



**Biblioteka Informatyki
Szkół Wyższych**

Information Systems Architecture and Technology

Service Oriented Networked Systems



Library of Informatics of University Level Schools

Series of editions under the auspices
of the Ministry of Science and Higher Education

The ISAT series is devoted to the publication of original research books in the areas of contemporary computer and management sciences. Its aim is to show research progress and efficiently disseminate current results in these fields in a commonly edited printed form. The topical scope of ISAT spans the wide spectrum of informatics and management systems problems from fundamental theoretical topics to the fresh and new coming issues and applications introducing future research and development challenges.

The Library is a sequel to the series of books including Multidisciplinary Digital Systems, Techniques and Methods of Distributed Data Processing, as well as Problems of Designing, Implementation and Exploitation of Data Bases from 1986 to 1990.

Wrocław University of Technology



Information Systems Architecture and Technology

Service Oriented Networked Systems

Editors

Adam Grzech

Leszek Borzemski

Jerzy Świątek

Zofia Wilimowska

Wrocław 2011

Publication partly supported by
Faculty of Computer Science and Management
Wrocław University of Technology

Project editor
Arkadiusz GÓRSKI

The book has been printed in the camera ready form

All rights reserved. No part of this book may be reproduced,
stored in a retrieval system, or transmitted in any form or by any means,
without the prior permission in writing of the Publisher.

© Copyright by Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2011

OFICyna WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
<http://www.oficwyd.pwr.wroc.pl>;
e-mail: oficwyd@pwr.wroc.pl
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-631-6

CONTENTS

Introduction	5
--------------------	---

PART I. SERVICE ORIENTED SYSTEMS – FUNDAMENTALS AND EXAMPLES

1. Andrzej RATKOWSKI, Krzysztof SACHA Business Process Design In Service Oriented Architecture	15
2. Ilona BLUEMKE, Wojciech KIERMASZ Experience In SOA Based Integration Of Systems	25
3. Szymon KIJAS, Andrzej ZALEWSKI, Krzysztof SACHA, Marcin SZLENK, Andrzej RATKOWSKI Formal Semantics Of Architectural Decision Making Models As A Component Of An Integrated Evolution Methodology For Service-Oriented Systems	37
4. Paweł STELMACH, Łukasz FALAS Service Composer – A Framework For Elastic Service Composition	47
5. Jan WEREWKA, Grzegorz ROGUS A Solution For Adaptation Of Legacy Enterprise Software For Private Cloud Computing Model	61

PART II. SOA-BASED APPLICATIONS – SELECTED ISSUES

6. Sergiusz STRYKOWSKI, Rafał WOJCIECHOWSKI Ontology-Based Modeling For Automation Of Administrative Procedures	79
7. Adam CZARNECKI, Cezary ORŁOWSKI Application of Ontology in the ITIL Domain	99
8. Adam WÓJTOWICZ, Jakub FLOTYŃSKI, Dariusz RUMIŃSKI, Krzysztof WALCZAK Securing Learning Services Accessible With Adaptable User Interfaces	109
9. Maciej ZIĘBA Ensemble Methods For Customer Classification In Service Oriented Systems	119
10. Bogumila HNATKOWSKA, Bartosz NOWAKOWSKI Windows azure cloud Computing versus classical solutions – COST Comparison....	129

PART III. MODELING AND MEASURING QUALITY OF KNOWLEDGE AND SERVICES

11. Leonardo MANCILLA AMAYA, Cesar SANIN, Edward SZCZERBICKI An Agent-Based Approach To Measure Knowledge Quality	143
--	-----

12. Piotr CZAPIEWSKI Strategy For A Robust Aggregation Agent In A Multi-Agent Automated Trading Environment.....	155
13. Katarzyna MICHALSKA, Tomasz WALKOWIAK Analytical And Experimental Dependability Metrics For Service-Based Information System.....	165
14. Adam CZYSZCZOŃ, Aleksander ZGRZYWA An Artificial Neural Network Approach To Restful Web Services Identification.....	175
15. Anna DOBROWOLSKA, Wiesław DOBROWOLSKI The Use Of Electronic Questionnaires In Service Quality Assessment	185

PART IV. SECURITY PROTOCOLS, PROCEDURES AND ALGORITHMS

16. Jakub FLOTYŃSKI, Willy PICARD Transparent Authorization and Access Control in Event-Based OSGi Environments	197
17. Tymon MARCHWICKI, Grzegorz KOŁACZEK Security Level Evaluation And Anomaly Detection In Data Exchange For Service Oriented System	211
18. Grzegorz Górski Novel Multistage authorization Protocol	221
19. Ewa IDZIKOWSKA Errors Detection In S-Boxes Of Hash Function HaF-256.....	231

PART V. NETWORKS

20. Józef WOŹNIAK, Przemysław MACHAŃ, Krzysztof GIERŁOWSKI, Tomasz GIERSZEWSKI, Michał HOEFT, Michał LEWCZUK Performance Analysis Of Mobility Protocols And Handover Algorithms For IP- Based Networks	243
21. Sylwester KACZMAREK, Magdalena MŁYNARCZUK, Marcin NARLOCH, Maciej SAC Evaluation Of ASON/GMPLS Connection Control Servers Performance	267
22. Mariusz GŁĄBOWSKI, Michał STASIAK Internal Blocking Probability Calculation in Switching Networks with Additional Inter-Stage Links.....	279
23. César DE LA TORRE Design Of Secure And Cost Efficient Networks To Support Cloud Computing Applications	289
24. Mariusz GŁĄBOWSKI, Maciej SOBIERAJ, Maciej STASIAK, Piotr ZWIERZYKOWSKI Modeling of Resource Management Mechanisms for Virtual Networks.....	303
25. Jan KWIATKOWSKI, Grzegorz PAPKAŁA Service Aware Virtualization Management System	317

INTRODUCTION

Information and communication technologies are facing ever increasing pressure to extend services and make them available everywhere. ICT now has to support processes and deliver information systems services which no longer ends at the organizational boundaries. The business requires new deployments and operations where variety of components and systems are assembled into enterprise applications and business processes with a higher degree of flexibility.

Observed trends in the contemporary ICT technologies and their applications may be distinguished as motivated by several important and perspective paradigms leading to communication enabled applications offered by service oriented systems with quality of services assured by intensive knowledge processing.

The service oriented systems paradigm means that the changing business scenarios are supported by components deployed on multiple hardware servers available in highly distributed heterogeneous environment. Distributed application components, combined in distributed environment, possess particular quality of service challenge. Services delivered in such an environment have to be supported by mechanisms allowing both individualization as well as customization of services requiring to process in real-time manner high volume of knowledge, information and data.

Communication enabled applications paradigm stresses the fact, that information and communication technologies components are integrated using a particular service-oriented architecture (SOA) to increase the productivity of an organization and/or improve the quality of users' experiences. Such approach is characterized by communication enablement (adds real-time networking functionality an application) and communications capability to an application (removes the human latency).

The book addresses some set of subjects dealing with various technological and applications aspects of distributed information and communication systems, i.e., technologies, organization, application and management involved in gain to increase business efficiency, computational and communication resources utilization, services flexibility, applications functionalities and quality of services offered by contemporary information and computer systems.

Chapters, selected and presented in the book, are related to number of issues important and representative both for available information and communication

technologies as well as information system users requirements and applications. Submissions, delivered within below presented chapters, are strongly connected with issues being important for contemporary distributed information systems, computer communication systems as well as intelligent knowledge, information and data processing supporting decision-making systems: service-oriented architectures, communication enabled applications, services composition, quality of service based and supported by intensive knowledge processing, security of data and services, etc.

The book is divided into five parts, which include twenty five chapters. Particular parts contain chapters addressing – sometimes very particular – issues both representative for contemporary research and technical topics and important for today’s information and communication technologies and their applications.

The five parts have been completed from chapters addressing some interesting, important and actual issues of distributed information and communication systems. The objective of the proposed decomposition of accepted set of chapters into parts is to compose units presenting similar problems or attempts, methods, algorithm and tools for information systems requirement analysis, processes and systems modeling, analysis of users communities and systems services as well as modeling, analysis and optimization of systems infrastructures enabling delivery of knowledge-based services.

The first part – **SERVICE ORIENTED SYSTEMS – FUNDAMENTALS AND EXAMPLES** – contains six chapters addressing some selected issues related to well-known, intuitive and intensively deployed service orientation concept, according which the business processes are supported by customized services composed of heterogeneous components delivered in distributed and heterogeneous environment. In chapters, included in this part, different aspects of customers and services profiles description, processing, customization and deployment are considered. In the chapters advantages as well as shortcomings and difficulties in implementations of service oriented architecture and service oriented knowledge utilities paradigms and concepts are presented.

The second part – **SOA-BASED APPLICATIONS – SELECTED ISSUES** – is composed of chapters where implemented and possible applications, based on service-oriented approach, are presented and discussed. Chapters included in this part show how business processes may be modeled, granulated and supported by information systems and their services to fulfill functional and nonfunctional requirements changes. In the chapters it is also possible to find how complex is the modeling of business processes and how knowledge-intensive is the mapping of business processes into information systems’ services and vice versa.

The next, third part – **MODELING AND MEASURING QUALITY OF KNOWLEDGE AND SERVICES** – is devoted to present and discuss some selected problems strongly connected with various quality of service delivery strategies for networked systems. Problems and issues discussed in this part are related to process of knowledge collecting, decision making, quality of services measures and methods of the measures calculation. Issues, elaborated in the presented chapters, show that new

attempts, models, methods and algorithms are required in gain to obtain better knowledge collecting procedure and most adequate measures of services quality required to knowledge exploitation.

The fourth, last part – **SECURITY PROTOCOLS, PROCEDURES AND ALGORITHMS** – is dedicated to present and discuss some issues related to methodologies, concepts, methods and algorithms necessary to assure required level of security in communication and processing systems. The chapters, included in this part, address various issues related to methods and tools as well as application of security assuring mechanisms in contemporary computer-based information systems. Discussed attempts show that the quality of service in distributed systems are complex issues and that the assurance of required security is influenced mainly by assumed data and information processing paradigms.

The last, fifth part – **SELECTED PROBLEMS OF NETWORKS** – contains chapters related to both circuit-switched and message-switched communication networks. The five chapters cover some selected aspects of networking being strongly connected with network services availabilities, communication enabled communication, network services for cloud computing and resources virtualization.

PART I. SERVICE ORIENTED SYSTEMS – FUNDAMENTALS AND EXAMPLES

The **Chapter 1** describes a method for transformational design and implementation of business processes in IT system based on Service Oriented Architecture (SOA) paradigm. The chapter presents the process from the initial specification expressed in Business Process Execution Language (BPEL) to the design phase consisting of a series of transformations, which change the internal process structure without changing the observable process behavior. The gain of presented work is to improve the quality of the business by increasing the efficiency of the process execution exploiting parallelism of services.

The basic aim of the **Chapter 2** is devoted to present experiences collected in the two medical systems integration based on Service Oriented Architecture (SOA) paradigm. The chapter contains selected information about the integrated systems as well as the web services exposed by the systems and used to integrate them. The presented integration is based on *Publish* integration pattern. Some technical details related to business process modeling, system architecture, services and implementation are given. Carried out integration of systems allows to describe weak and strong points of the SOA based integration.

In the **Chapter 3** the problem of systems evolution by providing means for composing new functionalities out of services provided by already existing systems, or by changing the implemented functionality by rearranging and/or enhancing services composition is discussed. In particular, the chapter addresses issues related to evolution methodology for service-oriented systems that would integrate software

development with IT operations management. Authors present original methodology aimed at bridging this above mentioned gap by integration of change management process, engineering methods, models and supporting tools. Moreover, the contribution aim is to show how formal semantics should help to resolve the discussed issues and make the concepts comprehensible and well-founded.

Next chapter, **Chapter 4**, is devoted to present an approach to service composition based on the idea of Smart Services. The proposed framework aids in definition of composition services with domain specific selection methods. A crucial part of this framework is a workflow engine capable of executing composed services as well as composition scenarios themselves. The engine is closely integrated with various composition scenarios enabling the user to define how a specific type of a service (or even a requirement) is meant to be executed.

The aim of the **Chapter 5** is to examine an issue, in which two or more companies have complementary software used by customers in a given industry domain. Reported results of the business analysis show a necessity of adapting current software to the cloud computing implementation model to provide potential customers with a wide range of services. Authors propose a solution for fast adaptation of legacy enterprise software in the private cloud computing model.

PART II. SOA-BASED APPLICATIONS – SELECTED ISSUES

The **Chapter 6** gain is discuss shortcomings of classic modeling methods based on the monolithic approach which do not allow creating models of administrative procedures at the level of detail allowing the automation. The main contribution of the chapter is a new approach to modeling of administrative procedures for their automation purposes. In the proposed approach, models of administrative procedures are dynamically composed of elementary processes. The selection of the elementary processes is performed based on analysis of legal circumstances occurring during the runtime phase. The analysis of the legal circumstances is performed by an inference engine evaluating decision rules against facts.

The next **Chapter 7** is devoted to present and discuss some issues related with the application of ontological description powered by the capabilities of the OWL language to express Information Technology Infrastructure Library (ITIL). One of the goals of this initial study is to determine the usefulness of such semantic model in organizations that adopted or plan to adopt ITIL.

In the **Chapter 8** some issues related to mobile devices limitations and attempts allowing to overcome such devices limitations in e-learning systems. To overcome such mobile devices' well-known limitations and to enable effective use of learning content on mobile devices Authors propose dynamically generated, adaptable user interfaces adjusted to a particular device, a particular user, and particular context. In this chapter, an extension of the MILES 2.0 learning system based on the Web 2.0

platform, with authentication, authorization, access control, and encryption is presented.

In the **Chapter 9** a SOA-based system's customers classification as important for services customization, services orchestration and quality of services evaluation is discussed. The ensemble classifier with switching class labels is proposed as classification method which solves two of identified problems related to the processing of data characterizing SOA-based systems' customers: imbalanced data and cost-sensitive learning problems.

The **Chapter 10** is to discuss advantages and shortcomings of the Windows Azure, which is one of the newest clouds available on the market. In the chapter a cost of using a system deployed in Windows Azure and servicing by a company is evaluated and presented. Some simulations results are presented to illustrate efficiency of cloud computing in assumed environment and circumstances.

PART III. MODELING AND MEASURING QUALITY OF KNOWLEDGE AND SERVICES

The aim of **Chapter 11** addresses increasing importance of knowledge as an organizational asset and the needs to explore new and diverse mechanisms to measure quality of knowledge. This chapter presents the design and implementation of a new agent-based approach to measure knowledge quality. User feedback and automated agent calculations are taken into account to obtain a percentage, which represents an individual's knowledge quality. This approach is used by the e-Decisional Community, an integrated knowledge sharing platform that aims at the creation of markets where knowledge is provided as a service.

The **Chapter 12** concerns the problem of building a multi-agent automated trading environment, where multiple agent types will participate in many aspects of decision support for trading financial instruments. A problem of decision aggregation occurs, when several decision support agents are to be used simultaneously, while only one recommendation should be given. A robust strategy agents decisions aggregation in the presence of uncertainty and in case of unreliability of communication medium or instability of participating agents are proposed and presented. The discussed attempts are illustrated by tests results.

In the **Chapter 13** several dependability metrics of service-based information systems and methods of the metrics calculations are provided. Two general groups of metrics, i.e., analytical (network cohesion in the system, number of services involved in the compound service, absolute importance, dependence and criticality of the service, overall reliability of the compound service) and experimental (task response time, service component availability) are considered. Authors illustrate the discussed metrics by numerical results obtained in own developed simulation environment.

In the next **Chapter 14** some issues related to investigate the state of Web services in gain to propose a solution to identify services and then to create discovery tool in

form of a Web service search engine. Presented approach is based on artificial neural network and is devoted to classify RESTful Web services based on their link structure patterns. Introduced research includes the analysis of service's resources and their parameters in order to create a generic description of particular Web service. The presented approach is illustrated by results of experiments.

In the **Chapter 15** selected topics connected with the use of electronic questionnaires in organizations for assessing the quality of services are discussed. First of all, it shows the advantages and disadvantages of electronic questionnaires, the procedure of electronically-assisted questionnaire surveys, as well as errors and problems associated with preparation of electronic questionnaires and with conducting the surveys, which affect the service quality assessment.

PART IV. SECURITY PROTOCOLS, PROCEDURES AND ALGORITHMS

The **Chapter 16** addresses Open Services Gateway Initiative Framework (OSGi) module security issues. An OSGi module, referred to as bundle, is an application element which has its own lifecycle: it can be remotely installed, started, updated and uninstalled without rebooting the environment. The openness of event-based OSGi environments raises the need for means for authorization and access control to bundles, to protect both applications and bundles from unauthorized accesses. In this chapter, a transparent authorization and access control protocol for event-based OSGi environments is proposed. It is based on an authorization method relying on security policies in which access control may be granted to an event sender using a particular application, for a given activity to be performed by a particular bundle, and with regard to contextual data.

In the **Chapter 17** an anomaly detection method and its application to security level monitoring in service oriented systems is presented. The discussed method main idea combines monitoring of the system's security status and risk factors identification which can be used for improvement of security management of the system. This anomaly detection offers two classes of services: monitoring service and security incidents detection. The proposed security analyzer is based on software agents dedicated to detect system's behavior, separate service and user behavior anomalies.

The gain of the **Chapter 18** is to present novel authentication protocol where typical architecture of user authentication protocol consisting of three independent protocol components i.e. authorization client, pass-through authenticator and authentication server is utilized. Due to applied multistage authorization sequence during each couple of protocol components must identify each other. The new solution can ensure transmitted data protection even in case of enemy pass-through authenticator component interception. The order of identity verifications between protocol components assumes that first authorization client and pass-through authenticator have to mutually prove their identities. This feature of new authentication protocol are shortly discussed.

The aim of the **Chapter 19** is to present dedicated cryptographic hash function suitable for integrity of data verification for both software and hardware implementation. In the chapter a parity based Concurrent Error Detection (CED) approach to protect the S-boxes core of function HaF is presented. The proposed approach capabilities are compared with the results obtained with Duplication With Comparison (DWC) scheme. Simulation results for single and multiple as well as for transient and permanent faults illustrates advantages of the presented approach.

PART V. SELECTED PROBLEMS OF NETWORKS

The **Chapter 20** is devoted to present trends and consequences of rapid growth of IP-based networks and services which created the vast collection of resources and functionality available to users by means of a universal method of access. One consequence of this development are multiple extensions of the IP protocol stack to support users and devices mobility. The general topics, related to the mobility in IP protocol-based environment, are accompanied by short overview of the most popular methods of handling mobility in IPv4 and IPv6 networks, along with their overall performance analysis and comparison. The overview precedes presentation of IP mobility mechanisms critical performance issues as well as optimizations methods already proposed in standardized solutions.

In the **Chapter 21** an architecture and performance of ASON/GMPLS Connection Control Servers (CCSs) are described. In order to evaluate the performance of the ASON/GMPLS connection control elements, a set of scenarios including setting-up and releasing connections were executed in different variants of testbed architecture. During the tests execution call setup and termination operations durations were measured. Test results of tests confirmed that connection control layer performance has the main impact on service request processing duration and the influence of the other testbed elements operation is negligible.

The aim of the **Chapter 22** is to present analytical approach to blocking probability calculation in multi-service switching networks with overflow links. In order to evaluate the accuracy of the proposed analytical method, the results obtained on the basis of the proposed method are compared with simulation results.

In the **Chapter 23** some aspects of secure and cost efficient networks to support cloud computing services over public infrastructure are discussed. The network design and resources management tasks are considered as a multistage process. The problems of optimal service visibility, dependencies among network's elements involved in delivering services, traffic measure and classification and security technologies integration in critical networks elements to enforce security policies are discussed.

The aim of the **Chapter 24** is to evaluate the effectiveness of resource management mechanisms within the context of virtual networks. The chapter presents resource management mechanisms for full-availability systems, e.g., dynamic reservation, static

reservation, threshold mechanisms and priority mechanisms. The effectiveness of the mechanisms under consideration is evaluated on the basis of analytical methods that made it possible to determine the value of the blocking probability and the value of offered traffic in the full-availability systems with BPP traffic and different resource management methods.

The last, not least, **Chapter 25** discusses computing resources allocation problem in SOA-based environment. To select and indicate computing resources, knowledge about the allocated communication resources and the current loading of computing resources are used. The solution allows to select resources, which are dynamically matched to services in gain to satisfy required nonfunctional requirements. The decision on the allocation of computing resources is further compared with the utilization of allocated resources. This allows to gather knowledge about the quality of the methods used for allocating resources and the need to modify them.

Wroclaw, September 2011

Adam Grzech

PART I

**SERVICE ORIENTED SYSTEMS
– FUNDAMENTALS AND EXAMPLES**

Andrzej RATKOWSKI*, Krzysztof SACHA*

BUSINESS PROCESS DESIGN IN SERVICE ORIENTED ARCHITECTURE

This chapter describes a method for transformational design and implementation of business processes in Service Oriented Architecture (SOA). The starting point of the method is an initial process specification expressed in Business Process Execution Language (BPEL). The design phase consists of a series of transformations, which change the internal process structure without changing the observable process behavior. The goal of each transformation is to improve the quality of the initial BPEL process, defined by a set of metrics, and to benefit from the parallel structure of services and improve the efficiency of the process execution. The result is a transformed BPEL process, which can be executed on a target SOA environment using a BPEL engine.

1. INTRODUCTION

A business process is a set of partially ordered activities, which produce a specific product or service that adds value for a customer. The structure of a business process and the ordering of activities reflect business decisions made by business people and, when defined, can be visualized using an appropriate notation, e.g. UML activity diagram [1], Business Process Management Notation [2] or Business Process Executable Language [3]. The implementation of a business process on a computer system is expected to exhibit the defined behavior at a satisfactory level of quality. Reaching such a level of quality may require restructuring of the initial process according to a series of technical decisions, which have to be made by technical people.

This chapter describes a transformational method for designing business process implementation in Service Oriented Architecture (SOA) [4]. The starting point of the method is an initial process specification, called a reference process, defined by busi-

* Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warszawa, Poland.

ness people as a program in Business Process Execution Language (BPEL). The method iterates through a series of steps, each of which makes a small transformation of the process structure. The transformations are selected manually by human designers (technical people) and performed automatically, by a software tool. Each transformation improves the quality of the process implementation, e.g. by benefiting from the parallel structure of services, but preserves the behavior of the reference process. When the design goals have been reached, the iteration stops and the result is a transformed BPEL process, which can be executed on a target SOA environment.

Basic elements of the method are described in the subsequent sections of this chapter. Process behavior and the meaning of behavior preservation are defined in Section 2. Sample transformations are described in Section 3. Quality metrics are introduced in Section 4. Conclusions and plans for further research are given in Section 5.

2. THE BEHAVIOR OF A BUSINESS PROCESS

A reference process defines a correct flow of computation that is acceptable from the application viewpoint. In the transformation phase, the structure and performance characteristics of the reference process are changed. However, this must not change the behavior of the process. The most important problem is how to define this behavior and prove that it has not been changed. The methods relying on a comparison of states generated during the execution, are inappropriate in a SOA environment.

