4, 1 and 3, and they compute the final reputation value based on the local penalty and the received penalty association. The value obtained by all receiving nodes is 97.

```
(2): Trust Packet sent (src=2 dest=65535 End_src=2 End_dest=65535 type=3
node=4 penalty=3)
(2): node=4 reputation=97
(5): Received from 2 dif_reputation=3 in node=4
(4): Received from 2 dif_reputation=3 in node=4
(1): Received from 2 dif_reputation=3 in node=4
(3): Received from 2 dif_reputation=3 in node=4
(3): node=4 reputation=97
(4): node=4 reputation=97
(5): node=4 reputation=97
```

Figure 7. Exchanging penalty associations

Unless the data packet from node 4 is dropped by node 2, the process is repeated by node 1, which also detects that the value sent by node 4 is erroneous and announces nodes 0, 8 and 9.

VIII. DISCUSSION

The TRUST_LIMIT value used to compute trust depends on the application and the behavior of the sensor nodes. If the reputation of a node continues to drop under a certain limit, the node should not be trusted anymore and the packets received from it should not be forwarded or used in the aggregation process. Therefore, the procedure for computing the reputation and trust has the advantage of eliminating both nodes that perform one serious error and nodes that generate many relatively small errors.

A problem arises if one of the non-leaf nodes that forwards data towards the base station starts modifying the data contained in the packets. This behavior could be determined by a failure in the node or because it is malicious. We found the solution to integrate a Message Authentication Code (MAC) into the message that would be computed with a secret key shared only between the source node and the base station. This way, if the data packet is modified on the way, the malicious node does not have the secret key of the source node, therefore it will not re-compute correctly the MAC.

Another problem could appear regarding the formula for computing the reputation, in which the reputation of a node can only descrease or stay constant, but can not increase or return to baseline. In some cases a redemption procedure is needed. The formula can be adjusted as follows: Reputation(X)=Previous_reputation(X)+Local_penalty(X)+ $\Sigma_Y(W_Y*\text{Received\_penalty}_Y(X))$, where the penalty is negative and proportional to the detected error in the case of fault detection, and the penalty is positive and equal to a value determined experimentally if measured values are detected as normal.

The major advantage of this protocol is that it can detect data packets generated by faulty or malicious nodes and drop them before reaching the base station. Therefore, a number of useless send and receive operations are avoided and energy consumption is minimized.

The filtering of erroneous data is also very useful for the data aggregation process. Once aggregated, the base station would not be able to detect errors in the received data. Therefore, data values must be verified before being aggregated.

IX. CONCLUSIONS

Wireless Sensor Networks are deployed in order to provide a service to the end user. Medical and military monitoring consists in critical services provided that must be protected using an efficient security solution.

We developed the Adaptive Trust Management Protocol for Wireless Sensor Network, a protocol that computes reputation and trust based on fault detection in three phases organized in cycles.

One cycle contains a Setup phase and a number of Learning and Exchange phases. In the Setup phase, the base station broadcasts Hello messages that reach every node in the network. A spanning tree overlay is build by learning the parent of each node from the first Hello message received in a cycle.

In the Learning phase, the nodes group the messages received in a predefined period of time by location and determine the erroneous data based on the assumption that two nodes that are close to each other should have similar sensor readings. Based on the errors detected, the local penalty values are modified.

In the Exchange phase the local penalty values are exchanged with their neighbors and the reputation and trust values are recomputed using the local penalty values and the received penalty associations.

The trust values can be used to filter erroneous data packets before reaching the base station, in order to minimize the energy consumption, and to obtain correct data during the aggregation process.

The protocol has been implemented in TinyOS and its functionality has been evaluated in a test scenario using TOSSIM.

BIBLIOGRAPHY

[1] L. Gomez and C. Ulmer, "Secure Sensor Networks for Critical Infrastructure Protection", 2010 Fourth International Conference on Sensor Technologies and Applications, 2010, pp. 144-150.

[2] K. Kabri and D. Seret, "An evaluation of the cost and energy consumption of security protocols in WSNs", 2009 Third International Conference on Sensor Technologies and Applications, 2009, pp. 49-54.

[3] F. C. Delicato, P. F. Pires, L. Pirmez, and T.V. Batista, "Wireless Sensor Networks as a Service", 2010 17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2010, pp. 410-417.

[4] E. Avilés-López and J.A.García-Macías, "TinySOA: a service-oriented architecture for wireless sensor networks", Service Oriented Computing and Applications, Vol. 3, No. 2, 2009, pp. 99-108.

[5] D.I. Curiac, M. Plastoi, O. Banias, C. Volosencu, R. Tudoroiu and D. Pescaru, "Software Development for Malicious Nodes Discovery in Wireless Sensor Network Security", 2010 Fourth International Conference on Sensor Technologies and Applications, 2010, pp. 402-407.

[6] L. Gheorghe, R. Rughiniş, R. Deaconescu, and N. Ţăpuş, "Authentication and Anti-replay Security Protocol for Wireless Sensor Networks", The Fifth International Conference on Systems and Networks Communications, August 22-27, 2010, pp. 7-13.

[7] J. Zheng and A. Jamalipour, "Wireless Sensor Networks A Networking Perspective", John Wiley & Sons, 2009.

[8] F. Ye, H. Luo, S. Lu, and L Zhang, "Statistical en-route filtering of injected false data in sensor networks", 23th Annual IEEE Joint Conference of the IEEE Computer and Communication Societies (INFOCOM'04), vol. 23, no. 4, March 2004, pp. 839-850.

[9] G.R. Abuaitah, "Trusted Querying over Wireless Sensor Networks and Network Security Vizualization", Master of Science Thesis, 2006.

[10] M. Blaze, J. Feigenbaum, and J. Lacy. "Decentralized Trust Management". In IEEE Symposium on Security and Privacy, 1996.

[11] Dong Hui-hui, Guo Ya-jun, Yu Zhong-qiang, Chen Hao, "A Wireless Sensor Networks Based on Multi-angle Trust of Node," ifita, vol. 1, pp.28-31, 2009 International Forum on Information Technology and Applications, 2009.

[12] S. Ganeriwal and M.B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '04), pp. 66-67, Oct. 2004.

[13] A. Boukerche, X. Li, and K. EL-Khatib, "Trust-Based Security for Wireless Ad Hoc and Sensor Networks," Computer Comm., vol. 30, pp. 2413-2427, Sept. 2007.

[14] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and Localized Trust Management Scheme for Sensor Networks Security," Proc. Third IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '06), pp. 437-446, Oct. 2006.

[15] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," J. Parallel and Distributed Computing, vol. 67, no. 2, pp. 215-228, 2007.

[16] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y. Song, "Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, pp. 1698 - 1712, 2009.

[17] V. Oleshchuk and V.Zadorozhny, "Trust-Aware Query Processing in Data Intensive Sensor Networks", 2007 International Conference on Sensor Technologies and Applications, 2007, pp. 176-180.

[18] P Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler, "TinyOS: An Operating System for Sensor Networks", Ambient Intelligence, 2005, pp. 115-148.

[19] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", In SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, 2003, pp. 126-137.

[20] L. Gheorghe, R. Rughiniş, and N. Ţăpuş, "Fault-Tolerant Flooding Time Synchronization Protocol for Wireless Sensor Networks", The Sixth International Conference on Networking and Services, ICNS 2010, March 7-13 – Cancun, Mexico, 2010, pp. 143-149.

[21] R. Rughiniş and L. Gheorghe, "Storm Control Mechanism for Wireless Sensor Networks", 9th RoEduNet IEEE International Conference, June 24-26, 2010, pp. 430-435.