The behavior of a reference process as well as the behavior of a transformed process may not be deterministic, due to the internal concurrence of services invoked during the execution. On the other hand, the developed software must be deterministic, in that it must not work properly or improperly at random. These two statements are not contradictory, but imply that the non-determinism must not touch these aspects of the software behavior, which are essential from the application viewpoint. We assume that the observable behavior of a process in a SOA environment consists of the values of all the variables that are visible to the outside world, i.e. variables that are passed as arguments when external services are being invoked and variables that are returned at the end of the process execution. This is sufficient, because the services are stateless [4] and return the same results if invoked with the same values of the arguments.

Therefore, we assume that a transformation does not change the process behavior if it does not change the values of the observable process variables. Such a definition neglects timing aspects of the execution. This omission is justified because there are many delays in a network of a SOA system and the correctness of a software must not rely on a specific order of activities, unless they are explicitly synchronized.

To capture the behavior of a process, we use a technique called program slicing [5,6], which allows finding all the commands in a program that influence the value of a variable

in a specific point of the program. For example, the value of a variable that is used as an argument by a service invocation command or by the final reply command of the process.

The conceptual tool for the analysis is Program Dependence Graph (PDG) [7,8], which nodes are commands of a BPEL program and edges are control and data dependencies between these commands. An algorithm for constructing PDG of a BPEL program consists of the following steps (Fig. 1):

1. Define nodes of the graph, which are commands at all layers of nesting.
2. Define control edges (solid lines in Fig. 1b), which follow the nested structure of the program, e.g. an edge from `<sequence>` to `<receive>` shows that `<receive>` command is nested within the `<sequence>` element.
3. Define data edges (dashed lines in Fig 1b), which show data dependencies between commands, e.g. an edge from `<receive>` to `<invoke_1>` shows that an output value of `<receive>` is used as an input argument of `<invoke_1>`. The edges at higher levels of nesting, e.g. from `<receive>` to `<sequence>`, are derived from the existence of edges between the leaf command nodes.
4. Add data edges from `<receive>`, which is the first command in the outer `<sequence>` of the program, to each command of this `<sequence>` element. Add data edges to `<reply>`, which is the last command in the outer `<sequence>` of the program, from each command of this `<sequence>` element. These edges reflect the semantics of receive-reply construct and are not shown in Fig 1b.

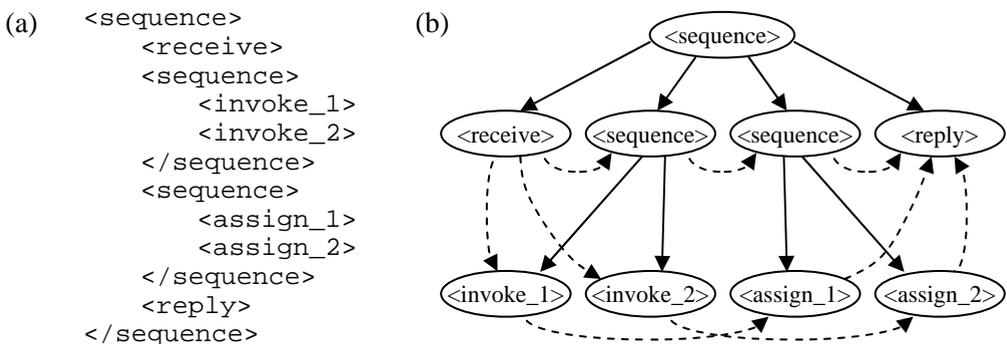


Fig. 1. A nested BPEL process: Process body (a) dependence graph (b)

A path composed of data edges in a program dependence graph expresses the data flow relationships between the commands and implies that the result of the command at the end of a path depend only on the results of the earlier commands of this path. Hence, a transformed program, which preserves the data paths of the reference program dependence graph and guarantees that the order of execution complies with the data paths, preserves the observable behavior of the reference program.

3. TRANSFORMATIONAL DESIGN

BPEL process consists of structured elements, such as $\langle sequence \rangle$, $\langle flow \rangle$, etc. that can be nested in each other. The behavior of a process results from the behavior and the order of execution of particular structured elements. A transformation applies to an element and consists in replacing one element, e.g. $\langle sequence \rangle$, by another element, e.g. $\langle flow \rangle$, composed of the same commands. If the behavior of both elements, i.e. the element before a transformation and the replacing element, and their position in a process are the same, then the behavior of the process stands also unchanged.

It is obvious from the above definition that a superposition of two or more transformations, which do not change the behavior of the transformed elements, preserves the behavior of the process.

Several transformations have been defined. The basic ones are the following: Permutation, parallelization and serialization of the process operations, aggregation of processes into a single entity and a split of a single process. The first three of these transformations are described in detail below.

Transformation 1: Permutation

Consider a BPEL $\langle sequence \rangle$ element, which contains n arbitrary commands C_1 through C_n (Fig. 2a) that are executed in a strictly sequential order. The particular commands can be simple actions, e.g. $\langle assign \rangle$ or $\langle invoke \rangle$, as well as structured elements, e.g. $\langle sequence \rangle$ or $\langle flow \rangle$. Transformation 1 changes the order of commands by exchanging two commands C_i and C_j (Fig. 2b).



Fig. 2. Permutation of commands: Before (a) and after (b) the transformation

Theorem 1: Exchanging commands C_i and C_j does not change the behavior of the $\langle sequence \rangle$ element, if for each command C_k , such that $i < k \leq j$, neither a path from C_i to C_k nor a path from C_k to C_j exists in the process dependence graph:

$$(\forall k: i < k \leq j) \sim [C_i \rightarrow C_k \vee C_k \rightarrow C_j]$$

Proof: The dependence graph of the element before the transformation (Fig. 2a) is shown in Fig. 3. Commands $C_1 \dots C_n$ are executed sequentially from left to right. The order of commands C_i and C_j has no influence on the result of any command C_l , $l < i$, which is executed before either C_i or C_j is started, as well as on any command C_m , $j < m$, which is executed later. However, permutation of C_i and C_j can influence the commands that are between. If a path from C_i to C_k exists in the graph, then permutation of C_i and C_j moves C_i after C_k , which depends on the result of C_i . Similarly, if a path from C_k to C_j exists in the graph, then permutation of C_i and C_j moves C_k after C_j , which depends on the result of C_k . If no paths between C_i , C_k and C_j exist in the graph, then the permutation can not change the result of any of these commands.

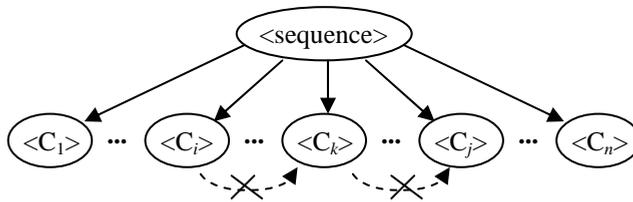


Fig. 3. Dependence graph of a sequence (Fig. 1a)

Transformation 2: Parallelization

Consider a BPEL *<sequence>* element, which contains n arbitrary commands C_1 through C_n (Fig. 4a). The particular commands can be simple actions as well as structured elements. Transformation 2 parallelizes the execution of commands by replacing BPEL *<sequence>* element by *<flow>* element, composed of the same commands (Fig. 4b), which – according to the semantics of *<flow>* – are executed in parallel.



Fig. 4. Parallelization of commands: Before (a) and after (b) the transformation

Theorem 2: Parallelization of commands C_1 through C_n does not change the behavior of the transformed element, if for each pair of commands C_i, C_j , $i, j \leq n$, neither a path from C_i to C_j nor a path from C_j to C_i exists in the process dependence graph:

$$(\forall i, j \leq n) \sim [C_i \rightarrow C_j \vee C_j \rightarrow C_i]$$

Proof: The lack of paths between the commands means that no dependencies between these commands exist and the result of any command does not depend on the order and position of other commands from C_1 through C_n . Hence, all the commands can be executed in any order, also in parallel. \square

Transformation 3: Serialization

Consider a $\langle flow \rangle$ element, which contains n arbitrary commands C_1 through C_n (Fig. 5a) that are executed in parallel. The particular commands can be simple actions as well as structured elements. Transformation 3 serializes the execution of commands by replacing BPEL $\langle flow \rangle$ element by $\langle sequence \rangle$ element, composed of the same commands (Fig. 5), which are now executed in parallel.

(a)	$\langle flow \rangle$ $\langle C_1 \rangle \langle /C_1 \rangle$ $\dots \dots$ $\langle C_n \rangle \langle /C_n \rangle$ $\langle /flow \rangle$	(b)	$\langle sequence \rangle$ $\langle C_1 \rangle \langle /C_1 \rangle$ $\dots \dots$ $\langle C_2 \rangle \langle /C_n \rangle$ $\langle /sequence \rangle$
-----	--	-----	--

Fig. 5. Serialization of commands: Before (a) and after (b) the transformation

Theorem 3: Serialization of commands C_1 through C_n does not change the behavior of the transformed element.

Proof: The proof is obvious. Parallel commands can be executed in any order, also sequentially.

Transformations 1 through 3 can be composed in any order, resulting in a complex transformation of the process structure. Transformations 4 and 5 play an auxiliary role and facilitate such a superposition. The proof of safeness of these two transformations is obvious, because neither of them changes the order of execution of commands.

(a)	$\langle sequence \rangle$ $\langle sequence \rangle$ $\langle C_1 \rangle \langle /C_1 \rangle$ $\dots \dots$ $\langle C_k \rangle \langle /C_k \rangle$ $\langle /sequence \rangle$ $\langle sequence \rangle$ $\langle C_{k+1} \rangle \langle /C_{k+1} \rangle$ $\dots \dots$ $\langle C_n \rangle \langle /C_n \rangle$ $\langle /sequence \rangle$ $\langle /sequence \rangle$	(b)	$\langle flow \rangle$ $\langle flow \rangle$ $\langle C_1 \rangle \langle /C_1 \rangle$ $\dots \dots$ $\langle C_k \rangle \langle /C_k \rangle$ $\langle /flow \rangle$ $\langle flow \rangle$ $\langle C_{k+1} \rangle \langle /C_{k+1} \rangle$ $\dots \dots$ $\langle C_n \rangle \langle /C_n \rangle$ $\langle /flow \rangle$ $\langle /flow \rangle$
-----	---	-----	---

Fig. 6. Partitioning of a set of commands: Sequential (a) and parallel (b)

Transformation 4: Sequential partitioning

Transformation 4 divides BPEL *<sequence>* element (Fig. 4a) into a nested structure of *<sequence>* elements (Fig. 6a).

Theorem 4: Partitioning does not change the behavior of the transformed element.

Transformation 5: Parallel partitioning

Transformation 5 divides BPEL *<flow>* element (Fig. 4b) into a nested structure of *<flow>* elements (Fig. 6b).

Theorem 5: Partitioning does not change the behavior of the transformed element.

To illustrate superposition of transformations, consider a process, which invokes two external services and makes two pieces of computation (Fig. 7a), with data dependencies between commands described by a process dependency graph (Fig. 7b).

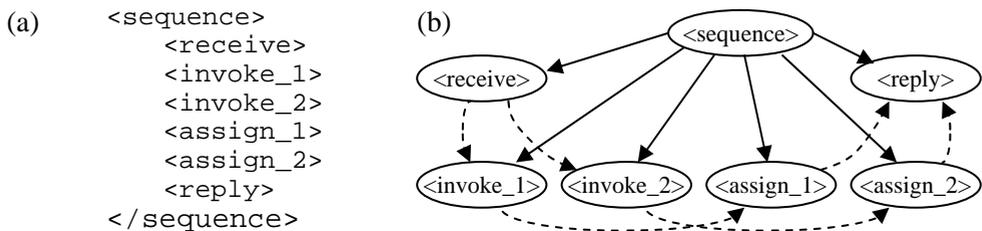


Fig. 7. Reference process: Process body (a) dependence graph (b)

To enable parallel execution of commands within the process, one can use transformation 4 first (Fig. 8a) and then transformation 2 (Fig 8b). The advantage of the transformed process is parallel execution of external services and parallel execution of the computations, which can result in faster response and increased efficiency.

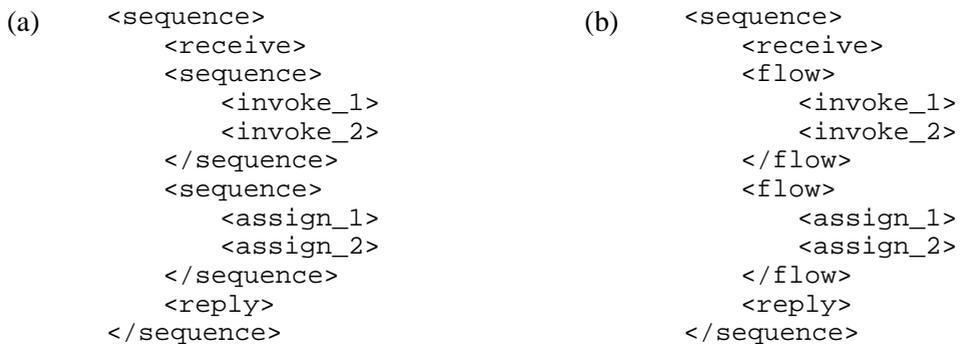


Fig. 8. Transformed process: After transformation 4 (a), and then after transformation 2 (b)

One another possibility of the process transformation is to reorder the commands (transformation 1), then partition the process using transformation 4 (Fig. 9a), and finally parallelize the sub-sequences of commands using transformation 2 (Fig. 9b). The advantages of the transformed process are similar as in the previous case. However, intuition suggests that the structure in Fig. 9b is better than the one in Fig. 8b.

(a)	<pre> <sequence> <receive> <sequence> <sequence> <invoke_1> <assign_1> </sequence> <sequence> <invoke_2> <assign_2> </sequence> </sequence> <reply> </sequence> </pre>	(b)	<pre> <sequence> <receive> <flow> <sequence> <invoke_1> <assign_1> </sequence> <sequence> <invoke_2> <assign_2> </sequence> </flow> <reply> </sequence> </pre>
-----	--	-----	--

Fig. 9. Transformed process: After transformations 1 and 4 (a), and then after transformation 2 (b)

In order to verify this impression, the reference process and the candidate processes obtained as result of transformations can be compared to each other, with respect to a set of quality metrics. Depending on the results, the design phase can stop, or a selected candidate (a transformed process) can be substituted as the reference process for the next iteration of the design phase.

4. QUALITY METRICS

There are many metrics to measure various characteristics of software [9,10]. In this research we use simple metrics that characterise the size of a BPEL process, the complexity and the degree of parallel execution. The value of each metric can be calculated using a program dependence graph.

Size of a process is measured by the number of commands in the BPEL program. More precisely, the value of this metric equals the number of leaf nodes in the program dependence graph of this process. For example, the size of a process in Fig. 7a, 8b and 9b is the same and equals 6.

Complexity of a process is measured by a relation of the number of nodes in PDG to the size of the process. For example, the complexity of a process in Fig. 7a equals 1,17 (7/6), while the complexity of processes in Fig 8b is 1,5 and in Fig. 9b is 1,67.

Number of threads is counted by assigning weights to the nodes of the program dependence graph of a BPEL process, starting from the leaves up to the root, according to the following rules:

- the weight of a simple BPEL command is 1,
- the weight of a *<flow>* element is the sum of weights of the descending nodes (i.e. the directly lower nodes in the hierarchy of nesting),
- the weight of a *<sequence>* element is the maximum of weights of the descending nodes (i.e. the directly lower nodes in the hierarchy of nesting).

The value of this metric equals the weight assigned to the top *<sequence>* node of the graph. For example, the maximum number of threads in a process in Fig. 7a is 1, while the maximum number of threads in processes in Fig. 8b and Fig. 9b equals 2.

Length of parallel threads is counted by assigning weights to the nodes of the program dependence graph of a BPEL process, starting from the leaves up to the root, according to the following rules:

- the weight of a simple BPEL command is 1,
- the weight of a *<sequence>* element is the sum of weights of the descending nodes (i.e. the directly lower nodes in the hierarchy of nesting),
- the weight of a *<flow>* element is the maximum of weights of the descending nodes (i.e. the directly lower nodes in the hierarchy of nesting).

The value of the metric equals the maximum weight assigned to a *<flow>* node or is 0, if such a node does not exist in the graph. For example, the maximum length of a thread in a process in Fig. 7a is 0, while the maximum length of a thread in processes in Fig. 8b is 1 and Fig. 9b is 2.

The thread metrics can be used to predict the efficiency of the process execution. One can expect that a higher number of threads will result in faster response time of the service, due to the internal parallelism of execution. Similarly, a higher length of parallel threads can result in faster response of the process.

To justify the later statement, denote the execution time of command *<cmd>* by *cmd*. The execution time of the process in Fig. 8b is now:

$$t_1 = receive + \max[invoke_1, invoke_2] + \max[assign_1, assign_2] + reply$$

while the execution time of the process in Fig. 9b equals:

$$t_2 = receive + \max[invoke_1 + assign_1, invoke_2 + assign_2] + reply$$

It is easy to see that $t_2 \leq t_1$.

5. CONCLUSIONS

The transformational method for designing business process implementation in SOA, described in this chapter, promotes separation of concerns and allows making

business decisions by business people and technical decisions by technical people. The former relates to the definition of a reference process, which reflects the flow of business process used in an organization and does not take into account the technical characteristics of the execution environment. The latter relates to the design phase, in which the reference process is transformed in order to improve efficiency and benefit from the parallel structure of services in a SOA environment. Other quality features, such as modifiability or reliability, can also be considered.

Transformations exemplified in Section 3 are correct in that they do not change the observable behavior of the reference process. This is a very restrictive assumption, which not always is justified in reality. Therefore, a real challenge is to find a method that would allow small changes to the reference process. The supporting tool would warn the designer that such a change have been made by a transformation and show precisely the consequences. The decision whether to allow or to deny such a change would be made by the human designer.

Acknowledgments. This research was supported in part by the Ministry of Science and Higher Education under the grant number 5321/B/T02/2010/39.

REFERENCES

- [1] OMG, *Unified Modeling Language (OMG UML): Superstructure, version V2.1.2*, November 2007, <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF>.
- [2] OMG, *Business Process Modeling Notation (BPMN)*, <http://www.omg.org/spec/BPMN/1.2>.
- [3] ANDREWS T. et al, *Business Process Execution Language for Web Services, Version 1.1*, 2003, <http://www.ibm.com/developerworks/library/specification/ws-bpel/>.
- [4] ERL T., *Service-oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, Englewood Cliffs, 2005.
- [5] Weiser M., *Program slicing*, IEEE Trans. Software Eng., Vol. 10, No. 4, 1984, 352–357.
- [6] BINKLEY D., GALLAGHER K.B., *Program slicing*, Advances in Computers, Vol. 43, 1996, 1–50.
- [7] OTTENSTEIN K.J., OTTENSTEIN L.M., *The program dependence graph in a software development environment*, Proc. ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments, ACM, 1984, 177–184.
- [8] MAO C., *Slicing web service-based software*, IEEE International Conference on Service-Oriented Computing and Applications, IEEE, 2009, 1–8.
- [9] McCABE, T.J., *A Complexity Measure*, IEEE Trans. Software Eng., Vol. 2, No. 4, 1976, 308–320.
- [10] PARIZI R. M., AZIM A., GHANI A., *An ensemble of complexity metrics for BPEL web processes*, Proc. ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE Computer Society, 2008, 753–758.

Ilona BLUEMKE*, Wojciech KIERMASZ *

EXPERIENCE IN SOA BASED INGEGRATION OF SYSTEMS

An real integration, based on the service oriented architecture (SOA), of two medical system is presented. The web services exposed by the systems are used to integrate them. A brief description of integrated systems is also given. An integration architecture is proposed. The integration is based on Publish integration pattern. BPEL processes, necessary in the integration, are graphically presented. Some technical details concerning the implementation are also given. The weak and strong points of the SOA based integration are pointed out.

1. INTRODUCTION

There are several definitions of service oriented architecture (SOA). OASIS [2] defines SOA as the following: *A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.* Definition of SOA can be also found in SOA Manifesto [15].

A system based on a SOA provides functionality as a suite of interoperable services that can be used within multiple, separate systems from several business domains. SOA also generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units,

* Institute of Computer Science, Warsaw University of Technology, Nowiejska 15/19, 00-665 Warsaw, Poland.

or services [6,7,9], which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services. The services should be precisely defined. Usually WSDL [12], which is an XML-based specification schema, is used for the description of a Web service”.

SOA can also simplify interconnection to – and usage of – existing IT (legacy) systems. The SOA approach to integration is presented in [7,9]. In this chapter an integration of COSMIC [4] system with CSAM® Plexus™ [5] system, performed by one of the coauthors, is described. The organization of this chapter is as follows. First, some information about Plexus system and its architecture are given. Next, integration patterns which can be used in the integration with Plexus system, are listed. A brief description of COSMIC system and Collaboration Portal is also given. The integration architecture is shown. The *Publish* integration pattern used in integration and the BPEL processes performing the integration are described. Finally advantages and disadvantages of the SOA based integration are given.

2. CSAM PLEXUS SYSTEM

CSAM (Clinical Systems All Managed) company was established in 2005 to integrate systems of Rikshospitalet University Hospital HF and oncology hospital Radiumhospitalet. The integration was fully successful and as a result CSAM® Plexus™ [5] system was built. It is an information and service collaboration tool designed to integrate existing applications and information within and between hospitals and healthcare professionals.

CSAM® Plexus™ collects, transform and combines information from different data sources into a common information model through a coherent service layer. CSAM® Plexus™ Portal offers single seamless access to information from all integrated clinical systems, across departments and institutions. All necessary information is updated and available to the healthcare professionals connected to the patient, at all times. System allows to introduce new and change old legacy health information systems while not affecting the overall cross-departmental and cross-institutional processes. The IT-systems are coordinated through a web based portal which gives the healthcare professionals quick access to all relevant patient information such as medical history and state, x-rays, laboratory results, investigation results, planned and executed activities, critical information, such as allergies, etc.

2.1. PLEXUS ARCHITECTURE

In Fig. 1 the architecture of Plexus system is presented. The yellow rectangle covers the parts responsible for the integration with external systems. Integration is described in more details in section 3. Following the SOA principles, CSAM Health is using HL7v3 [8] messaging standard to build interfaces for both cross application communication and for application specific services.

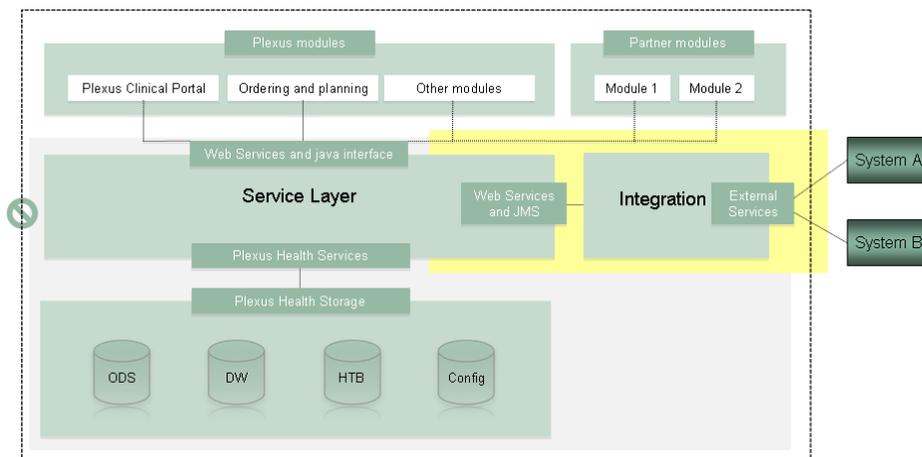


Fig. 1. Plexus architecture

2.2 INTEGRATION PATTERNS

Patterns were introduced by Alexander in 1977 for architecture and in mid nineties were proposed to use in software. A design pattern is a way of reusing abstract knowledge about a problem and its solution. A pattern is a description of the problem and the essence of its solution. It should be sufficiently abstract to be reused in different settings. There are many pattern for SOA described in [9,14,16].

One of the main CSAM® Plexus' functionalities is publishing information by web services, so for this requirement *Publish* pattern may be used. *Publish/Subscribe* [3] pattern helps keep cooperating systems synchronized by one-way propagation of messages because one publisher sends a message to any number of intended subscribers. In CSAM® Plexus HL7v3[8] message standard for *Publish* pattern is used, so the messages could be easily "consumed" by receivers. Plexus is also an integration engine for different medical systems so messages from one system could be distributed to others. For such requirement the *Subscribe* pattern was adopted. To facilitate the integration with external systems two other integration patterns were designed and implemented: *Outgoing Request-Reply* and *Incoming Request-Reply*. These integration pattern are described in [11].

3. INTEGRATION OF COSMIC SYSTEM

The CSAM@Plexus system was integrated with COSMIC (Compliant Open Solutions for Modern Integrated Care) [4] in Västmanland province of Sweden. One of the coauthors of this chapter was in the integration team. COSMIC system contains several modules supporting the medical care of patients. The goal of the integration was to notify the COSMIC system about all changes in critical patients information introduced by other physicians.

3.1 COLLABORATION PORTAL

The goal of *Collaboration Portal* is to input patients' critical information which must be taken into consideration while making decisions concerning the treatment and medicines dedicated to this patient. In Sweden the critical information are divided into three categories:

- Observanda (ang. *Observation*),
- Smitta (ang. *Contamination*),
- Varning (ang. *Warning*).

Observation contains patient' information which should be considered while choosing appropriate treatment e.g. allergy, implants. In Fig. 2 a screen with exemplary critical information is shown. *Contamination* contains information e.g. about the patients' blood deficiencies. *Warning* contains details about the reaction of the patient to medicines and other artifacts used in the therapy classified with Anatomical Therapeutic Chemical (ATC) [1] Classification System [1] (ATC codes).

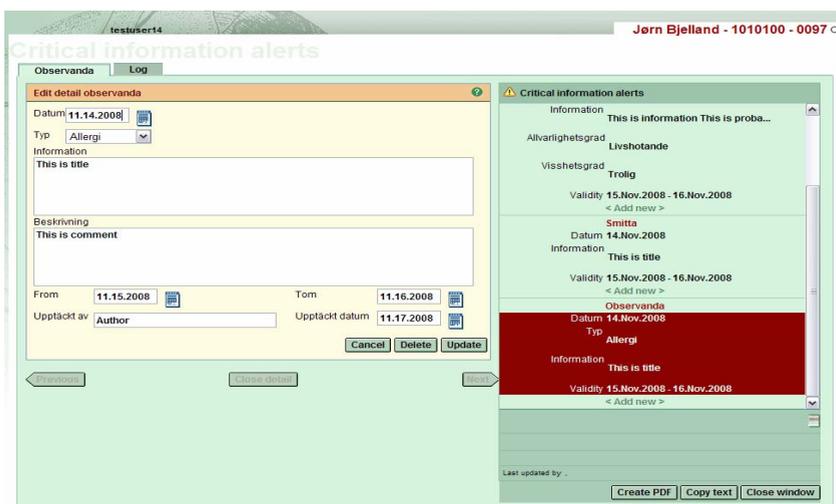


Fig. 2 Screen from *Collaboration Portal*

The newly added critical information contains following fields: ID automatically assigned identifier, author of information, period of observation, comments, title of inserted element and ATC code for *Warning* and type for *Observation*. The name of user making the modification, the date and the time are also stored (Fig. 2). If the critical information is deleted, the reason of the deletion must be given as well.

3.2 INTEGRATION ARCHITECTURE

The integration architecture is shown in Fig. 3.

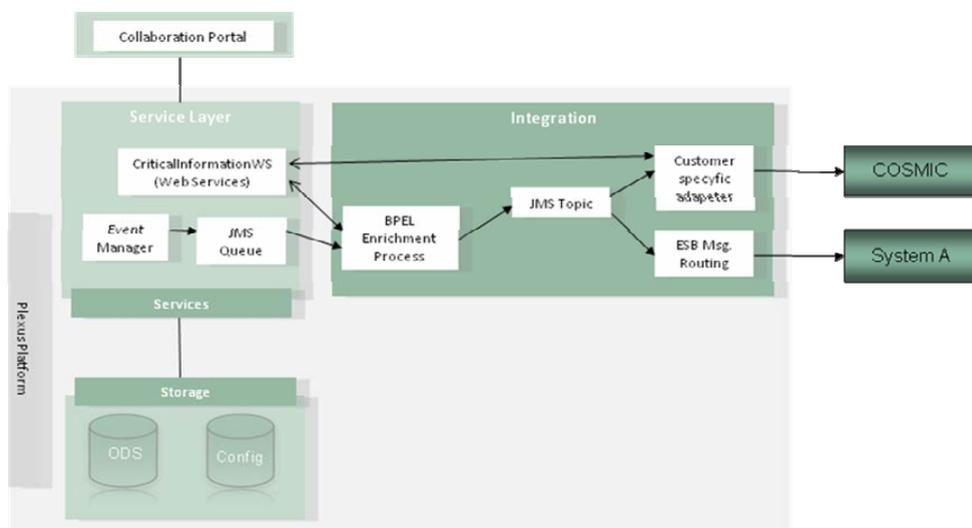


Fig. 3 Integration of COSMIC system an architecture

The integration architecture is based on *Publish* integration pattern (section 2.1, [9, 11]). The typical use case is started by modify/add/delete critical patient information. Appropriate service (from PLEXUS service layer) stores information in data store - *Operational Data Store (ODS)* and notifies the *Event Manager*. It checks, if the information should be send to other systems. Then, the *Event Manager* creates message and puts it in communication channel – *JMS [10,13] Queue*. This channel is still in the service layer and joins it with the integration module. The next flow of activities is following:

1. BPEL [2] process monitors the *JMS Queue* and finds new message.
2. If some additional data are necessary to create the HL7v3 message the BPEL Process calls appropriate web services available in service layers and collects missing data.
3. Complete HL7v3 message is written in w *JMS Topic*.

4. BPEL process, also an adapter for COSMIC system, takes the message from *JMS Topic* and transforms it (from HL7v3) into the format acceptable for the interface of COSMIC system.
5. BPEL process passes the message to COSMIC system.
6. COSMIC system answers:
 - If an error is detected, the information describing it is sent to the service layer (service *CriticalInformationWS*).
 - If the new critical information was successfully added its identifier is passed to the service *CriticalInformationWS* in the service layer.

Service layer of CSAM® Plexus system provides web service *CriticalInformationWS* with synchronic operation *retriveItem*. The call of *retriveItem* (with identifier of critical information as an argument) returns complex structure *retrievelItemResponseElement* (described in [11]) containing patients' critical data.

From HL7v3 [8] standard for passing the critical data *Allergies Intolerance Concern* schema was chosen (described also in [11]). Received from web service *CriticalInformationWS* data are transformed in BPEL process into HL7v3 message.

3.3 IMPLEMENTATION OF INTEGRATION

The implementation of integration was on Oracle platform. BPEL processes were created in JDeveloper 10.1.3 environment. The first step in building BPEL process was the definition of partners. Partners are defined by *partnerLinks*, *role* and *portType* in WSDL [12]document for the service. Next, the variables used in the process were defined. Messages defined in WSDL document were the source of variables. Variables are used by the process as in/out parameters of Web services and internally to store some data. Finally the activities describing the process execution were defined. One of prepared BPEL process named *CollabPortalPublishCI* is show in Fig.4.

The instance of this process is created when activity, type *Receive*, named *ReceivePublishEvent* gets a message from *JMS Queue*. Next steps are as follows:

- Activity – type *Assign*, named *InitRetrieveCICall*, based on received from *JMS Queue* message creates message for web service *CriticalInformationWS*
- Activity – type *Invoke*, named *CallRetrieveCI* synchronically calls operation *retriveItem* of service *CriticalInformationWS* and as a result gets description of patient' critical information.
- Activity – type *Assign*, named *Assign_Data*, fills created HL7 message with data taken from *JMS Queue*.
- Activity – type *Transform*, named *Transform_ToHL7*, using defined transformation file *xslt* transforms data from the critical information into the HL7v3 standard.
- Activity – type *Invoke*, named *PublishCI*, puts HL7v3 message in communication channel *JMS Topic*.

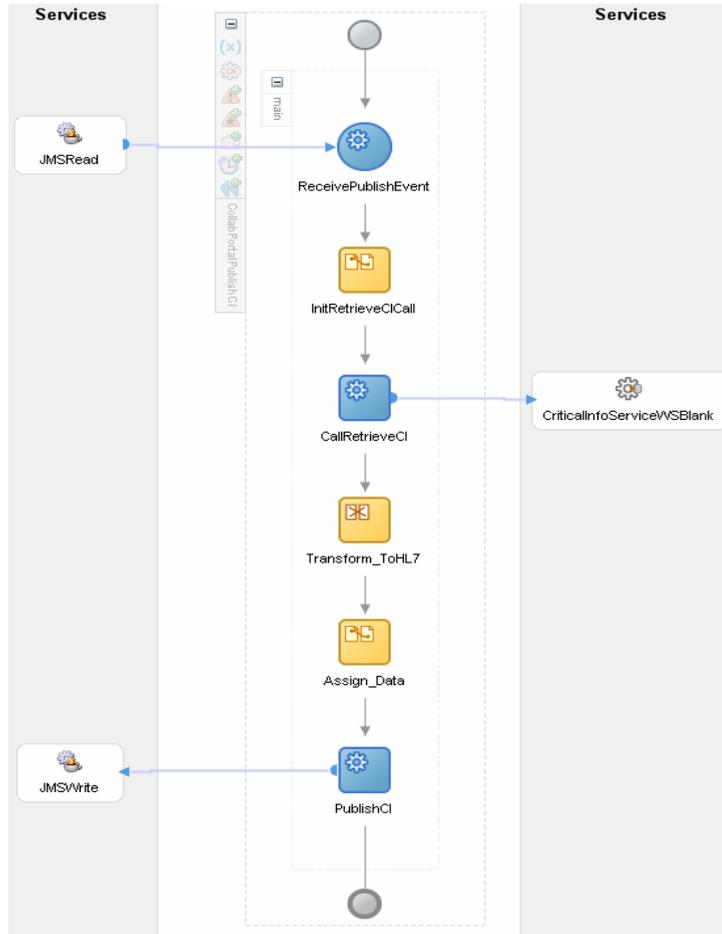


Fig. 4 BPEL *CollabPortalPublishCI* process

In communication with JMS the BPEL process is using Oracle adapters so it should be deployed in Oracle BPEL engine. WSDL document of *CriticalInformationWS* web service is used in communication of BPEL process with this service. Messages send to (received from) this service are constructed (understand) based on this document too. The HL7v3 message is created according to a schema *xsd* in attached to this process definition files.

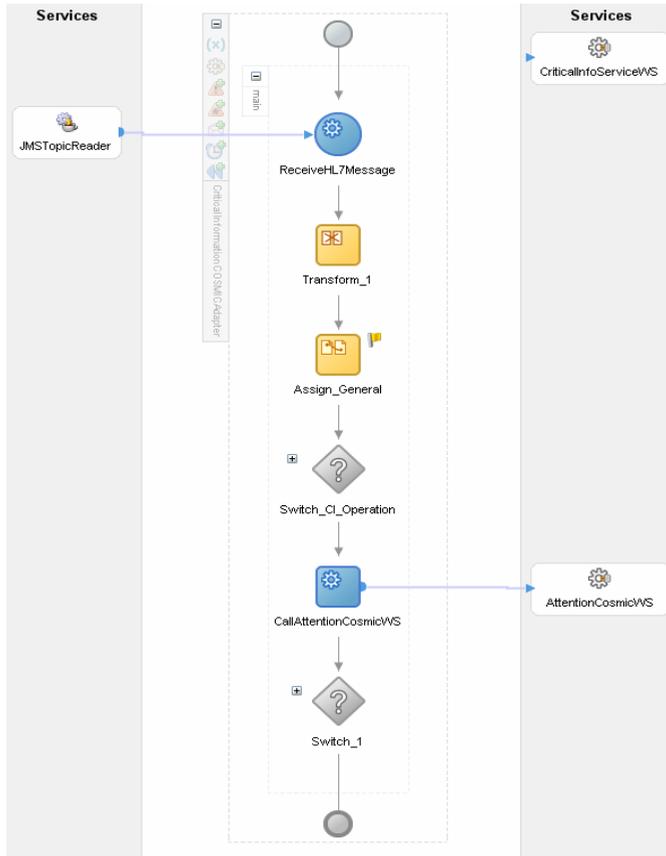


Fig. 5 BPEL *CriticalInformationCOSMICAdapter* process

Another implemented BPEL process (*CriticalInformationCOSMICAdapter*) presented in Fig.5, transforms a HL7v3 message into the format of COSMIC system. This process is also responsible for the answers generated by COSMIC system. The decision block from Fig. 5 (after activity *Assign_General*) is shown in Fig. 6. The first path is executed if new critical information is added and as a result the identifier assigned by COSMIC is given. Then, the operation *updateSourceId* of web service *CriticalInformationWS* is called to store this identifier. The second path is used in case of an error to notify *CriticalInformationWS*. The handling of COSMIC answer is presented in Fig 7. Proces *CriticalInformationCOSMICAdapter* also should be deployed on Oracle platform.

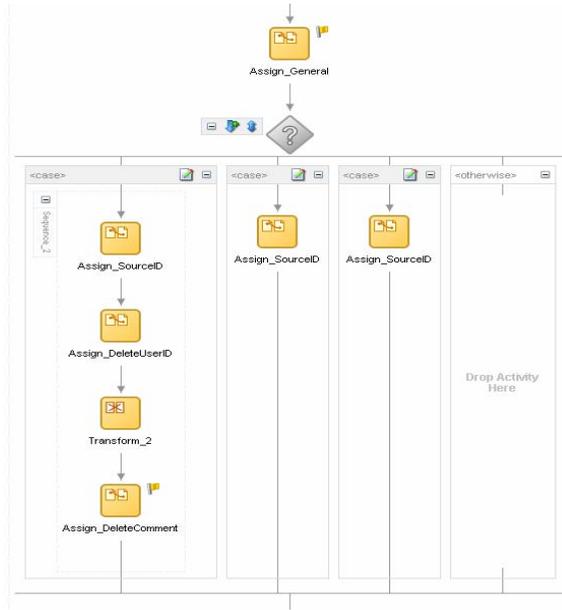


Fig. 6 Decision block (new critical information, modification, removal)

Communication channels (*JMS Queue* and *JMS Topic*) used in the integration are deployed on Oracle data base server. Oracle Web Service Manager was used to improve the management of web services. It can be seen as a simple directory of services. All created BPEL processes are calling web services by URL Web Service address and the name of service. The Web Service Manager was passing query to appropriate service. If the service changes its location only small changes in the configuration of Web Service Manager are necessary and there is no need to change anything in BPEL process.

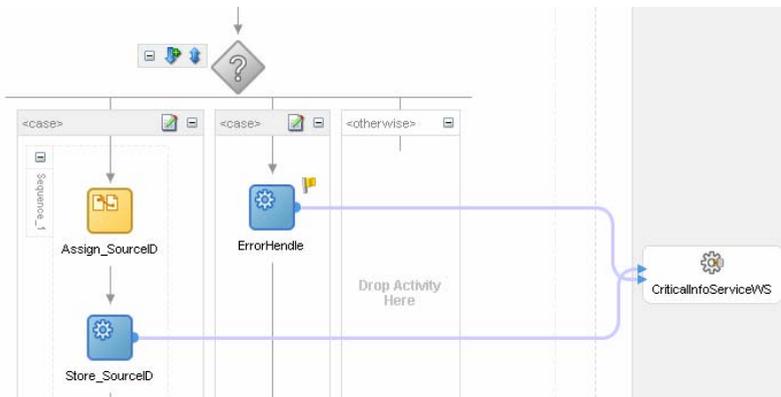


Fig. 7 Handling the operation *setAttention* in service *AttentionCosmicWS*

4.CONCLUSIONS

In this chapter an integration of COSMIC system with CSAM® Plexus system, implemented by one of the coauthors, based on SOA is described. An integration architecture and integration pattern were proposed. However the integration was implemented on Oracle platform the solution is so general, that can be without any difficulties moved to other platform supporting message communication, BPEL engine and ESB e.g. IBM WebSphere, MQ Series.

The integration was based on web services, one of SOA elements, exposed by integrated systems. The loosely coupled services made the integration simple. It was possible to integrate systems working on different platforms without special adapters or bridges. The problems may be caused with the common structure of information passed between services. In this integration the HL7v3 [HL7] standard was used. The universal format for all messages made the messages very complex. The fields of common structure message were used only in small percentage. Communication with complex messages decreased the efficiency of communication. This phenomenon is especially “painful” in Oracle platform, working with high efficiency if the messages are small.

Other SOA element which can be used in integration is service composition. In the above described integration BPEL processes were used. Composition was only used in the implementation of integration logic. Additionally JMS [10,13] communication channels were used for queuing messages so the BPEL engine was not overloaded with queries. JMS was also used for publishing messages to subscribes, such functionality is not present in BPEL.

The implemented integration of CSAM® Plexus and COSMIC systems revealed the usability of SOA based integration especially for systems working on different platforms, in different domains. The implementation revealed also immaturity of several tools used in the integration and the inconsistency in coding SOAP messages. Problems were also with specific for each system identifiers of the same data. To solve this problems mappings tables were built. SOA based integration is not solving one of important integration problems such as inconsistency in data models of integrated systems. SOA based integration prevails previous integration solutions but integrated systems must provide interface in the form of web services WS.

REFERENCES

- [1] ATC: http://www.whocc.no/atc/structure_and_principles/ (access July 2011).
- [2] BPEL Standard, <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>, (access July 2011).
- [3] BUSCHMANN, F. at all., *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns.*, John Wiley & Sons Ltd, 1996.
- [4] COSMIC system: <http://www.cambio.se/document/sv-se/Technical%20Overview.pdf> (July 2011).

- [5] CSAM Plexus: www.csamhealth.com (access July 2011).
- [6] ERL T., *Service-Oriented Architecture (SOA)*.: Prentice Hall, 2008.
- [7] den HAAN J., *SOA Approach to Integration*, 2009.
- [8] HL7standard: <http://www.hl7.org> (access July 2011).
- [9] HOHPE G., WOOLF B.: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Boston: Addison-Wesley Professional, 2005.
- [10] JMS: <http://www.oracle.com/technetwork/java/jms/index.html> (access July 2011).
- [11] KIERMASZ W.: *Integration of information systems in SOA*, Msc thesis, Institute of Computer Science, Warsaw University of Technology, 2009.
- [12] PAPAZOGLU M.P.: *Web services principles and technology*, Harlow, 2008.
- [13] RICHARDS, M.; MONSON-HAEFEL R., CHAPPELL D. A., *Java Message Service, Second Edition*. O'Reilly. ISBN 978-0-596-52204-9, 2009.
- [14] ROTEN-GAL-OZ, A., *SOA patterns*, 2009.
- [15] SOA manifesto: <http://www.soa-manifesto.org> (access July 2011)
- [16] SOA patterns: www.soapatterns.org, (access July 2011).

Szymon KIJAS*, Andrzej ZALEWSKI*,
Krzysztof SACHA*, Marcin SZLENK*, Andrzej RATKOWSKI*

FORMAL SEMANTICS OF ARCHITECTURAL DECISION MAKING MODELS AS A COMPONENT OF AN INTEGRATED EVOLUTION METHODOLOGY FOR SERVICE-ORIENTED SYSTEMS

Service-oriented architectures address problem of systems evolution by providing means for composing new functionalities out of services provided by already existing systems, or by changing the implemented functionality by rearranging and/or enhancing services composition. System changes are usually carried out according to organisational procedures described in ITIL/ISO 20000 standard. However, there is currently no mature evolution methodology for service-oriented systems that would integrate software development with IT operations management (esp. change management). This is a challenge that large organisations are forced to resolve ad hoc. The methodology we are working on is aimed at bridging this gap by the integration of change management process, engineering methods, models and supporting tools. Architectural decisions are supposed to become a vehicle used to represent the architectural knowledge documenting system's evolution and architects' intent. Existing models of architectural decisions define information content and intuitive semantics only. This causes that the concepts of architecture decision making are rather vague and difficult to verify or validate. Formal semantics should help to resolve such issues and make the concepts comprehensible and well-founded.

1. INTRODUCTION

Modern software intensive systems never reach a stable state in which they are subject to maintenance only. They are always subject to perpetual changes caused by changing business requirements. The effective evolution of software intensive systems

* Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland.

becomes a primary concern of modern organisations as it is often a necessary condition for achieving a competitive advantage. It is also common knowledge nowadays that the cost of evolution is at least comparable, and in many cases exceeds, the cost of initial systems development.

Service-oriented architecture addresses the challenge of adaptation to rapidly changing requirements by offering system construction out of a set of independent, distributed and loosely coupled components orchestrated into business processes. However, there are currently no mature evolution methodologies dedicated to the evolution of SOA systems. The development of such a methodology is the research challenge of our project. An important component of such a methodology are architectural decisions, which will be used to represent the flow of system evolution. Models of architectural decisions are informal in nature, vague and ambiguous. Formal semantics could help to resolve these issues, enabling formal analysis and verification.

2. OUTLINE OF AN INTEGRATED EVOLUTION METHODOLOGY FOR SOA MOTIVATION

Changes to IT systems are usually managed with change management processes defined in ITIL or ISO 20000. They define the change management process aimed at ensuring that all changes will be “assessed, approved, implemented and reviewed in a controlled manner” [1]. However, standards define what should be done, but not how.

On the other hand, existing development methodologies have not been sufficiently verified in industry, or are not sufficiently supporting systems evolution:

- Semiformal analysis and design methods dedicated for SOA systems like [2], [3] are supposed to become entirely new approaches aimed mainly at SOA systems development being not fully verified by the industry;
- Formal approaches like [4], [5], [6] address selected issues like services orchestration or choreography and similarly to the other formal software engineering methods are of a limited industrial use;
- Traditional object-oriented methodologies [7], [8], proven practically in a number of large projects, are targeted mainly at systems development, and only partially support systems evolution;
- Agile approaches [9] perceived as an alternative to the object-oriented rely on knowledge capturing and sharing by person-to-person communication. The lack or very limited range of formal documentation may become a bottleneck in the case of systems evolution lasting at least a couple of years.

Therefore, there is a gap between existing change management processes indicating what should be done to change the systems in a controlled way and SOA technology developed to support rapid systems evolution. Our research is aimed at filling this gap with a change-oriented (evolution-oriented) methodology dedicated for SOA systems.

The evolution of SOA systems is about extending them with new services and processes, and/or making changes in the structure and operation of processes and, the resulting changes in the services. The evolution of SOA systems consists of steps resulting from gradual changes made under the change management process [1]. The elaborated methodology would support the following essential elements of the change management process:

- request for change (RFC) referring to the existing system;
- change assessment – the evaluation of the impact of a change on the rest of the system in terms of affected services and processes and the impact on quality attributes such as performance, security, modifiability;
- change implementation;
- change documentation;
- change deployment.

In the proposed methodology the evolution consists of a series of consecutive changes. Each of them is seen as a transformation of a set of system models and system itself leading from the current state of the system to the one where the change is applied. This set of models should at least include the following: analytical models of business processes expressed in BPMN and/or UML, design models of business processes expressed in BPMN or as orchestration code in BPEL, and the set of architectural decisions describing the system architecture [10], [11]. The set of models can be seen both as documentation of the system or – in view of the loose relations between services – as a description of the system configuration.

The evolution-oriented methodology will consist of:

- In the area of the change specification: significant changes in the service-oriented system are changes in the business processes of an organisation that adapts these processes to changed or extended requirements. In this subject the methodology anticipates the use of techniques of analytical modelling of the business processes in BPMN, as well as techniques of transforming the description of the business process into a description of the service orchestration in BPEL.
- In the area of the change impact assessment: a method of evaluating how the change influences the existing system will be provided (described with a set of models) including the method of the design and evaluation of chosen qualitative features of the software being developed. The basis for achieving this goal will be a variant of GQM [12] that will be elaborated and oriented towards SOA. Security aspects of the service-oriented system will be modelled and designed using the role-based trust management language RTT [13].

- In the area of the change implementation: support for the automatic transformation of the orchestration code in BPEL will be provided. The functionality of the system after the application of such transformations will remain intact, while the qualitative features (such as performance, reliability, etc.) will be changed. This will be achieved by adapting the method for the transformational design of the orchestration code in BPEL proposed in [14]. This method is based on LOTOS language, and thus allows the correctness of the process transformations to be proven.
- In the area of change documentation: a method of modelling changes in the system architecture will be provided, described with a set of architectural decisions. This method would be the elaboration of the MAD (Maps of Architectural Decisions) model proposed in [15]. Such models will also become a tool for sharing information about the system architecture. This is the focus of the rest of the paper.

3. MOTIVATION AND RELATED WORK

Architectural decisions [10], [11] are often perceived as another wave in architecture modelling – compare “third epiphany” in [19]. However, they are still represented as text records [11], [17], [18], sometimes accompanied with illustrating diagrams [20]. The limitations of textual models are well-known in the genre of software engineering. Therefore, sets of hundreds of architectural decisions necessary to sufficiently represent architecture of a large system, are difficult to comprehend, analyse, verify and ensure completeness and consistency or even just to navigate through them.

Diagrammatic (based on graphs) models have been proposed as a way to represent architectural decision and decision making process comp. [16], [24], [15], [25] in a more comprehensible way. Graphical models and tools supporting architectural decision and decision making have been presented in [26], [25], [24].

The classifications have been developed to help to organise large sets of architectural decisions. Most influential classifications by Kruchten [18] (existence, non-existence, property and management ADs) and Zimmermann [16] (executive, conceptual, technology, vendor asset ADs) substantially help to navigate through a set of ADs. However, these categories are not always precise, and in many cases can confuse engineers.

Relations between architecture decisions should help to navigate through and make architectural decisions traceable by representing the dependencies between them. In [18] Kruchten indicates ten kinds of relations between ADs. Thus, it could take some time to learn how to recognise each of these kinds.

The most advanced model described in [16] combines classifications, relations and decision making models. Architectural decisions are classified using the following

structure: The "Topic Group" entities have been defined on the first level of the model. They classify another "Topic Groups" as subgroups on the next levels (further subgroups can still appear on many levels). Actual decision problems - "Issues" can only be connected to the lowest "Topic Group" in the hierarchy (the most detailed).

Decision making process is represented with "Issues" and "Relations". "Issues" are connected with possible solutions ("Alternatives"). Result of decision making process is represented as "Outcome", which is assigned to "Issue" and indicates selected solution ("Alternative") of solved decision problem.

Moreover, relations can have been defined for this model: influences, refinedBy, decomposesInto, forces, isIncompatibleWith, isCompatibleWith, triggers, hasOutcome. For complete reference – see [16].

Another diagrammatic notation MAD 2.0 [25] has been developed by our team. MAD 2.0 has been developed to support architect-practitioners working on systems evolution. It does not impose any predefined classification or hierarchy of architectural decisions and assumes a limited number of kinds of relations between architectural decisions. This makes a model of the decision process intuitive and easy to comprehend. To explain the choices made and capture their rationale, the entire decision situation is presented, including: the decision topic, considered design options, relevant requirements, the advantages and disadvantages of every considered option.

Even the most advanced approach [16] is founded on intuitive semantics of applied model. It is missing a rigorous semantics making its meaning vague and subject to individual interpretation.

4. SEMANTICS OF ARCHITECTURAL DECISIONS MAKING MODEL – THE CHALLENGE

Formal semantics of architecture decision making models would:

- define its concepts in a systematic, rigorous and formal way;
- eliminate possibility of wrong use of relations and inconsistency.

We are developing a framework of denotational semantics of architectural decision making models. Denotational semantics map language expressions directly on to their meaning. In our case the declarative specification language Alloy [21] is supposed to be used. It is a formal language based on the first-order logic.

There exists a software tool Alloy Analyser [21], which can be used to analyse properties of the models defined in Alloy. This is a piece of software which combines functionality of "model checker" and "theorem prover" tools. Diagram generated using Alloy Analyser is presented in fig. 1.

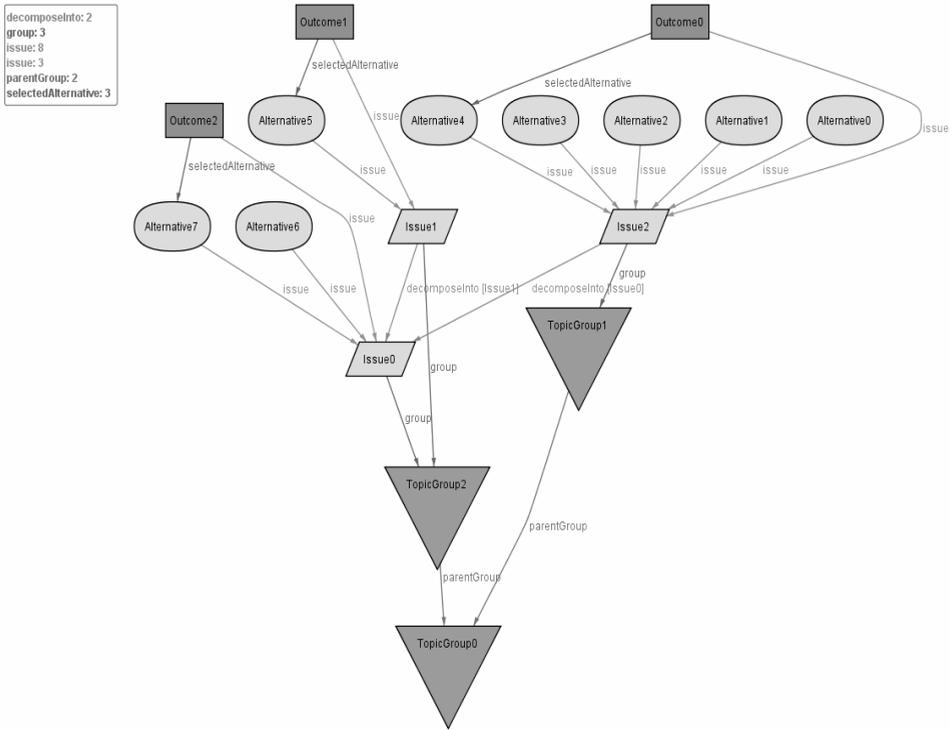


Fig. 1. Diagram generated using Alloy Analyser

The choice of Alloy has been inspired by the construction of the language and its similar applications [22], [23]. A mapping example of model [16] to Alloy representing some of its basic properties is presented in fig. 2:

- Entities of model [16] ("Topic Group", "Issue", "Alternative", "Outcome") have been defined using Alloy's key word "Sig".
- Classification of "Topic group" entities has been defined using "sig TopicGroup { *parentGroup : lone TopicGroup* }" source code (selected by italic font). It means that "Topic Group" could have one parent "Topic group" but it doesn't have to.
- "sig Issue { *group : one TopicGroup*, *decomposeInto : Issue lone -> Issue* }" – italic source code means that "Issue" entity have to be assigned to one and only one "Topic Group" - classification of "Issues".
- "sig Issue { *group : one TopicGroup*, *decomposeInto : Issue lone -> Issue* }" – italic source code (selected by italic font) defines properties of decomposeInto relation. It means that one "Issue" can be in a relation with many other "Issues", but doesn't have to – one issue could be decomposed into some more detailed ones.

- “sig Alternative { *issue : one Issue* }” – italic source code means that “Alternative” have to be assigned to one and only one “Topic Group” – “Issue” can have some variants of their solution.
- “sig Outcome { *issue : one Issue* , selectedAlternative : Alternative}” – italic source code means that “Outcome” have to be assigned to one and only one “Issue” – Decision problem (“Issue”) can have solution.
- “sig Outcome { *issue : one Issue* , *selectedAlternative : one Alternative*}” – italic source code means that “Outcome” have to be assigned to one and only one “Alternative” – selected solution of a decision problem (“Issue”).

```
//Definition of "TopicGroup"
sig TopicGroup { parentGroup : lone TopicGroup }

//Definition of "Issue" with decomposeInto relation
sig Issue { group : one TopicGroup, decomposeInto : Issue lone -> Issue }

//Definition of "Alternative"
sig Alternative { issue : one Issue }

//Definition of "Outcome"
sig Outcome { issue : one Issue , selectedAlternative : Alternative}
```

Fig. 2. Model mapping [16] into Alloy sentences

Semantics will be aimed at formalising and verifying inference rules for the relations between entities of model [16]. First the full mapping (including all types of relations) of the model concepts on to Alloy has to be created. Then the rules for relations between entities will be developed (using features of Alloy language). Finally correctness of semantics using Alloy Analyser tool can be verified.

Examples of "facts" (in Alloy) with the help of which will be developed semantics of model [16] have been presented on fig. 3:

- “no tg: TopicGroup | tg in tg.parentGroup” – means that none TopicGroup can't be its own parent.
- “lone tg1:TopicGroup | all tg2:TopicGroup | tg1 in tg2.parentGroup” – means that TopicGroup can have only one parent.
- “one tg: TopicGroup | no tg.parentGroup” – means that one and only one TopicGroup has to be the root TopicGroup (TopicGroup without parent).
- “all disj tg1, tg2: TopicGroup | tg1 in tg2.parentGroup => tg2 not in tg1.parentGroup” – means that all of pairs of "TopicGroups" have to satisfy the condition: If TopicGroup tg1 is the parent of TopicGroup tg2 then TopicGroup tg2 can't be the parent of TopicGroup tg1 (child can't be the parent of its own parent).
- “all is:Issue | all tg:TopicGroup | is.group not in tg.parentGroup” – means that “Issue” can't be connected to “TopicGroup” which is the parent of other “TopicGroups”.

- “all is : Issue | lone o : Outcome | o.issue = is” – means that "Issue" can have one and only one "Outcome", but don't have to.
- “all o : Outcome | o.issue = o.selectedAlternative.issue” – means that "Outcome" can be only connected to "Alternative" assigned to the same "Issue".
- “no is: Issue | is in is[decomposeInto].Issue” – means that "Issue" can't be in relation with oneself.
- “all disj is1, is2: Issue | is1 in is2[decomposeInto].Issue => is2 not in is1[decomposeInto].Issue” – means that all of pairs of "Issues" have to satisfy the condition: If "Issue" is2 is the sub-"Issue" of "Issue" is1 then "Issue" is1 can't be sub-"Issue" of "Issue" is2 (decomposed "Issue" can't be sub-"Issue" of own sub-"Issue").

```

fact {
no tg: TopicGroup | tg in tg.parentGroup

lone tgl:TopicGroup| all tg2:TopicGroup | tgl in tg2.parentGroup

one tg: TopicGroup | no tg.parentGroup

all disj tg1, tg2: TopicGroup | tg1 in tg2.parentGroup
                               => tg2 not in tg1.parentGroup

all is:Issue | all tg:TopicGroup | is.group not in tg.parentGroup

all is : Issue | lone o : Outcome | o.issue = is

all o : Outcome | o.issue = o.selectedAlternative.issue

no is: Issue | is in is[decomposeInto].Issue

all disj is1, is2: Issue | is1 in is2[decomposeInto].Issue
                          => is2 not in is1[decomposeInto].Issue

}

```

Fig. 3. Example of “facts” for model [16] in Alloy

5. SUMMARY

As software system evolution becomes a primary concern, engineering approaches should address this issue efficiently. It requires the integration of design techniques with a change management process defined by system operations standards. The proposed methodology is supposed to integrate these two currently distinct areas.

One of its components are supposed to be architecture decision models with formal syntax and semantics. Formal semantics of architectural decisions and decision making models:

- will provide a clear, unambiguous meaning of the models
- will enable validation of the concepts comprising architecture decision and decision making models
- will enable automated verification of model instances.

This work was sponsored by the Polish Ministry of Science and Higher Education under grant number 5321/B/T02/2010/39.

REFERENCES

- [1] ISO/IEC 20000-1:2005 and 20000-2:2005, *Information technology – Service management*, ISO 20000-1: Specification. ISO 20000-2. Code of practice. ISO/IEC 2005.
- [2] BELL, M., *Service Oriented Modelling. Service analysis, design and architecture*, John Wiley and Sons, 2008.
- [3] ARSANJANI, A., Service-oriented modeling and architecture, IBM, <http://www.ibm.com/developerworks/webservices/library/ws-soa-design1>, 9 November 2004.
- [4] CAMARA, J., et al., *Formalizing WS-BPEL Business Processes Using Process Algebra*, *Electr. Notes Theor. Comput. Sci.*, Vol. 154(1), 2006, 159–173.
- [5] FERRARA, A., *Web Services: A Process Algebra Approach*, In ICSOC '04: Proceedings of the 2nd International Conference on Service oriented computing, New York, NY, USA, ACM Press, 2004, 242–251.
- [6] FOSTER, H.: et al., *Model-Based Verification Of Web Service Compositions*, In 18th IEEE International Conference on Automated Software Engineering (ASE), 2003.
- [7] BOOCH, G. et al., *Object-oriented analysis and design with applications*, third edition, Addison-Wesley Professional 2007, ISBN: 9780201895513.
- [8] LARMAN, C., *Applying UML and Patterns—An Introduction to Object-Oriented Analysis and Design and the Unified Process*, second ed., Prentice Hall, 2002.
- [9] HAZZAN, O., DUBINSKY, Y., *Agile Software Engineering*, 1st Edition., ISBN 978-1-84800-198-5, Springer 2009.
- [10] BOSCH, J., JANSEN, A., *Software Architecture as a Set of Architectural Design Decisions*, 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05), IEEE Computer Society, 2005, 109–120.
- [11] TYREE, J., AKERMAN, A., *Architecture Decisions: Demystifying Architecture*, IEEE SOFTWARE, Volume: 22, Issue: 2, March-April 2005, 19–27.
- [12] BASILI, V. R., et al., *The Goal Question Metric Paradigm*. Encyclopedia of Software Engineering (Marciniak, J.J., editor), Volume 1, John Wiley & Sons, 1994, 578–583.
- [13] CZENKO, M., et al., *An Introduction to the Role Based Trust Management Framework RT*, LNCS 4677, Springer, Berlin Heidelberg 2007, 246–281.
- [14] RATKOWSKI, A., ZALEWSKI, A., *Transformational Design of Business Processes in BPEL Language*, e-Infomatica Software Engineering Journal. Volume 3, Issue 1, 2009.
- [15] ZALEWSKI, A., LUDZIA, M., *Diagrammatic Modeling of Architectural Decisions*. Software Architecture, Second European Conference, ECSA 2008 Paphos, Cyprus, September 29–1 October, 2008 Proceedings. Lecture Notes in Computer Science, vol. 5292, pp. 350–353.

- [16] ZIMMERMANN, O. et al., *Managing architectural decision models with dependency relations, integrity constraints*, and production rules, *Journal of Systems and Software*, vol. 82, no. 8, 2009, 1249–1267.
- [17] HARRISON, N., B., AVGERIOU, P. and ZDUN, U., *Using Patterns to Capture Architectural Decisions*. *IEEE Software*, vol. 24, no. 4, 2007, 38–45.
- [18] ALI BABAR M. et al., *Architecture knowledge management. Theory and Practice.*, Springer-Verlag, Berlin Heidelberg 2009.
- [19] KRUCHTEN, P., CAPILLA, R., DUEÑAS, J., C., *The Decision View's Role in Software Architecture Practice*, *IEEE Software*, vol. 26, no. 2, March/April 2009, 36–42.
- [20] CAPILLA, R., NAVA, F., and DUEÑAS, J., C., *Modeling and Documenting the Evolution of Architectural Design Decisions*, *Proc. 2nd Workshop Sharing and Reusing Architectural Knowledge Architecture, Rationale, and Design Intent*, IEEE CS Press, p. 9., 2007.
- [21] Alloy, <http://alloy.mit.edu/>.
- [22] SUN, J., ZHANG, H., WANG, H., *Formal Semantics and Verification for Feature Modeling*, *Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems*, June 16–20, 2005, 303–312.
- [23] CUNHA, A., PACHECO, H., *Mapping between Alloy Specifications and Database Implementations*, *Software Engineering and Formal Methods*, 2009 Seventh IEEE International Conference, 23–27 Nov. 2009, 285–294.
- [24] MOJTABA SHAHIN, M., LIANG, P., REZA KHAYYAMBASHI, M., *Improving understandability of architecture design through visualization of architectural design decision*, *SHARK '10 Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge*, ACM 2010.
- [25] ZALEWSKI A., KIJAS S., SOKOŁOWSKA D., *Capturing Architecture Evolution with Maps of Architectural Decisions 2.0*, *ECSA 2011*, Essen, Germany, September 13–16, 2011. *Lecture Notes in Computer Science* Vol. 6903, 2011.
- [26] JANSEN, A., AVGERIOU, P. and VAN DER VEN, J., *Enriching Software Architecture Documentation*, *Journal of Systems and Software*, Volume 82, Issue 8, Elsevier, August 2009, 1232–1248.

Paweł STELMACH*, Łukasz FALAS*

SERVICE COMPOSER – A FRAMEWORK FOR ELASTIC SERVICE COMPOSITION

Demand for service composition is increasing, yet there is no single approach to this problem that would be widely accepted. Adding domain knowledge to already numerous algorithms and methods is a source of multiple approaches solving the service composition problem. In Service Composer we propose an architecture based on the idea of Smart Services. We also postulate that the composition itself is performed by the specialized atomic services composed into a Smart Service that is fit for a specific problem and consists of appropriate methods. Service Composer allows for elastic binding of many composition schemes demanding well-defined descriptions of services representing them. Our framework aids in definition of composition services with domain specific selection methods. Also a crucial part of this framework is a Workflow Engine capable of executing composed services as well as composition scenarios themselves. In Service Composer a Workflow Engine is closely integrated with various composition scenarios enabling the user to define how a specific type of a service (or even a requirement) is meant to be executed.

1. INTRODUCTION

Many organizations use information technology to manage their business processes. Almost all have expanded their applications to make use of the Internet to enable better communication with their clients and within the company. However, for many this transformation was a costly step and a need for a more distributed and elastic architecture was a fact determining their existence in the market. In this context, Service Oriented Architecture (SOA) is the application framework that enables organizations to build, deploy and integrate these services independent of the technology systems on which they run [8]. In SOA, applications and infrastructure can

* Wrocław University of Technology, Institute of Computer Science, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.

be managed as a set of reusable assets and services. With the use SOA business can respond faster to market opportunities and get more value from their existing technology assets [9].

The final success of the SOA concept can be obtained if those service enabled applications could be effortlessly developed and integrated by many groups, both internal and external to the organization. The composition of web services enables SOA architects to build composite services performing advanced tasks. In this context tools aiding in that process are necessary to manage such complex systems with least effort. Nowadays in organizations the composition process is mostly manual, however with the increasing number of available services it is unavoidable that architects will need automated methods of selecting services that fulfill their functional requirements and work well together. Besides the obvious software and message compatibility issues a good service composition should be done with respect to the Quality of Service (QoS) requirements. For the client preserving the non-functional requirements (availability, performance and security, etc.) is a key factor [6],[7].

To this date researchers have approached the service composition from different perspectives. Some have presented specialized methods for services selection [4] or composite service QoS-based optimization [5]. However, despite the importance of their contribution, those solutions are not widely used by other researchers. Some propose complete end-to-end composition tools like [1] introducing a concept of two-staged composition: logical composition stage to prune the set of candidate services and then composing an abstract workflow. METEOR-S [2] presents a likewise concept of binding web services to an abstract process and selecting services fulfilling the QoS requirements. Notions of building complete composition frameworks are also clear in SWORD [3], which was one of the initial attempts to use planning to compose web services. However, the proposed approaches are closed and do not support implementation of other methods and, because of this, it is difficult to call them frameworks. And a framework-based approach is what is currently needed in SOA field in order to create composition approaches that are fitted to different domains and problems characteristic for them.

In this work we propose a software framework allowing incorporating various composition approaches and the use of different knowledge repositories like ontologies, social networks, rule engines, etc. It is open for expansion via web service enabled composition methods and aids developers in building, testing and executing elastic composition scenarios. Our aim was to enable knowledge sharing among service providers and their customers and to support automatic composition of services into business processes within a unified software framework.

The remainder of this work is organized as follows: in Section 2 a general composition scenario is introduced. Then in Section 3 the architecture of Service Composer is discussed and in sections 4 and 5 key elements of the framework – Smart Services concept and Workflow Engine – are presented in more detail. Section 5 covers implementation aspects.

2. A GENERAL COMPOSITION SCENARIO

To compose a composite service is to find a set of atomic services and bind them together so that they, as a new service, fulfill all user functional and nonfunctional requirements. Many aspects of service composition are done by hand but researchers always tried to develop more automated approaches, helping in selection of atomic services, connecting them accordingly and watching over passing of data between them. Figure 1 depicts a general approach to service composition. Typical fully automated composition process consists of three stages, however not all of them are necessary in every situation. Those stages are:

- building of a composite service **structure**,
- building of a composite service **scenario**,
- finding an optimal **execution plan** of a composite service.

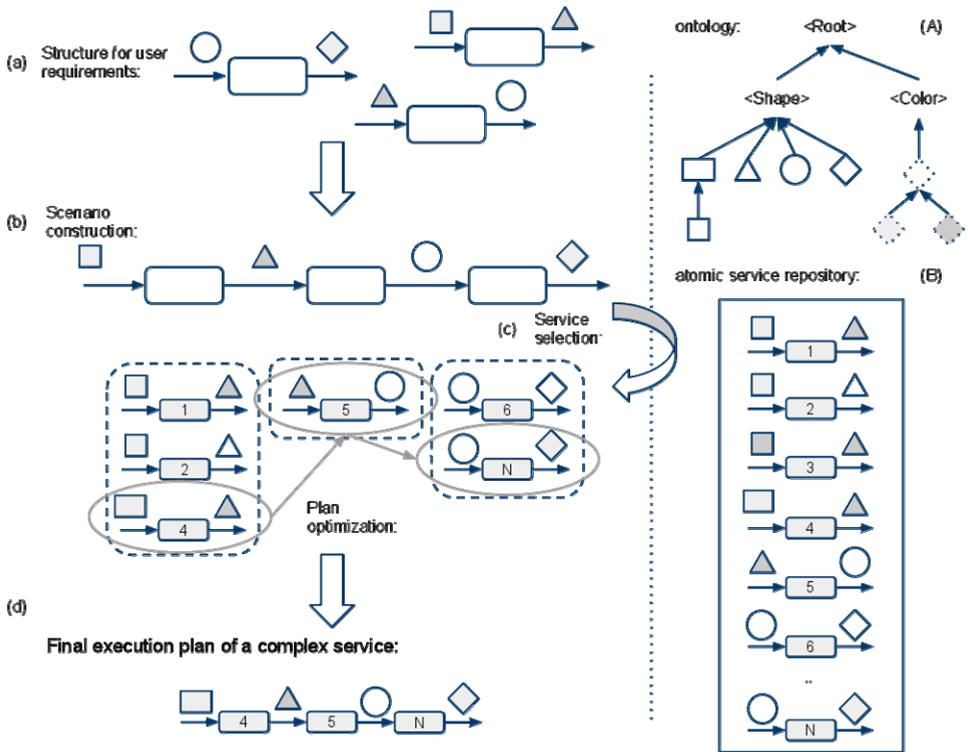


Fig. 1. An example of automated service composition

Inputs and outputs of particular services are described with concepts from the domain ontology (Fig. 1.A) (here: geometric figures) and services descriptions are stored in atomic service repository (Fig. 1.B). The composition is a process of

transforming user requirements (described similarly to services in the repository) into a fully defined composite service that fulfills them. In the first stage of composition process user requirements are analyzed and assembled into a single structure that represents a structure of a final composite service (Fig. 1.a – a structure is represented by a graph where nodes contain user requirements and edges will connect those requirements determining the order in which a final composite service will be executed). Then, with the use of knowledge engineering, this structure is enhanced so it defines a scenario of execution of particular atomic services in a composite service (Fig. 1.b). In this stage no requirement can be disconnected from other requirements. Also in this stage the scenario is filled with candidate services that fulfill each of the requirements according to required functionalities (Fig. 1.c). However, they may differ in non-functional properties (as execution time or cost) and so in the last stage (Fig. 1.d) for each functional requirement a single atomic service is selected, so that all services, that build a composite service, jointly fulfill non-functional requirements.

3. SERVICE COMPOSITION TOOL ARCHITECTURE

Section 2 presents a general composition scenario and indicates that each of the stages could be performed using different methods – beginning with AI Planning methods for producing a complete scenario; then one could use different semantic selection methods when searching for services fulfilling each requirements (or even propose different distance measures for concepts in the ontology); finally, various optimization techniques could be used to produce the composite service fulfilling the non-functional requirements, not to mention that a variety of non-functional parameters could be demanded and optimized by. In some situations not all stages are necessary and sometimes stages can be expanded so they consist of different sub-stages – like combining many different algorithms for creating a composite service scenario or connecting different semantic filters in a sequence to produce results that better reflect user requirements.

All this led to designing a composition framework with an elastic architecture that would allow composition service designers to incorporate various approaches, test them and deploy in a form of service enabled composition tools well fitted to different domains and problems.

Fig. 2 presents Service Composer organization, in which the framework consists of two main parts: one is a front end of the application (2.a) and allows a business client to define his domain by connecting to external service and knowledge repositories (here: ontologies), which will be used to construct composite services; the second part (2.b) is a layer of Service Composer *engine* services which provides a variety of services but mainly composition services.

Both layers of the tool have a similar architecture and consist of three main elements:

- repository of composite services,
- repository of atomic services,
- knowledge repository for provided services (usually domain ontology).

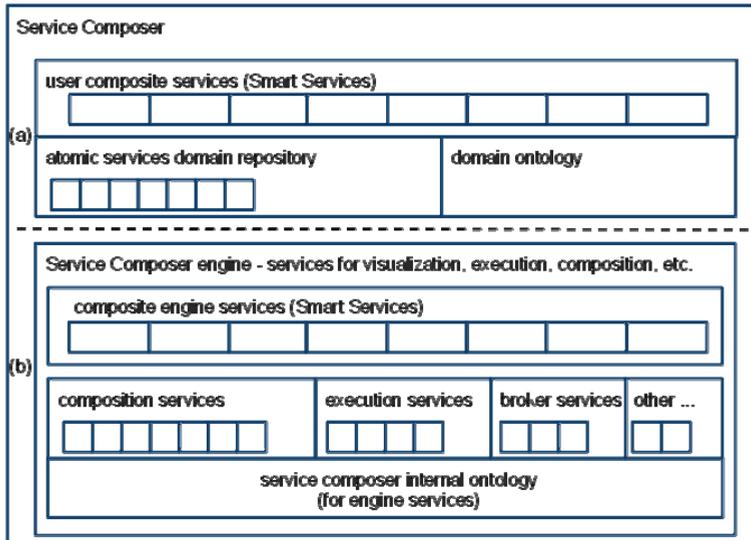


Fig. 2. Service Composer framework organization

The two layer Service Composer architecture is founded on following assumptions:

- first layer is responsible for definition of composite services and triggering their execution and composition in the second layer,
- second layer consists of supporting engine services – like services responsible for composition, execution, finding the best instances of services in distributed system, searching for services composed earlier, personalization etc.,
- all services (from both layers) can be directly called from an external application through SOAP protocol,
- for most of the time the second layer services should be hidden from the business user and called in various situations:
 - directly i.e. after composition request,
 - by graphical user interface while composing a composite service by hand,
 - by the Workflow Engine configured to call them when executing parts of a composite service,

- if it is desired by a business user, he can configure the engine services (second layer) and the Workflow Engine itself (to define the execution scenarios).

Service Composer stores composite services as Smart Services, described in SSDL language – it is an internal language designed specially for composition purposes. Its expressive syntax allows for simultaneous definition of atomic services and requirements of different types in a composite service. It is developed in parallel with the Workflow Engine that supports its broad functionality and knows how to “execute” functionalities defined in the Smart Service. More information on Smart Services and the Workflow Engine can be found in sections 4 and 5.

Users usually will restrain to main functionality of the tool, which is defining new composite services. In this case they will use graphical user interface to input their requirements and appropriate composition services will be executed to produce an optimal composite service. However, a user can define what “appropriate composition services” means in his domain. Configuring the Service Composers’ engine allows the user to choose different composition services for various situations, replace parts of composite composition services or add his atomic services and bind them to existing solutions. All this can be saved in user profile and executed in well-defined situations when a composition service is requested. Especially this functionality supports the notion of Service Composer being a framework for composition of services. Engine services can be delivered as web services by various institutions developing composition methods or other methods supporting the composition process (like composite services search methods, brokering for the best execution, different execution engines, etc.). Afterwards, registered services can be bound with other services in a form of composite services. Finally, developers can exploit other properties of Smart Services designing their composition services to change with time, better fitting to specific requests or user groups (by using supporting engine services like social networks analysis, clustering, personalization, etc.).

4. SMART SERVICE

The idea of Smart Services extends the concept of a composite service. A Smart Service is in fact a type of a composite service, however, its elements don’t necessarily have to be atomic services. A Smart Service is represented by a combination of interconnected nodes, some of which can represent concrete services, some sets of services and some various types of requirements. Many different classes of nodes can be defined in Service Composer as long as Workflow Engine is configured to interpret and “execute” them. In this regard a concept of Smart Services fully supports the Service Composer tool as a framework for service composition.

The SSDL language (Smart Service Description Language), used to describe Smart Services, together with the Workflow Engine, allows for complete definition of a composition scenario for a particular requirement of a composite service. The composition can also be delayed by *executing* the requirement nodes by the Workflow Engine (instead of composing first and executing services later), which will lead to dynamic composition at runtime. This will be explained in more detail in section 5.

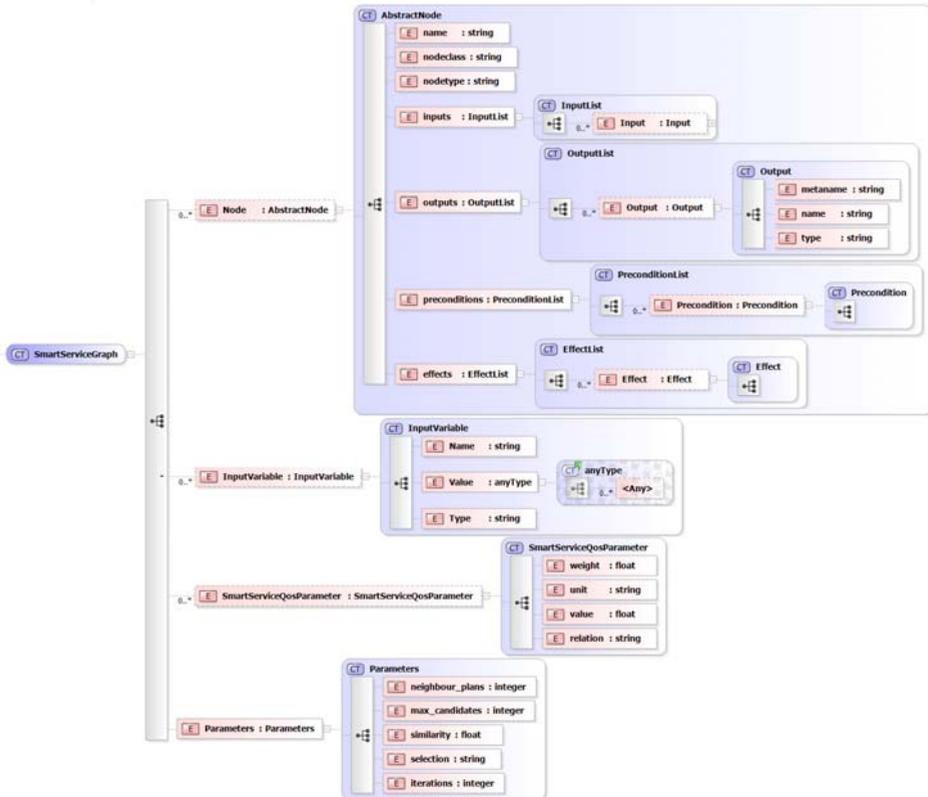


Fig. 3. SSDL node types defined in XSD

The basic class describing a node of SSDL defined composite service is an AbstractNode class:

```
<xs:complexType name="AbstractNode">
  <xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="nodeclass" type="xs:string" />
    <xs:element name="nodetype" type="xs:string" />
    <xs:element name="inputs" type="InputList" />
```

```

<xs:element name="outputs" type="OutputList" />
<xs:element name="preconditions" type="PreconditionList" />
<xs:element name="effects" type="EffectList" />
</xs:sequence>
</xs:complexType>

```

The above example shows that the abstract node already contains much information about required input and output parameters of services and language constructs defining preconditions and effects of a service. Other types of nodes inherit after AbstractNode class – they can describe different kinds of requirements, services, broker agent requests etc. In fact the list is open and limited only by the ability of the Workflow Engine to recognize those types of nodes and execute them appropriately. However, some basic types of nodes were predefined and user defining a request in SSDL can be certain that the Workflow Engine will be able to interpret node classes described in following subsections.

4.1. FUNCTIONALITIES

```

<xs:complexType name="FunctionalityNode">
  <xs:complexContent>
    <xs:extension base="AbstractNode" />
  </xs:complexContent>
</xs:complexType>

```

Currently functionalities (FunctionalityNode) are not much different than the basic AbstractNode, however, for the Workflow Engine this is sufficient information that this kind of node should be executed differently than a basic abstract node.

Typically execution of a functionality node means that some kind of selection service should be run and then its result should be executed. Also functionality nodes connected together make up a basic service composition request and can be an input to a variety of end-to-end composition services.

Those two types of approaches to functionality node should not be confused, because executing a FunctionalityNode is in fact a greedy approach to composition (yet it can have its own benefits – like composition at runtime or using a broker service) and typical end-to-end composition can apply completely different methods, before sending a composite service to the Workflow Engine.

4.2. SERVICES

```

<xs:complexType name="ServiceNode">
  <xs:complexContent>

```

```

<xs:extension base="AbstractNode">
  <xs:sequence>
    <xs:element name="address" type="xs:string" />
    <xs:element name="method" type="xs:string" />
  </xs:sequence>
</xs:extension>
</xs:complexType>

```

Executing a service can mean various things – from typical execution of a remote web service to executing a broker agent (as a web service) that, given an url of a desired web service, will try to find and execute an instance of that service that is at the moment best fit for our system (least used, in vicinity in the network, etc.).

The service node can also preserve the abstract description of user requirements, and, after error occurrence, an error handling procedure can be executed and a selection service can find another candidate to be executed in the originals place.

4.3. WORKFLOW ENGINE INSTRUCTIONS

```

<xs:complexType name="ControlNode">
  <xs:complexContent>
    <xs:extension base="AbstractNode">
      <xs:sequence>
        <xs:element name="controlype" type="xs:string" />
        <xs:element name="condition" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

This type of node contains instruction in engines domain specific language (DSL): conditional statements, loops and more complex scenarios or even calculations. Those instructions are interpret live and using metaprogramming techniques can be added to Workflow Engines code.

4.4. SMART SERVICE SUBGRAPH

```

<xs:complexType name="SmartServiceGraph">
  <xs:sequence>
    <xs:element name="Node" type="AbstractNode" minOccurs="0"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="InputVariable" type="InputVariable" minOccurs="0"
maxOccurs="unbounded" />
<xs:element name="SmartServiceQosParameter" type="SmartServiceQos
Parameter" minOccurs="0" maxOccurs="unbounded" />
<xs:element name="Parameters" type="Parameters" />
</xs:sequence>
</xs:complexType>

```

This type of node allows for isolation of part of a Smart Service and delegating its execution to a remote workflow engine. In this situation a service executed is in fact a workflow engine service that accepts a Smart Service subgraph.

5. WORKFLOW ENGINE

The main feature distinguishing Service Composer's workflow engine from other engines is its focus on composition and, together with the expressive nature of SSDL language it interprets, ability to configure its behavior.

The Workflow Engine can be configured to interpret and execute different kinds of node classes defined in SSDL. By execution we understand performing a series of actions on the nodes content, to finally obtain a web service to be executed in its place. Those actions can be defined by Smart Services, which are assigned to specific node classes. This again shows an elastic nature of SSDL language and Service Composer framework itself, because services defined in the Service Composer engine can serve for composition purpose while defining a composite service with the *graphical user interface* and perform composition (as well as other execution scenarios) *at runtime* during execution of a requested composite service.

Fig. 4 shows that the Workflow Engine works in two phases. First, in the *init* phase (Fig. 4.a) it performs validation of the SSDL service execution request and can run services transforming the whole SSDL: like broker agent request (which can indicate all service instances at once) or even end-to-end composition. Then, in the *process* phase (Fig. 4.b) the engine executes each of the nodes of the SSDL. Both in the *init* and the *process* phase all actions, if not implemented in the Workflow Engine, can be performed by appropriate web services indicated by the Workflow Engine configuration (Fig. 4.c).

The Workflow Engines' architecture allows for further extension of its capabilities. Whether a central repository of execution scenarios would be maintained, a local workflow engine could try and execute classes of nodes it does not know at the moment. Of course this would need more control over the process and in this section is merely signaled as a potential capability to interpret various types of user requirements and a method for extending Service Composer capabilities with its use by various researchers.

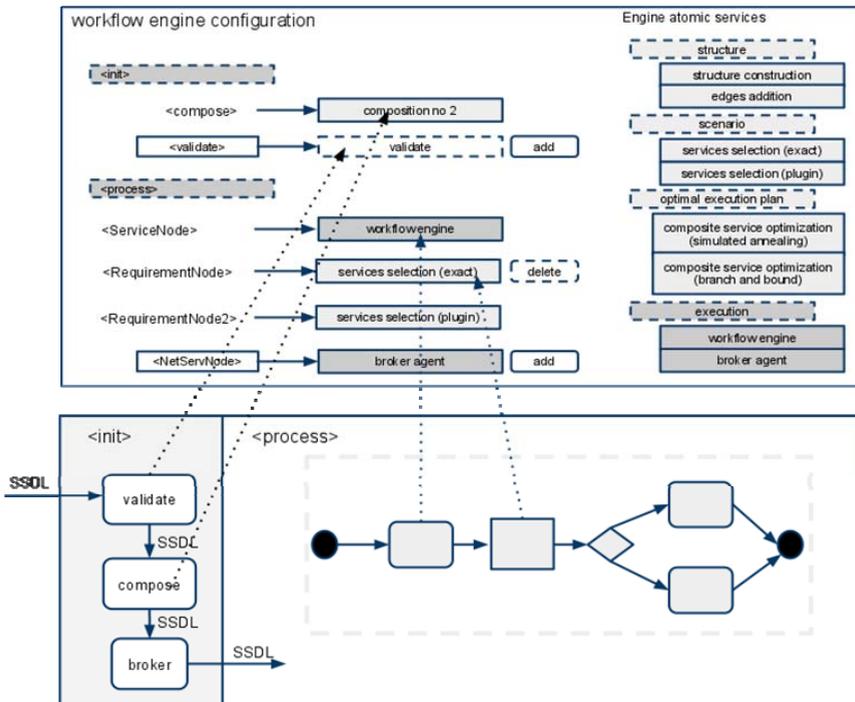


Fig. 4. SSDL executed by the Service Composers' workflow engine

6. SERVICE COMPOSER IMPLEMENTATION

The Service Composer was implemented in the Ruby on Rails web application framework, using mostly Ruby language and jQuery javascript framework for the front-end programming. The approach used to develop the Service Composer utilizes the Model View Controller programming pattern.

The main components of the framework are:

1. *GUI controllers* – controllers responsible for the Service Composer web portal (the front end that enables the business user to create a profile and register his repositories and ontologies as well as define specialized composition services; the same controller allows developers to add second layer services to aid the composition process)
2. *Workflow Engine controller* – responsible for running the local instance of the Workflow Engine executing the SSDL defined user and composition Smart Services (it can also act as a remote workflow engine for others)

3. *Mediator controllers* – responsible for connecting to remote service repositories and ontologies
4. *Service repositories and ontologies models* – responsible for maintaining example service repositories and ontologies
5. *Composer application* – a main composition application delivering methods for service composition to the Service Composer engine by the means of web services
6. *Edges application* – an example of an external composition application that delivers its functionalities as web services supporting the structure stage of service composition
7. *Security estimates application* – an example application that delivers specialized QoS estimates by the means of web services enabled security monitoring multiagent system

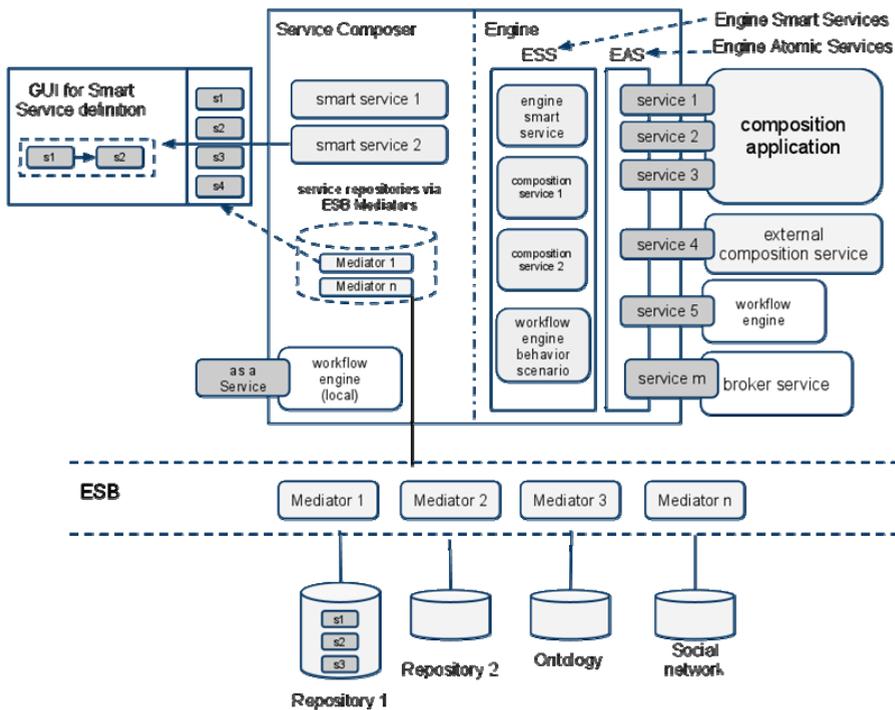


Fig. 5. Detailed Service Composer Architecture

Fig. 5 illustrates how various parts of Service Composer are interconnected. External services and knowledge repositories are accessible via mediators, which are organized together forming a service bus. The Service Composer portal aggregates data from repositories and makes them available to a range of applications. Specialized graphical

user interface can aid in the process of composition by hand and engine composition services can utilize the data in their activities. Developers may submit their solutions via web service interface and register them in the Service Composer engine. Many end-to-end and specialized composition services may exist in parallel and they can be supported by other web services and bound together in a form of Smart Services. This approach allows designers to test various composition methods and via the Workflow Engine configuration define situations in which they would be most appropriate.

7. CONCLUSIONS

In this work the architecture for a service composition framework was proposed. The framework consists of methods aiding the construction and deployment of end-to-end composition service. Additionally we have introduced the concept of Smart Services which, using the SSDL language, describe its functionalities and, along with the Workflow Engine included in the Service Composer framework, self interpret at execution.

To our best knowledge it is the first framework for service composition that implements this kind of elastic approach. Some composition frameworks described in literature were in fact complex composition methods with well-specified components, not composition frameworks to *aid* in composition applications *construction*.

Proposed architecture and mechanisms allow designers to incorporate into the Service Composer a range of composition approaches, personalize them for specific domains and deliver them as web services. In future work we intend to utilize this important aspect of Service Composer by developing various methods showing the elastic nature of the framework and encouraging researchers to include composition methods of their own.

ACKNOWLEDGEMENTS

The research presented in this work has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

REFERENCES

- [1] AGARWAL V., CHAFLE G., DASGUPTA K., KARNIK N., KUMAR A., MITTAL S., SRIVASTAVA B., *Synthy: A system for end to end composition of web services*, Web Semantics: Science, Services and Agents on the World Wide Web In World Wide Web Conference 2005, Semantic Web Track, Vol. 3, No. 4., 2005, 311–339

- [2] AGGARWAL R., VERMA K., MILLER J., MILNOR W., *Constraint Driven Web Service Composition in METEOR-S*, Proceedings of the 2004 IEEE International Conference on Services Computing, 2004, 23–30
- [3] PONNEKANTI S. R. AND FOX A.: *SWORD: A developer toolkit for Web service composition*. In Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA (2002)
- [4] KLUSCH M., FRIES B., SYCARA K., *OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services*, Web Semantics: Science, Services and Agents on the World Wide Web 7, 2009, 121–133
- [5] JONG M. K., CHANG O. K., ICK-HYUN K., *Quality-of-service oriented web service composition algorithm and planning architecture*, The Journal of Systems and Software 81, 2008, 2079–2090
- [6] JINGHAI R., XIAOMENG S., *A Survey of Automated Web Service Composition Methods*, Semantic Web Services and Web Process Composition, First International Workshop, SWSWPC 2004, San Diego, CA, USA, 43–54.
- [7] MILANOVIC N., MALEK M., *Current Solutions for Web Service Composition*, IEEE Internet Computing 8(6), 2004, 51–59.

Jan WEREWKA*, Grzegorz ROGUS *

A SOLUTION FOR ADAPTATION OF LEGACY ENTERPRISE SOFTWARE FOR PRIVATE CLOUD COMPUTING MODEL

The study examines an issue, in which two or more companies have complementary software used by customers in a given industry domain. The results of the business analysis have shown a necessity of adapting current software to the cloud computing implementation model to provide potential customers with a wide range of services. Adaptation of software requires execution of several steps. One of them is standardization of software as reusable components according to the principles of SOA.

To promote a rapid implementation of the objectives, the assumed activities must be unified in a well-defined project. The work proposes a project definition, which describes a solution for development and adaptation of legacy enterprise software in the private cloud computing model.

Undertaking software adaptation is discussed from the viewpoint of senior management of the IT enterprise which intends to start the transition project. That is important because decisions connected with organizational changes and changes in the software development process should be taken.

The solutions presented in the work are based on well-known standards and patterns and therefore they can be easily adapted for other similar software transition projects.

1. INTRODUCTION

IT enterprises are facing a challenge of putting their software to the cloud and simultaneously performing software modernization towards the service oriented architecture. This direction of software development is possible and necessary due to a dramatic improvement in telecommunication regarding transmission throughput, speed, reliability and heavy transmission costs reductions.

Cloud computing is a new deployment technology of software systems. The reasons driving cloud technology are in general [10]: reduction of IT operating costs; the

* AGH University of Science and Technology, ul. Mickiewicza 30, 30-059 Kraków, Poland.

ability to obtain new resources quickly; simplification of IT infrastructure and management; increase of overall IT flexibility and agility; the ability to replace existing solutions; the ability to scale up and down quickly; improvement of service availability and uptime; lack of necessity to spend funds to expand; energy efficiency, lower carbon footprint; reduction of IT staff; reduction of IT risk.

For IT enterprises transferring software to the cloud have a slightly other meaning. For the IT companies developing software it means replacing capital expenses (CAPEX) with operational expenses (OPEX). For companies this means that they will go towards operational activities. As a result the IT Company's long-term expenses will be lower and the number of activities that require in house IT expertise will decrease.

The IT enterprises are confronted with a necessity to deliver valuable software to their customers in the shortest release cycles. For shortening the release cycles MMF (Minimal Marketable Features) approach reducing product scope may be proposed. The development should facilitate software modernization towards reusable components. The software modernization process should be based on proven standards. Relying on SOA (Software Oriented Architecture) paradigm and using agile development which gives ability to respond to changes and new requirements is a good solution for most enterprises. In order to utilize SOA effectively the lean approach ought to be used which is not limited only for software development. That is why three viewpoints are considered in this work: technological (software and hardware solutions), economical (which solution will be the best from economical point of view, using e.g. SWOT analysis), managerial (how to succeed with project classical constraints costs, time and scope).

2. SCOPE OF THE CLOUD COMPUTING PROJECT

For the cloud computing project high level scope should be determinate, basing on known reference characteristics. NIST (National Institute of Standards and Technology) defined one of the most popular and recognized definitions of the cloud computing which will be used in this work. The definition states that [9] "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". The model defines five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service), three cloud service models (Software as a Service (SaaS), Platform as Service (PaaS) and Infrastructure as a service (IaaS)), and four deployment models (private, community, public, and hybrid clouds).

The first step will be connected with selection of cloud computing model which should be deployed. One of the simplest categorization of cloud computing models is Cloud Cube Model proposed by the Jericho Forum [11]. The model defines four dimensions:

- Internal/External (private/public cloud), determines if the cloud that will be used exists inside or outside organization's boundaries.
- Proprietary/Open, defines the type of ownership of the cloud technology. Proprietary means that the organization providing the service keeps the means of provision under their ownership. Clouds that are open are using open technology for more suppliers.
- Perimetrised/ Deperimetrised. A perimeter is a path that surrounds an area. Perimeterised implies operating within the traditional IT perimeter like "network firewalls". In a deperimeterised frame data would be encapsulated with meta-data and mechanisms that would protect data from inappropriate usage.
- Insourced and Outsourced, means that the service is provided by a 3rd party (outsourced) or by own staff (insourced).

Basing on the above definitions, it is assumed that the system taken into consideration in this work is internal, proprietary, insourced in the first stage of development and perimetrised and deperimetrised in the second stage. For such a project with a private cloud destination it can be presumed that organization owns infrastructure and is responsible for the infrastructure management (facilities, computer, network, storage devices, etc.). In the next phase the infrastructure ownership and management can be shifted to the third party provider. The cloud can be as on-premise and inserted in trusted environment. Trusted environments mean that the consumers of services can be considered as a part of the organization. In the second deployment stage the infrastructure may be put in off-premise in an untrusted environment in which the consumers may be authorized to use services but they are treated as not a part of an organization.

The results of the business analysis have shown a need of adapting current software to the cloud computing implementation model to provide potential customers with a wide range of services. Adaptation of software requires execution of several steps. One of the steps is standardization of software components according to the principles of SOA. It is assumed that software deployment to the cloud should be tied with software modernization towards SOA architecture.

To avoid opacity OASIS definition of SOA will be selected as reference. OASIS (Organization for the Advancement of Structural Information Standards) defines SOA as [4, 13] "A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provided a uniform means to offer, discover, interact with and use capabilities to produce desired effects consisting with measurable preconditions and expectations".

It is evident that the concepts of SOA and cloud computing have synergies which should be exploited: SOA is used for dividing processes into IT services to gain flexibility and reuse, cloud computing gets rid of the maintenance and infrastructure.

The software transfer to SOA should be based on standards for SOA governance and SOA lifecycle management. The SOA Governance Reference Model (SGRM) [3] from the Open Group is a generic model that is used as a baseline for SOA Governance Model to expedite the process of tailoring SOA Governance Model for an organization. The SOA Governance Vitality Method (SGVM) is [3] a process that utilizes the SOA Governance Reference Model (SGRM) as a baseline and then follows a number of phased activities to customize this baseline model to cater for the organization's variants.

In the considered project a SOA solution should be chosen considering importance of: software agility, lean approach to software development and short development cycles basing on MMF principles.

3. ORGANIZATION COMPETENCY ASSESSMENT FOR SOA TRANSFORMATION

Building efficient and flexible software based on SOA is possible when the enterprise environmental factors are at a required level. The competency of the organization regarding SOA development can be measured by using maturity models. The most popular maturity model CMMI (Capability Maturity Model for Integration) [7] proposed by Software Engineering Institute from Project Management Institute may be used. In case of a software company which modernizes its products in the SOA direction, a good way to determine the goals is to investigate special models connected with the mentioned architecture. Each large software organization has its own models, like Infosys SOA Maturity Model, Microsoft Maturity Model (SOAMM), IBM SOA Maturity Model (SIMM), Sonic and partners SOA Maturity Model, The Open Source SOA Roadmap, OSIMM model from the Open Group. In this work OSIMM will be used as a reference maturity model.

The Open Group Service Integration Maturity Model [2] provides means to assess a corporation's service maturity. The model is quantitative and it is presented as a two-dimension matrix. In Fig. 1 dimension icons were added to ensure better illustration. The columns are characterized by a level of maturity, and the rows concern so called dimensions which are the areas of effective service adoption. The OSIMM model defines 7 levels (Table 1) characterized mainly by de-coupling and flexibility [2].

The OSIMM model applied to the considered project shows that it is not enough to consider changes in software development but an influence of changes in business organization, communication and infrastructure must be also taken into account.

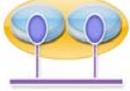
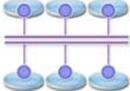
The OSIMM model applied to the considered project, gives information that it is not enough to consider changes in software development, but influences of changes in business, organization, communication and infrastructure must be also taken into account.

	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7
	 Silo	 Integrated	 Componentized	 Services	 Composite Services	 Virtualized Services	 Dynamically Re-configurable Services
 BUSINESS	Isolated Business Line-driven	Business Process Integration	Componentized Business	Componentized Business Offers Services	Processes through Service Composition	Geographical-independent Service Centers	Mix-and-match Business and Context-aware Capabilities
 ORGANIZATION	Ad hoc LOB IT Strategy & Governance	Ad hoc Enterprise IT Strategy & Governance	Common Governance Processes	Emerging SOA Governance	SOA and IT Governance Alignment	SOA and IT Infrastructure Governance Alignment	Governance through Policy
 METHODS	Structured Analysis & Design	Object-oriented Modeling	Component-based Development	Service-oriented Modeling	Service-oriented Modeling	Service-oriented Modeling for Infra (CDSP)	Business Grammar-oriented Modeling
 APPLICATIONS	Modules	Objects	Components	Services	Proces Integration via Services	Proces Integration via Services	Dynamic Assembly: Context-aware Invocation
 ARCHITECTURE	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid-enabled SOA	Dynamic Re-configurable Architecture
 INFORMATION	Application-specific	LOB or Enterprise-specific	Canonical Models	Information as a Service	Enterprise Business Data Dictionary and Repository	Virtualized Data Services	Semantic Data Vocabularies
 INFRASTRUCTURE	LOB Platform-specific	Enterprise Standards	Common Re-usable Infrastructure	Project-based SOA Enviroment	Common SOA Environment	Virtual SOA Enviroment; S&R	Dynamic Sense, Decide & Respond

Fig. 1. OSIMM maturity matrix

The first three levels starting from silo to componentized level are defined as Service Foundation Levels. The foundation levels are used for enabling legacy environment to determine efforts to modernize the application architecture and to provide a consistent approach to legacy code reuse.

Table 1. OSIMM model levels

Level	Icon	Description
1.		Silo (data integration). Individual parts of the organization are developing their own software independently, with no integration of data, processes, standards, or technologies.
2.		Integrated (application integration). Technologies are used to communicate between the silos and to integrate data and interconnections. Connecting two systems requires a possibly complex conversion of data, operations and protocols.
3.		Componentized (functional integration). The IT systems in the silos are broken down into component parts. Although components interact through defined inter-faces, they are not loosely coupled, which limits agility and interoperability. This makes it hard to develop cross-organization business processes.
4.		Simple services (process integration). Composite applications are built from loosely-coupled business services. The way in which services may be invoked is based upon open standards and is independent of the underlying application technology. However, at this stage the composition of services is still performed by developers writing bespoke code.
5.		Composite services (supply-chain integration). It is now possible to construct a business process for a set of interacting services, not just by bespoke development, but by the use of a composition language to define the flow of information and control through the individual services.
6.		Virtualized services (virtual infrastructure). The business and IT services are now provided through a façade, a level of indirection. The service consumer does not invoke the service directly, but through a virtual service.
7.		Dynamically reconfigurable services (eco-system integration). Prior to this level, the business process assembly, although agile, is performed at design time by developers (under the guidance of business analysis and product managers) using suitable tooling. Now this assembly may be performed at run time, either assisted by the business analysts using suitable tooling, or by the system itself.

Seven areas (Table. 2) of service adoption described as dimensions [2] are defined in the maturity matrix.

Table 2. OSIMM model dimensions

Level	Icon	Description
1.		Business: concerns organization business practices and policies. The area considers cost and flexibility of IT capabilities, business agility and service-level agreements.
2.		Organization & Governance: concerns a structure and design of the organization, measures of organizational effectiveness in the SOA context
3.		Methods: concerns methods and processes employed by the organization for its IT and business transformation, maturity of the software development life-cycle, software engineering practices and guidelines used for the design and development of SOA.
4.		Applications: concerns an application style, structuring of the application and functionality of the underlying applications, including the attributes of reusability, flexibility, reliability and extensibility of the applications.
5.		Architecture: concerns a structure of the architecture style which includes a use of reference architectures, logical and physical topologies, integration techniques and patterns, enterprise architecture decisions, standards and policies, the web services adoption level, experience in SOA implementation, SOA compliance criteria, and typical artifacts produced.
6.		Information: concerns the way information is structured, the method of access to enterprise data, abstraction of data access from the functional aspects, data characteristics, data transformation, service and process definition, handling of identifiers, security credentials, knowledge management, the business information model and content management including the mechanisms for integration of information across the enterprise.
7.		Infrastructure: concerns infrastructure capabilities including service management, IT operations, IT management and IT administration, SLA compliance, monitoring and types of integration platforms provided.

The model should be used for determining architectural strategy by SOA adaptation, software product roadmap regarding legacy transformation and project management plan considering software modernization. The initial steps that a project schedule should include: Assessment of the current maturity levels of the business, organization and IT. For that a set of questions is provided by OSIMM standard to help understand the organization position in each dimension; Determination of the required maturity levels needed to reach defined business goals; Determination of the gaps in maturity levels for each dimension.

Assessment of maturity model describes information needed to determine maturity attributes. Information required should be obtained by reviewing different stakeholders, e. g.: software development and a deployment group; an IT operation group, managers and business staff supporting a service or a business area.

The base OSIMM model can be extended by adding additional maturity indicators, assessment questions and corresponding attribute mappings to encompass maturity indicators specific to industry or enterprise.

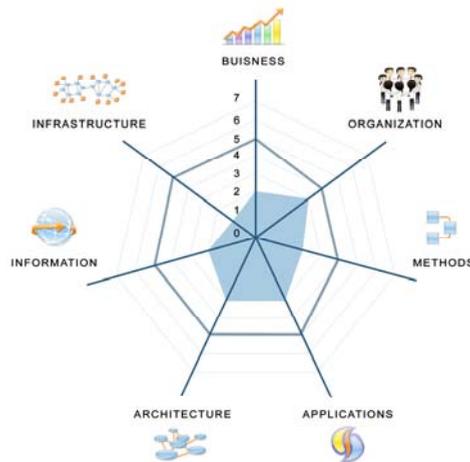


Fig. 2. Gap determination between target and current maturity

The gap analysis between target and current maturity for each dimension in OSIMM matrix, as presented on Fig. 2, will be used to determine the transformation strategy. For the considered IT project the gap analysis between current and target levels (cl, tl) may look as follow: Business (2, 5), Organization (3, 4), Methods (2, 4), Application (3, 5), Architecture (3, 5), Information (2, 5), Infrastructure (1, 5).

Estimating the cost and effort needed for SOA adaptation is not an easy task. Moreover, the referenced framework may be proposed for the scope, cost and effort estimation for SOA projects (e. g. [15]). Transformation to SOA gives advantages connected with cost efficiency, agility, adaptability and legacy leverage so the best solution according to Deming cycle is to apply a pilot project to obtain a better understanding of risks and to be well prepared for full transformation.

4. APPLICATION OF TOGAF IN SOFTWARE TRANSITION TO CLOUD COMPUTING MODEL

Released software components are the most important deliverables of the transition project. To identify software project deliverables in a systematic way, a suitable architecture framework should be applied. This framework will be also used for defining milestones or iterations in software development process.

There are different frameworks that can be used for enterprise architecture development, e.g.: Zachman Framework, FEAF (Federal Enterprise Architecture Framework), DoDAF (Department of Defense Architecture Framework), TOGAF (Open Group Ar-

chitecture Framework), Meta Group (currently Gartner framework). In the work TOGAF is used as reference architecture framework.

TOGAF (The Open Group Architecture Framework) can be considered as one of the most popular architectural framework which is used to create enterprise architecture. TOGAF defines four main components [5]:

- 1) High-level general framework, defining key concepts which can be described by content metamodel.
- 2) Method of creating architecture ADM (Architecture Development Method) which is mainly described by the architecture development cycle.
- 3) Reference architecture (TOGAF Foundation Architecture) containing: technical reference model, knowledge base of the Open Group standards and information base on architectural blocks.
- 4) Resource base (TOGAF Resource Base) which is a set of tools and techniques that can be used when working with the TOGAF ADM.

The discussion concerning transition of the software to the cloud will be concentrated on a content metamodel and architecture development cycle.

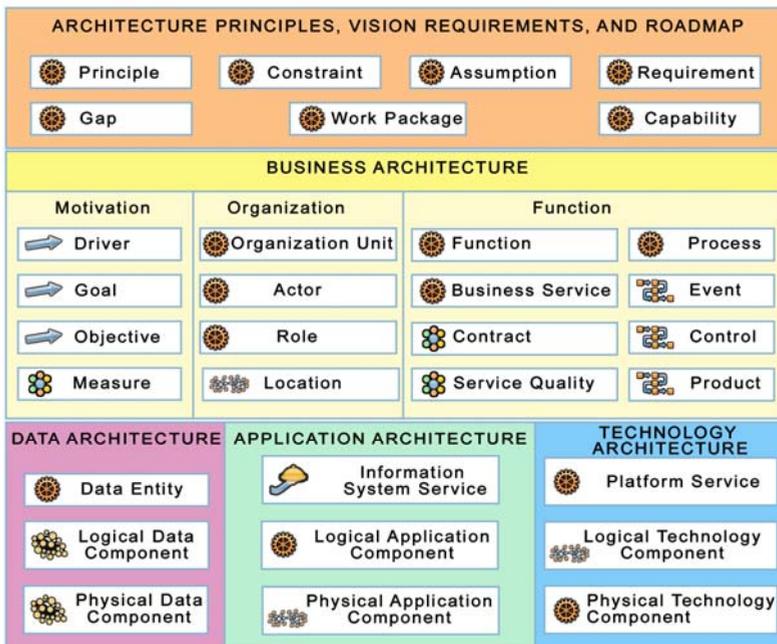


Fig. 3. TOGAF Content Metamodel

The content metamodel presented in Fig. 3 aids to define artifacts produced during the software development process. TOGAF introduces a concept of extension, which may be used to describe additional meta-model entities which should be considered

when developing SOA [16] or Cloud [17] architectures. The content metamodel defines a set of extensions which are presented in Table 3. It is vital for the transition project to find out a relation between the cloud services and the elements of TOGAF architecture. Such a basic mapping should be determined. It will help to identify artifacts of the project.

Table 3. Extension of the content metamodel

Icon	Extension type
	Motivation Extension
	Infrastructure Consolidation Extension
	Governance Extension
	Process Modeling Extension
	Data Modeling Extension
	Services Extension
	Core Content

The TOGAF ADM shows architecture development as phases that are included in an architecture iteration cycle (Fig. 5). Each phase can be treated as a process with defined objectives, inputs, activities (ordered in steps) and outputs.

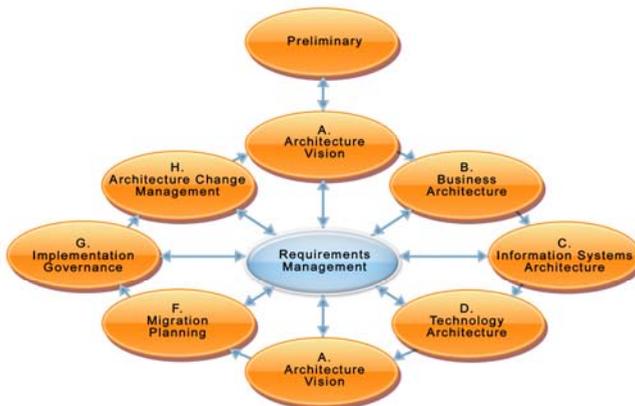


Fig. 4. TOGAF architecture development cycle

Table 3 presents ADM phases with the proposal of additional activities required to transition to SOA [16] and cloud computing [17] architecture.

Table 4. TOGAF phases mapped to SOA and Cloud Computing extensions

TOGAF Phase	SOA extensions	Cloud Computing extension
Preliminary Defining “where, what, why, who, and how” Enterprise Architecture will be done.	Identify SOA specific stakeholder concerns. Define Scope specific to SOA, ensure that a scope is appropriate for SOA, and tailor deliverables to level of architecture. Evaluate business capabilities – SOA readiness. Confirm SOA supporting principles.	Creation of a strategy for the consumption and management of cloud services. Strategy contains: Cloud categories definitions, Transactions and list of activities addressed to: IT Service Management, Risk Management, License and Asset management, Vendors Management, Quality Management, Security Management. Identification owners, processes, roles and responsibilities related to Cloud services and operations
A. Architecture Vision Defining scope, identifying stakeholders, creating Architecture Vision and obtaining approvals	Identify SOA specific stakeholder concerns. Define Scope specific to SOA, tailor deliverables to level of architecture. Evaluate business capabilities – SOA readiness. Confirm SOA supporting principles.	“no additional activities”
B. Business Architecture Aligns the enterprise's business processes, people, operations, and projects with its overall strategy	Select Reference models, viewpoints and tools: SOA metamodel & content extension, Information Entity & Information component.	Consider a Cloud Reference Model from: The Open Cloud Consortium, The Cloud Security Alliance, The Cloud Computing Reference Model (CC-RM) and Reference Architecture framework from Agile Path, The Accenture Cloud Reference Model for Application Architecture.
C. Information Systems Architectures	Select Reference models, viewpoints & tools: SOA metamodel & content extension, IS Service Contract, Relation between IS Service & Data Entity.	Determine data and privacy classification
D. Technology Architecture Map application components into a set of technology components	Select Reference models, viewpoints & tools: SOI reference model, Relationship between Logical Technology & Logical Application Component.	“no additional activities”
E. Opportunities & Solutions Migration Planning	Select Reference models, viewpoints & tools: Physical Data Physical Application, Technology Application Component and SOA Solution.	Identify candidates services in the Cloud

F. Migration Planning Implementation and Migration Plan	“no additional activities”	Provide operational expenditure outlines.
G. Implementation Governance Provides an architectural oversight of the implementation	“no additional activities”	Relocation of business processes, applications, data, technical services. Implement security
H. Architecture Change Management Ensure that the architecture achieves its original target business value.	Define change management policies on a two-dimensional perspective: as a specific version of a service or other services progresses through its System Development Life Cycle.	“no additional activities”
Requirements Management	“no additional activities”	“no additional activities”

The rough comparison of contents of TOGAF phases between standard and extension towards SOA and cloud architecture shows significant similarities and differences. The transition project should be carefully defined and requirements must be evaluated during the decision process which is in or out of the project scope.

5. TRANSFERRING OF LEGACY SOFTWARE TO THE CLOUD AS THE IT PROJECT

Transferring the legacy software based on services to the cloud is complex and high risk undertaking especially when IT enterprise has competency deficiencies and is unprepared for this venture. That is why; the undertaking should be described in the same way as a well-defined project.

The fundamental question is how to define a success in this case. Classical definition of project success states that a project is successful if it was finished within the boundaries of time and costs and all customers’ requirements were met. A better definition should be based on project strategy which will depend on product: superiority, cost advantage, time-to-market or customer intimacy.

The definition of the project success should consider cloud business model and sustainability. Sustainability means an ability of adaptation to external and internal changes. Sustainability here may concern software which is the main deliverable of the project and IT enterprise organization as well. The hexagon model is used to describe strengths and weaknesses of cloud business models. The model consists of six elements divided into pairs [1]: people (consumers and investors), business (popularity and valuation), and job done with job variance (GTJD – get the job done, innova-

tion). The business may be more oriented to consumers or investors, based on popularity or on economic value of assets, towards delivering new features (innovation) or concentrated on getting the client's problems resolved.

One of the most important decisions in the software transferring is to decide which parts of software should be made by IT Company and which part of software should be bought. The easiest division is doing make software by IT Company which is domain specific; the rest of the software should be acquired. In the decision taking ROI and TCO parameters should be investigated.

In the first stage software components should be identified (whether they can be acquired and easily integrated with the rest of software). One of the most crucial parts of SOA architecture is ESB (Enterprise Service Bus). In general ESB provides functionality in five areas: architecture, connection, mediation, orchestration, change and control. The ESB product capabilities are used for meeting general integration needs and supporting efforts to implement a service-oriented architecture. Choosing ESB functionality for your project should be carefully considered. Forrester Research defined evaluation criteria [12], five popular commercial and four open source ESB product vendors: Fuse Source, IBM, Mule Soft, Oracle, Progress Software, Red Hat, Software AG, Tibco Software, and WSO2. There are many offers for cloud software and platforms. Selection of the ESB software that will be acquired is a complex decision. There are some proposals which can help with buying or making decisions e.g. [14].

After clarification of the project success and evaluation which part of the software should be acquired, the high level project schedule can be determinate. To promote a rapid implementation of the objectives, the activities should be unified in one project and it may consist of the following phases: The assessment of maturity of existing software and the organizations producing software based on the OSIMM model; Development of high-level business process model for offered cloud computing software on the basis of TOGAF; The analysis of the possibility of building or buying parts of software, particularly software needed for ESB and general cloud computing software; Identification of project constraints and assumptions by developing TCO and ROI models; Implementation of pilot solutions for selected services; The Decision which software will be rewritten and which will be wrapped up; Implementation and deployment of the private cloud computing for the defined business model.

The transition to the cloud computing model will shift IT enterprise activities towards operation. This is generally a positive shift but the organization strategy should be oriented to project and operation planning and management. In the case the project should include software deployment and maintenance, and then ITIL (Information Technology Infrastructure Library) may be a good reference point for operations focusing service management.

6. CONCLUSIONS

Cloud computing together with building reusable software components is the fastest growing part of IT industry, offering benefits for both customers and IT enterprises.

Due the short development cycles it is necessary to integrate parts of legacy software into new systems. Transforming legacy software to the cloud may be a high risk task; therefore the transformation should be managed like a well-defined project. The senior management of IT enterprise responsible for the transformation project should take into account technological, economical and managerial issues. The work states that before starting any SOA or cloud computing software development, it is necessary to undertake high level decision making and planning proposed in the work. The described proposals may be used as a starting point for beginning of similar projects.

Important part of the project is definition of business processes and services. The authors developed models for business processes for project management activities in project oriented IT enterprises. The project managers or PMO can integrate business processes and services originating from different project methodologies (e.g. classical and agile [8]). The developed model will be used to define service oriented software architecture which should be deployed as B2B private or community cloud.

REFERENCES

- [1] CHANG V., WILLS G., DE ROURE D., *A Review of Cloud Business Models and Sustainability*, 2010 IEE 3rd Int. Conf. on Cloud Computing, 2010, IEE Computer Society, pp. 43–50.
- [2] *The Open Group Service Integration Maturity Model (OSIMM), Technical Standard*, The Open Group, 2009, p. 73, <http://www.opengroup.org/bookstore/catalog/c092.htm>
- [3] *SOA Governance Framework, Technical Standard*, The Open Group, August 2009 (C093), www.opengroup.org/bookstore/catalog/c093.htm
- [4] *OASIS Reference Model for SOA (SOA RM)*, Version 1.0, OASIS Standard, 12 October 2006; docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf
- [5] *The Open Group Architecture Framework (TOGAF)*; www.opengroup.org/architecture/togaf9
- [6] *Organizational Project Management Maturity Model (OPM3®) Knowledge Foundation*, 2nd Edition, Project Management Institute, 2008, p. 180.
- [7] *Capability Maturity Model Integration (CMMI) Version 1.1, CMMI for System Engineering, Software Eng., Integrated Product and Process Development and Supplier Sourcing*. Carnegie Mellon SEI, CMMI Product Team, 2002, p. 724.
- [8] WEREWKA J., SZWED P., ROGUS G., *Integration of classical and agile project management methodologies based on ontological model*, Production engineering in making, ed. Piotr Lebkowski. Krakow: AGH University of Science and Technology Press, 2010, pp. 7–28.
- [9] MELL P., GRANCE T., *The NIST Definition of Cloud Computing (Draft), Recommendations of the National Institute of Standards and Technology*, The National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Special Publication 800–145, 2011, p. 6., pp. 7–28.

- [10] *The Cloud Computing Research Study*. An 1105 Government Information Group Research, www.FCW.com/DownloadingCloudComputing, 2011, p. 15.
- [11] *Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration*, www.jerichoforum.org, Version 1.0, 2009, p. 7.
- [12] VOLLMER K., GILPIN M., ROSE S., *The Forrester Wave™: Enterprise Service Bus*, Q2 2011, 2011, www.forrester.com, p.15.
- [13] *Reference Architecture Foundation for Service Oriented Architecture*, Version 1.0, Committee Draft 02, OASIS, <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf>, 2009, p. 119.
- [14] *Cloud Buyers' Decision Tree, A Proposal for Discussion*, The Open Group, 2010, p. 37.
- [15] O'BRIEN L., *A Framework for Scope, Cost and Effort Estimation for Service Oriented Architecture (SOA) Projects*, Australian Software Engineering Conference 2009.
- [16] *Using TOGAF to define and Govern Service-Oriented Architectures*, *Open Group Guide*, The Open Group, Mai 2011, <http://www.opengroup.org>, p. 59.
- [17] THORN S., *Cloud Computing requires Enterprise Architecture and TOGAF® 9 can show the way*. www.architecting-the-nterprise.com/articles/cloud_computing_requires_enterprise_architecture_and_togaf_9_can_show_the_way.php

PART II

SOA-BASED APPLICATIONS
– SELECTED ISSUES

Sergiusz STRYKOWSKI*, Rafał WOJCIECHOWSKI*

ONTOLOGY-BASED MODELING FOR AUTOMATION OF ADMINISTRATIVE PROCEDURES

The key competence of public administration is execution of administrative procedures. In order to improve efficiency and quality in this area, public administration should automate this execution by replacing, as far as possible, the participation of human factor with IT systems. However, the automation of administrative procedures requires very detailed models containing all necessary activities which are determined to be performed due to legal circumstances occurring in a course of a specific procedure. Classic modeling methods based on the monolithic approach do not allow creating models of administrative procedures at the level of detail allowing the automation. Furthermore, the models are fixed in the design phase and cannot be adapted to legal circumstances occurring during execution. In this work, a new approach to modeling of administrative procedures is introduced focusing on their automation. In the proposed approach, models of administrative procedures are dynamically composed of elementary processes. The selection of the elementary processes is performed based on analysis of legal circumstances occurring during the runtime phase. The analysis of the legal circumstances is performed by an inference engine evaluating decision rules against facts. The facts are instances of classes defined in the ontology developed for administrative law.

1. INTRODUCTION

The concept of e-government refers to the use of information and communication technology in public administration for increasing the availability of public services offered to citizens and businesses, as well as improving the quality, efficiency and effectiveness of the processes in public agencies. This means that e-government should not be limited to providing public services through electronic access channels only, but primarily should rely on the implementation of new solutions based on the information technology that improve various aspects of public administration operations.

* Poznań University of Economics, al. Niepodległości 10, 61-875 Poznań, Poland.

From the perspective of citizens and businesses, the area of public administration operations which they deal with the most is the execution of administrative procedures. Therefore, improvements in this area are the most wanted points. One of the key improvements here is the automation of administrative procedures defined as replacing to the greatest possible extent human labor with work of information systems. Such automation provides the following benefits: shortening transaction times, reducing costs, higher quality of operations, independence of geographical distances and locations, independence of working hours, and opportunity of automatic reaction to signals initiating and controlling administrative procedures.

The automation of administrative procedures requires the use of their detailed models. For modeling of administrative procedures the techniques used in workflow management systems can be applied. In classical workflow management systems models are created in accordance with the monolithic approach. However, modeling of entire administrative procedures using the monolithic approach is a complex and overwhelming task due to the large number of potential execution courses taking into account all possible legal circumstances that can arise during its execution for a variety of different cases. The problem of modeling the administrative procedures at a high level of detail is the main factor restraining the proliferation of e-government solutions offering a high degree of the automation. Therefore it is necessary to develop new methods and approaches for modeling and automation of administrative procedures.

In this work, the OMAP (Ontology-based Modeling of Administrative Procedures) approach for advanced automation of administrative procedures is proposed. The OMAP approach assumes that there is no modeling of administrative procedures in a form of end-to-end monolithic models. Instead of that, the course of an administrative procedure is dynamically constructed by ongoing adaptation to the current legal circumstances of that procedure.

The remainder of this work is organized as follows. In Section 2, different approaches to process modeling and applications of ontologies in the legal domain are presented. In Section 3, an overview of the OMAP approach is presented. In Section 4, the application of the OMAP approach for automation of the issue of determining the individual's legal capacity is presented. Finally, Section 5 concludes the work.

2. MODELING AND EXECUTION OF ADMINISTRATIVE PROCEDURES

One of the most common purposes of administrative procedure modeling is work standardization. Public agencies hire groups of analysts to develop the optimal models of administrative procedures. Then public agencies seek to ensure that their employees (clerks) always carry out the procedures according to those models. The purpose of the

standardization is to guarantee the constant high quality of work regardless of the individual competence of a given employee. The most often currently used notations for creating work-standardizing models are EPC (Event-driven Process Chain) standard of ARIS (Architecture of Integrated Information Systems) methodology 0 and BPMN (Business Process Modeling Notation) standard 0 developed by Object Management Group (OMG) 0.

ARIS, which stands for Architecture of Integrated Information Systems, is an approach to holistic modeling of processes performed in an organization. ARIS is also a name of a modeling platform which allows creating models according to the ARIS approach. In the heart of ARIS methodology is the concept of the ARIS House: a structured approach to modeling all information aspects related to processes. The ARIS House is a foundation for arranging various types of models and resources, and to define their relationships. It makes possible to reduce the complexity of individual models, while preserving the capability to analyze their contents and all types of their interrelationships. The ARIS House consists of five views for an organization which represent that organization from five different viewpoints: Organization hierarchy view, Data view, Function view, Product view, and Process view. The first four views focus on the different areas constituting the organization and present its static assets, while the Process View (also called Control View) focuses on the dynamic processes carried out in the organization and combines elements of the other four views. The modeling standard used in the Process view is the Event-driven Process Chain (EPC).

The EPC standard assumes modeling processes as sequences of events and functions. The event represents a change in the environment which may occur during process execution. The event is therefore a single point in time. The function represents activity that must be performed in the process. The function can be performed by a human or by a system or by a human supported by a system. The function requires inputs (information or materials), produces outputs (information or product) and may consume resources. The EPC standard assumes that function is always activated by an event and the result of executing the function is also always the event. Thus, events trigger functions, and the functions produce new events, which in turn trigger the next functions, which produce more events, etc. This leads to a chain of events and functions and hence the model name: Event-driven Process Chain.

The name of the second standard, BPMN, stands for Business Process Modeling Notation. BPMN is a standard for modeling processes at the conceptual level. It is based on the use of graphic symbols to create a model of a process. The primary goal of BPMN creators was to develop a graphical notation which would be easy to understand for both business managers and IT specialists. The second goal of the creators was to equip the graphical symbols with formal execution semantics which would enable to transform graphical process models into executable ones. To do that, all symbols are accompanied with properties that store data and information needed during the transformation. The current version of BPMN standard is limited to issues

related to process modeling only. Other types of modeling; e.g., organizational hierarchies, functional breakdowns, data and information modeling, needed for the full analysis of an organization remain out of the standard scope.

Nowadays in most of public agencies, the execution of administrative procedures is supported by *workflow management systems*. Such systems enable specifying and storing administrative procedure models (often referred as administrative procedure definitions), creating and managing their instances, and controlling their interactions with participants; e.g., clerks and information systems (e.g., public registers) 0.

Workflow management systems can be divided into two categories: *ad hoc* and *production* 0.

The ad hoc workflow management systems do not make direct use of administrative procedure models. In these systems, work and its flow are performed and supervised by clerks who are responsible for activities ordering and making coordination decisions during the execution 0. The ad hoc workflow management systems are capable of notifying about new work and after it is completed, of registering what was done, by whom, and when. A clerk must manually, outside the system, consult with the administrative procedure model how to do the work and to who forward its results. The level of administrative procedure automation offered by the ad hoc workflow management systems is very limited.

A significant step towards the automation of administrative procedures was made with the advent of the production workflow management systems. These systems are fed with models of administrative procedures and then use the models to automate the execution of those procedures 0. However, the production workflow management systems require the models to be created according to the monolithic approach. The monolithic approach means that a model has a form of a single end-to-end flow. The shape of the flow is defined and embedded in the model at design time and does not change during the execution. Even if the administrative procedures are modeled as a set of models and submodels, during execution the submodels are invoked according to the design.

Therefore, the monolithic approach to modeling makes it virtually impossible to create comprehensive administrative procedure models that would take into account all possible execution variants of these procedures, and within the specific variant – all detailed operational activities which should be performed to guarantee that a procedure course complies with all legal circumstances applicable in this variant.

This problem results from the fact that the course of administrative procedure is dictated by the provisions of the legislation. Two issues relating to the nature of the legislation significantly influence the form of administrative procedure models:

- Complexity of administrative legislation. If the models of administrative procedures were supposed to include all details reflecting all possible variants resulting from the provisions of legislation, they would have to be very developed and complex.

- The many-to-many relationship between models and legislation: one specific act has an impact on the form of many administrative procedures, and thus on their models, and one administrative procedure depends on the provisions of many acts. In this way, using the monolithic approach to the administrative procedure modeling, any change in the text of any act entails the need to update models of all procedures to which this act applies.

The workflow management systems currently running in public administration agencies are therefore based on general models. Such models do not include detailed activities and therefore are easy to develop, small in size and very little prone to the necessity of continuous updating in order to remain in compliance with legislation changes. For example, a typical activity that appears at the beginning of most models is “Check the correctness of the application and all necessary attachments”. The model containing such an activity does not specify the conditions when specific attachments are necessary, or which conditions the specific attachments should satisfy to be considered as being correct. A typical activity that appears in the middle of most administrative procedure models is “Inform the appropriate institutions of the actions taken” or “Turn to the appropriate agencies for opinions and agreements”. Models containing activities defined this way are very likely to remain valid even in case of multiple legislation changes.

The monolithic models presented above can be used to automate execution of administrative procedures by means of production workflow management systems. However, such automation requires intensive human (clerk) contribution. The clerk is required to analyze and interpret the general descriptions included in a monolithic model and decide how to translate them into specific operational activities that should be performed during the specific execution. The detailed model of each execution is thus created by the clerk in his mind on ad-hoc basis. Therefore, this type of automation should be named as the human-driven automation.

The advanced automation of administrative procedures that takes into account all possible legal circumstances that may arise during execution of these procedures is feasible using the techniques of artificial intelligence and declarative approaches to modeling. In such an approach, fragments of administrative procedures related to particular legal aspects should be modeled. Those fragmentary models can be used to dynamically construct the comprehensive models of administrative procedures tailored to the legal requirements of individual cases. The selection of the fragmentary models should be performed as a result of reasoning from the legal requirements. In order to make such an inference possible the requirements have to be formally represented using legal concepts defined in ontology. The ontology is a conceptual model of a specific domain 0.

In the recent years, an increasing role of ontologies and artificial intelligence in the field of law can be observed 0. The most important applications of legal ontologies in information systems and knowledge-based systems are the following 0:

- Information retrieval – can help users to find information relevant to his/her query by encoding knowledge about the meaning of concepts and relations among them;
- Translation of legal documents – can facilitate users to translate legal documents between different languages;
- Automated classification and summarizing – can support the automatic classification and making summaries of documents to help users to find relevant documents;
- Question answering – can enable to automatically answer questions based on the modeled legal knowledge;
- Agent technology – can enable intelligent autonomous communication between different computer systems by modeling rules governing that communication;
- Decision support and decision making – can be used to encode the decision steps and the contents of the decision rules in decision structures that allow making legal decisions or qualifications.

The most sophisticated application of ontologies in the legal domain is decision support and decision making. In this application, ontologies enable not only to model the concepts contained in legislation and the inference rules based on the modeled knowledge, but also facilitate decision making by public administration employees.

3. ONTOLOGY-BASED MODELING OF ADMINISTRATIVE PROCEDURES

The concept of advanced automation of administrative procedures is based on the assumption that IT systems take over execution of these parts of the procedures about which it is known in advance how to perform them. In the administrative procedures virtually all activities relate to dealing with information. Thus, IT systems should be responsible for routine processing of information, and humans (clerks) should be responsible for making subjective decisions based on information prepared by IT systems. Therefore, the advanced automation concept requires models of administrative procedures to contain all possible variants of those procedures, and within each variant, the detailed breakdown of all operational activities. Unfortunately, as discussed above, it is not possible to create such models using the monolithic approach.

In this work, the OMAP (Ontology-based Modeling of Administrative Procedures) approach for advanced automation of administrative procedures is proposed. The approach assumes that there is no modeling of administrative procedures in a form of end-to-end models. Instead of that, the course of the administrative procedure is dynamically constructed by ongoing adaptation to the current legal circumstances that occurred at the moment of the execution of that procedure. This way, each execution

of the specific administrative procedure has its own unique model, strictly corresponding to all characteristics of that execution.

The OMAP approach allows the implementation of advanced decision support and decision making systems, in which it is possible not only to write declaratively legal regulations and inference rules, but also to provide procedural descriptions of activities related to legal acts. Such an approach is especially suitable in the case of administrative law which regulates the course of administrative procedures. The administrative law in Poland, like in other civil law jurisdiction, consists of the following main areas 0:

- constitutional administrative law – defining the structure and functioning of public administration;
- substantive administrative law – regulating the rights and obligations of public authorities and citizens;
- procedural administrative law – governing the execution of administrative procedures.

To enable automation of administrative procedures it is required to model all of the above areas.

The OMAP approach is founded on three main concepts: an ontology, decision rules, and elementary processes. Both the constitutional and substantive administrative law is modeled declaratively using an ontology and decision rules. The procedural administrative law describing the course of administrative procedures is modeled using a procedural approach in the form of elementary processes.

An elementary process is a sequence of activities executed by humans (clerks) or IT systems to conduct a separate part of the administrative procedure constituting a logical whole. The example of the elementary process is checking if a person in a photograph submitted to be placed in an identification card wears dark glasses and if so, verification of a medical certificate of disability that requires constant use of such glasses. Decision rules specify legal circumstances under which specific elementary processes should be selected for execution during the course of an administrative procedure. The ontology contains definitions of administrative law concepts and relations between these concepts. These concepts are used in the specifications of elementary processes and decision rules.

The OMAP approach algorithm for constructing and executing administrative procedures consists of three main phases:

- legal circumstances analysis;
- decision rules activation;
- elementary process execution.

The first phase is the analysis of legal circumstances which appeared at a given moment of administrative procedure execution. The legal circumstances are represented by facts that are instances of concepts in the ontology. The ontology concepts can be real or abstract. An example of the real concept is an individual whose application is the subject of an administrative procedure; an example of abstract concept is

the individual's legal capacity which is the ability of individuals to make binding amendments to their rights, duties and obligations. Based on the results of the legal circumstances analysis, it is made a selection of decision rules to be activated in the next phase of the algorithm.

A single decision rule is a structure consisting of two sections: the *when* section and the *then* section. The section of *when* contains a conditional expression relating to the existence (or the lack of existence) of certain facts; i.e., legal circumstances, and to the values of attributes of those facts; i.e., specific characteristics of legal circumstances. The section of *then* contains the set of actions to be performed.

The conditional expression in the *when* section is considered as true if there exists a fact whose attributes have values equals to those specified in the expression. The rules whose conditions are evaluated as true during the analysis phase are selected for activation. Since all rules are independent of each other it is possible to select for activation more than one of them in a single analysis phase.

The rule activation consists in performing actions of the *then* section. The result of performing a single action can be either an operation on fact (asserting, modification, retracting) or a selection of an elementary process which should be instantiated and performed at the current moment of administrative procedure execution.

If performing the action resulted in changing at least one fact then the administrative procedure composition algorithm goes back to the legal circumstance analysis phase and next to the decision rule activation phase. This loop is repeated until the facts representing legal circumstances reach a stable state; i.e., as the result of activating decision rules, none of the facts will be changed.

If performing the action results in selecting elementary processes, the algorithm proceeds to the elementary process execution phase. These activities included in the elementary processes can be divided into three groups:

- activities related to documents; e.g., verifying application contents;
- activities related to data in public administration information systems; e.g., updating data in the national civil register;
- activities related to facts: asserting, modifying or retracting a fact.

If executing elementary process results in changing a fact, the algorithm goes back to the legal circumstances analysis phase.

The termination of the algorithm and thus the completion of the administrative procedure execution happen when decision rules activation phase results in no elementary process selection or when elementary process execution phase results in no fact change.

The architecture of the OMAP approach is presented in Fig. 1. The approach is composed of the following components: knowledge base, automation server, and execution portal.

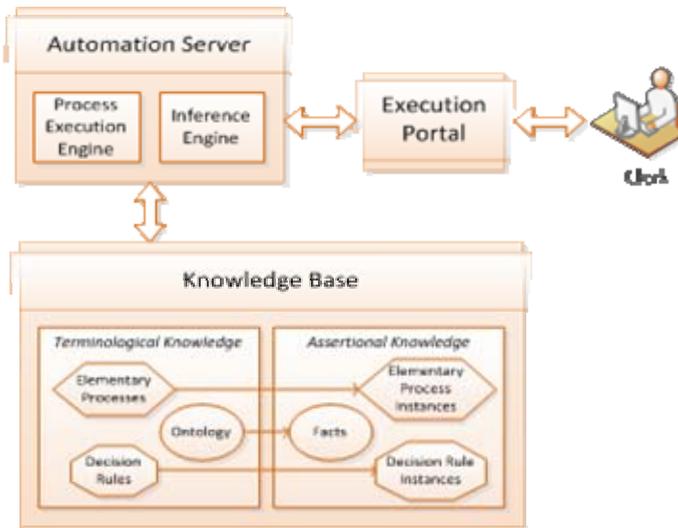


Fig. 1. The OMAP approach architecture

The knowledge base component stores comprehensive knowledge about administrative procedures. The knowledge is divided into *terminological* and *assertional*.

The *terminological knowledge* includes descriptions of concepts and their properties (roles). In OMAP approach, these are elementary processes, decision rules, and administrative law ontology, all as presented above. The most common formalisms used for modeling ontologies are: frames combined with first-order logic and description logic 0. However, other commonly used methods of ontology modeling are software engineering techniques such as UML and entity-relationship diagrams. In the OMAP approach, the ontology is modeled using UML class diagrams.

The *assertional knowledge* includes instances of concepts and the values of their properties (roles). In OMAP approach, these are elementary process instances, decision rule instances, and facts (i.e. instances of concepts defined in the ontology). Elementary process instances and decision rule instances which were created and executed during the course of a specific administrative procedure together with facts representing concrete legal circumstances that occurred during this course compose the instance of this procedure.

The automation server is responsible for the automated creation and execution of administrative procedures. It consists of two main components: a process execution engine and an inference engine. The process execution engine is responsible for executing instances of elementary processes. The selection of the elementary processes is performed by the inference engine based on decision rules evaluated against the facts.

Public clerks executing administrative procedures interact with the automation server using the execution portal. The basic functions offered by the execution portal

include: initiating new administrative procedure instances, monitoring the progress of the ongoing instances, and carrying out the tasks arising during execution of the instances.

4. EXEMPLARY ONTOLOGY-BASED MODEL

The OMAP approach has been proved in the automation of the exemplary aspect of administrative procedures. The aspect chosen for automation is the issue of determining the individual's legal capacity which occurs in the vast majority of administrative procedures. The issue of determining the individual's legal capacity is regulated by civil code and family law. Based on analysis of these acts, the ontology of concepts, the elementary processes, and the decision rules, all related to the issue of determining the individual's legal capacity, were designed. The ontology is presented in Fig. 2.

There are three types of the individual's legal capacity: full, limited, and none (lack of it). These types are modeled by three classes: *Full Legal Capacity*, *Limited Legal Capacity* and *No Legal Capacity*. All these classes inherit from the abstract class, *Legal Capacity*. The primary factor determining the type of the individual's legal capacity is a legal status. There are two types of the legal status: minority and majority, modeled by *Minority* and *Majority* classes inheriting from the abstract class, *Legal Status*. The primary factor determining the legal status is an age. Individuals under the age of 18 are recognized by law to be minors (their legal status is minority), individuals over the age of 18 are recognized to be major (their legal status is majority).

The individual's legal status also depends on the individual's marital status. The act of marriage permanently altered the legal status to the majority, even for individuals under the age of 18. The minority legal status is not taken away even in case of a divorce, a spouse's death or a marriage annulment. However, in virtually all countries there is a minimum age required by law at which an individual is allowed to get married. The age varies between countries. Also in many countries the age is different for men and women. In Poland, marriage may be entered only by a man who is over the age of 18 – the law does not provide any exception to this rule. This is not the case for women. As a rule, a woman also must be at least 18 years old, but for important reasons; e.g., pregnancy, a guardianship court may allow the woman who has attained the age of 16 years to get married if the circumstances indicate that it will be in accordance with the good of the family. The different types of the marital status are modeled by the hierarchy of classes under the abstract class, *Marital Status*. Four statuses referring to the act of marriage are connected to *Marriage Act* interface.

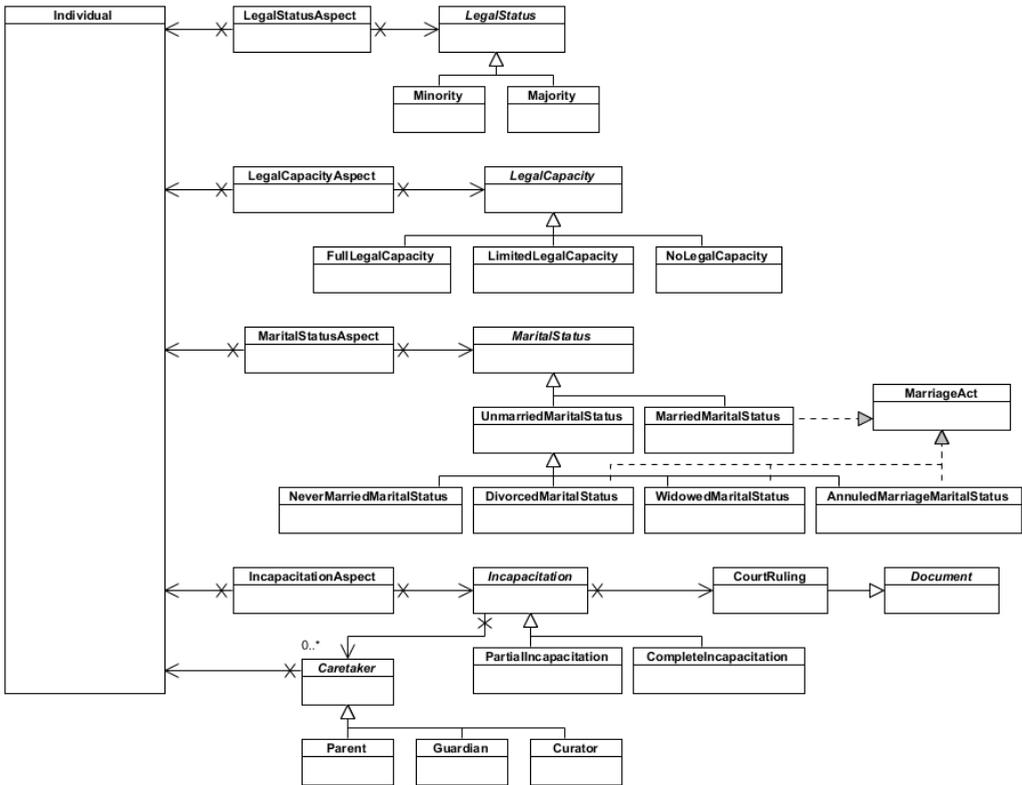


Fig. 2. Ontology of concepts related to the issue of determining the individual’s legal capacity

The legal capacity can also be altered by an incapacitation. There are two types of incapacitation: partial and complete, modeled by *Partial Incapacitation* and *Complete Incapacitation* classes inheriting from the abstract class, *Incapacitation*. Incapacitation is always established by a court ruling (*Court Ruling* class) which determines the incapacitation type and appoints caretakers. The caretaker can be a parent, guardian or curator, depending on the individual’s age and a family situation. The partial incapacitation altered the individual’s legal capacity to limited one, whereas the complete incapacitation altered it to none.

Elementary processes related to the issue of determining the individual’s legal capacity are presented in Table 1.

Table 1. Elementary processes related to the issue of determining the individual’s legal capacity

Elementary Process Code	Elementary Process Description
P_Individual_Data	Process execution engine: - Create a human task for collecting individual’s personal data. Public clerk:

	<ul style="list-style-type: none"> - Enter individual's personal data into a web form. Process execution engine: <ul style="list-style-type: none"> - Assert <i>Individual</i> fact.
P_Marital_Status	Process execution engine: <ul style="list-style-type: none"> - Connect to the national vital records and retrieve the current individual's marital status. - Assert <i>Marital Status</i> fact.
P_Incapacitation	Process execution engine: <ul style="list-style-type: none"> - Create a human task for verifying individual's incapacitation. Public clerk: <ul style="list-style-type: none"> - Enter information on the individual's incapacitation into a web form. Process execution engine: <ul style="list-style-type: none"> - If incapacitation then assert <i>Incapacitation</i> fact

Decision rules related to the issue of determining the individual's legal capacity are presented in Table 2.

Table 2. Decision rules related to the issue of determining the individual's legal capacity

Decision Rules Code	Decision Rules Description
R_Individual_Data	Rule: <ul style="list-style-type: none"> - Condition: no individual's personal data - Action: execute P_Individual_Data elementary process
R_Legal_Status_Aspect	Rule: <ul style="list-style-type: none"> - Condition: individual's age is under 18 - Action: assign minority legal status Rule: <ul style="list-style-type: none"> - Condition: individual's age is over 18 - Action: assign majority legal status
R_Legal_Capacity_Aspect	Rule: <ul style="list-style-type: none"> - Condition: individual's legal status is majority - Action: assign full legal capacity Rule: <ul style="list-style-type: none"> - Condition: individual's legal status is minority and age is over 13 - Action: assign limited legal capacity Rule: <ul style="list-style-type: none"> - Condition: individual's legal status is minority and age is under 13 - Action: assign lack of legal capacity (no legal capacity)
R_Marital_Status_Aspect	Rule: <ul style="list-style-type: none"> - Condition: individual has minority legal status, sex is femal, and age is over 16 - Action: execute P_Marital_Status elementary process Rule: <ul style="list-style-type: none"> - Condition: individual has ever committed the act of marriage - Action: assign majority legal status (regardless of the age or the fact if the current individual's marital status is unmarried)
R_Incapacitation_Aspect	Rule: <ul style="list-style-type: none"> - Condition: no facts about incapacitation

	<ul style="list-style-type: none"> - Action: execute P_Incapacitation elementary process <p>Rule:</p> <ul style="list-style-type: none"> - Condition: individual is completely incapacitated - Action: assign lack of legal capacity (no legal capacity) <p>Rule:</p> <ul style="list-style-type: none"> - Condition: individual is partially incapacitated - Action: assign limited legal capacity
--	--

To verify the correctness of designed artifacts, several research cases were carried out. The aim of each case was to create and execute the part of administrative procedure responsible for determining the legal capacity of a specific individual.

4.1. RESEARCH CASE 1

Assumptions of the Research Case 1:

- Individual's age: 35 years;
- Marital status: divorced;
- Incapacitation: no.

A sequence of actions performed by OMAP system to create and execute a part of administrative procedure for determining the individual's legal capacity is as follows:

1. No *Individual* fact → activate *R_Individual_Data* decision rules.
2. Execute *R_Individual_Data*:
 - Start *P_Individual_Data* elementary process.
3. Execute *P_Individual_Data*:
 - Public clerk enters individual's personal data into a web form.
 - Assert *Individual* fact.
4. No *Legal Status Aspect* fact → activate *R_Legal_Status_Aspect* decision rules.
5. Execute *R_Legal_Status_Aspect*:
 - Assert *Legal Status Aspect* fact.
 - Individual's age is over 18 → assert *Majority* fact.
6. No *Legal Capacity Aspect* fact → activate *R_Legal_Capacity_Aspect* decision rules.
7. Execute *R_Legal_Capacity_Aspect*:
 - Assert *Legal Capacity Aspect* fact.
 - *Majority* fact exists → assert *Full Legal Capacity* fact.
8. No *Marital Status Aspect* fact → activate *R_Marital_Status_Aspect* decision rules.
9. Execute *R_Marital_Status_Aspect*:
 - Assert *Marital Status Aspect* fact.
10. No *Incapacitation Aspect* fact → activate *R_Incapacitation* decision rules.
11. Execute *R_Incapacitation*:

- Assert *Incapacitation Aspect* fact.
 - *Legal Capacity* fact exists, different than *No Legal Capacity* → start *P_Incapacitation* elementary process.
12. Execute *P_Incapacitation*:
 - Public clerk checks in a web form that the individual is not incapacitated.
 13. End.

The final result of the Research Case 1 is a full legal capacity determined for the individual characterized in the case assumptions.

4.2. RESEARCH CASE 2

Assumptions of the Research Case 2:

- Individual's age: 17 years;
- Sex: female;
- Marital status: married;
- Incapacitation: no.

A sequence of actions performed by OMAP system to create and execute a part of administrative procedure for determining individual's legal capacity is as follows:

1. No *Individual* fact → activate *R_Individual_Data* decision rules.
2. Execute *R_Individual_Data*:
 - Start *P_Individual_Data* elementary process.
3. Execute *P_Individual_Data*:
 - Public clerk enters individual's personal data into a web form.
 - Assert *Individual* fact.
4. No *Legal Status Aspect* fact → activate *R_Legal_Status_Aspect* decision rules.
5. Execute *R_Legal_Status_Aspect*
 - Assert *Legal Status Aspect* fact.
 - Individual's age is under 18 → assert *Minority* fact.
6. No *Legal Capacity Aspect* fact → activate *R_Legal_Capacity_Aspect* decision rules.
7. Execute *R_Legal_Capacity_Aspect*:
 - Assert *Legal Capacity Aspect* fact.
 - *Minority* fact exists and age is over 13 → assert *Limited Legal Capacity* fact.
8. No *Marital Status Aspect* fact → activate *R_Marital_Status_Aspect* decision rules.
9. Execute *R_Marital_Status_Aspect*:
 - Assert *Marital Status Aspect* fact.

- *Minority* fact exists, individual's sex is femal, and individual's age is over 16 → start *P_Marital_Status* elementary process.
- 10. Execute *P_Marital_Status*:
 - Process connects to the national vital records and retrieves the current individual's marital status.
 - Assert *Married Marital Status* fact.
- 11. *Marriage Act* fact exists → activate *R_Marital_Status_Aspect* decision rules.
- 12. Execute *R_Marital_Status_Aspect*:
 - Retract existing *Legal Status* fact.
 - Assert *Majority* fact.
- 13. New *Majority* fact exists → activate *R_Legal_Capacity_Aspect* decision rules.
- 14. Execute *R_Legal_Capacity_Aspect*:
 - Retract existing *Legal Capacity* fact.
 - Assert *Full Legal Capacity* fact.
- 15. No *Incapacitation Aspect* fact → activate *R_Incapacitation* decision rules.
- 16. Execute *R_Incapacitation*:
 - Assert *Incapacitation Aspect* fact.
 - *Legal Capacity* fact exists, different than *No Legal Capacity* → start *P_Incapacitation* elementary process.
- 17. Execute *P_Incapacitation*:
 - Public clerk checks in a web form that the individual is not incapacitated.
- 18. End.

The final result of the Research Case 2 is a full legal capacity determined for the individual characterized in the case assumptions.

4.3. RESEARCH CASE 3

Assumptions of the Research Case 3:

- Individual's age: 16 years;
- Sex: female;
- Marital status: never married;
- Incapacitation: complete due to mental illness;
- Caretakers: natural parents.

A sequence of actions performed by OMAP system to create and execute a part of administrative procedure for determining individual's legal capacity is as follows:

1. No *Individual* fact → activate *R_Individual_Data* decision rules.
2. Execute *R_Individual_Data*:
 - Start *P_Individual_Data* elementary process.

3. Execute *P_Individual_Data*:
 - Public clerk enters individual's personal data into a web form.
 - Assert *Individual* fact.
4. No *Legal Status Aspect* fact → activate *R_Legal_Status_Aspect* decision rules.
5. Execute *R_Legal_Status_Aspect*:
 - Assert *Legal Status Aspect* fact.
 - Individual's age is under 18 → assert *Minority* fact.
6. No *Legal Capacity Aspect* fact → activate *R_Legal_Capacity_Aspect* decision rules.
7. Execute *R_Legal_Capacity_Aspect*:
 - Assert *Legal Capacity Aspect* fact.
 - *Minority* fact exists and age is over 13 → assert *Limited Legal Capacity* fact.
8. No *Marital Status Aspect* fact → activate *R_Marital_Status_Aspect* decision rules.
9. Execute *R_Marital_Status_Aspect*:
 - Assert *Marital Status Aspect* fact.
 - *Minority* fact exists, individual's sex is femal, and individual's age is over 16 → start *P_Marital_Status* elementary process.
10. Execute *P_Marital_Status*:
 - Process connects to the national vital records and retrieves the current individual's marital status.
 - Assert *Never Married Marital Status* fact.
11. No *Incapacitation Aspect* fact → activate *R_Incapacitation* decision rules.
12. Execute *R_Incapacitation*:
 - Assert *Incapacitation Aspect* fact.
 - *Legal Capacity* fact exists, different than *No Legal Capacity* → start *P_Incapacitation* elementary process.
13. Execute *P_Incapacitation*:
 - Public clerk checks in a web form that the individual is incapacitated.
 - The form directs him to ask about details on this issue. All information is entered into the form.
 - Assert *Complete Incapacitation* fact.
 - Assert *Court Ruling* fact.
 - Assert two *Parent* facts, one for an individual's mother and the other for a father – due to court ruling, parents are individual's caretakers; assert the references from *Complete Incapacitation* to *Parent* facts; assert two *Individual* facts (mother's and father's personal data); assert the references from *Parent* facts to the appropriate *Individual* facts.

14. *Complete Incapacitation* fact exists → activate *R_Incapacitation* decision rules.
15. Execute *R_Incapacitation*:
 - Retract existing *Legal Capacity* fact.
 - Assert *No Legal Capacity* fact.
16. End.

The final result of the Research Case 3 is a lack of legal capacity (no legal capacity) determined for the individual characterized in the case assumptions.

4.4. RESEARCH CASE 4

Assumptions of the Research Case 4:

- Individual's age: 5 years.

A sequence of actions performed by OMAP system to create and execute a part of administrative procedure for determining individual's legal capacity is as follows:

1. No *Individual* fact → activate *R_Individual_Data* decision rules.
2. Execute *R_Individual_Data*:
 - Start *P_Individual_Data* elementary process.
3. Execute *P_Individual_Data*:
 - Public clerk enters individual's personal data into a web form.
 - Assert *Individual* fact.
4. No *Legal Status Aspect* fact → activate *R_Legal_Status_Aspect* decision rules.
5. Execute *R_Legal_Status_Aspect*:
 - Assert *Legal Status Aspect* fact.
 - Individual's age is under 18 → assert *Minority* fact.
6. No *Legal Capacity Aspect* fact → activate *R_Legal_Capacity_Aspect* decision rules.
7. Execute *R_Legal_Capacity_Aspect*:
 - Assert *Legal Capacity Aspect* fact.
 - *Minority* fact exists and age is under 13 → assert *No Legal Capacity* fact.
8. No *Marital Status Aspect* fact → activate *R_Marital_Status_Aspect* decision rules.
9. Execute *R_Marital_Status_Aspect*:
 - Assert *Marital Status Aspect* fact.
10. No *Incapacitation Aspect* fact → activate *R_Incapacitation* decision rules.
11. Execute *R_Incapacitation*:
 - Assert *Incapacitation Aspect* fact.
12. End.

The final result of the Research Case 4 is a lack of legal capacity (no legal capacity) determined for the individual characterized in the case assumptions.

5. CONCLUSIONS

The proposed OMAP approach enables advanced automation of administrative procedures. The approach has been verified for several research cases characterized by very different legal circumstances. The courses of administrative procedure aspects constructed and executed according to the OMAP approach have complied with the legal circumstances resulted from the characteristics of those cases.

The research results provide a basis to formulate the thesis that the use of the OMAP approach significantly reduces the complexity of administrative procedure models compared to the traditional monolithic approach. At the same time, compared to the monolithic approach, final models of administrative procedure created in the OMAP approach provide a higher level of details on operational activities. Modeling at a high level of details is a necessary prerequisite for the advanced automation of administrative procedures, defined as replacing to the greatest possible extent human labor with work of IT systems. In turn, such advanced automation is one of the main foundations of the electronic government.

The future research aims at discovering and specifying administrative legislation modeling patterns related to ontology concepts, decision rules and elementary processes.

REFERENCES

- [1] JAGIELSKI J., *Administrative Law*, In: Introduction To Polish Law, Frankowski S. (Ed.), Kluwer Law International, 2005, 153–187.
- [2] MOMMERS L., *Ontologies in the Legal Domain*, In: Theory and Applications of Ontology Theory and Applications of Ontology: Philosophical Perspectives, Poli R., Seibt J. (Eds.), Springer, 2010, 265–276.
- [3] GEORGAKOPOULOS D., HORNICK M., SHETH A., *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure*, Distributed and Parallel Databases, Vol. 3, No. 2, Boston, Kluwer Academic Publishers, 1995, 119–153.
- [4] GOMEZ-PEREZ A., FERNANDEZ-LOPEZ M., CORCHO O., *Theoretical Foundations of Ontologies*, In: Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web, London, Springer-Verlag, 2004, 1–45.
- [5] GRUBER T.R., *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, Vol. 5, No. 2, 1993, 199–220.
- [6] KOBIELUS J., *The rhythm of work: A buyer's guide to workflow tools*, Network World – Collaboration, Vol. 12, No. 42, 1995, 12–18.

- [7] VAN DER AALST W.M.P, BERENS P.J.S., *Beyond Workflow Management: Produkt-Driven Case Handling*, In: Group'01 Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, New York, ACM, 2001, 42–52.
- [8] VAN DER AALST W.M.P., Mans R.S., Russell N.C., *Workflow Support Using Procllets: Divide, Interact, and Conquer*, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 32, No. 3, 2009, 16–22.
- [9] VAN ENGERS T., BOER A., BREUKER J., VALENTE A., WINKELS R., *Ontologies in the Legal Domain*, In: Digital Government – E-Government Research, Case Studies, and Implementation, Chen H., Brandt L., Gregg V., Traunmüller R., Dawes S., Hovy E., Macintosh A., Larson C.A. (Eds.), New York, Springer, 2008, 233–261.
- [10] WORKFLOW MANAGEMENT COALITION, *WFMC-TC-1011 Ver. 3 Terminology and Glossary*, 1999, <http://www.wfmc.org/>.
- [11] *ARIS Platform*, http://www.softwareag.com/corporate/products/aris_platform/.
- [12] *Business Process Modeling Notation*, <http://www.bpmn.org/>.
- [13] *Object Management Group*, <http://www.omg.org/>.

Adam CZARNECKI*, Cezary ORŁOWSKI*

APPLICATION OF ONTOLOGY IN THE ITIL DOMAIN

Business standards tend to be less formal in description than strict technical norms. Authors of this chapter suggest applying ontological description (powered by the capabilities of the OWL language) to express Information Technology Infrastructure Library (ITIL). One of the goals of this initial study is to determine the usefulness of such semantic model in organizations that adopted or plan to adopt ITIL.

1. INTRODUCTION

In order to provide valuable services to their customers, IT organizations try to optimize their processes. To achieve a greater maturity in service management these organizations often adopt standards or codes of best practices (ITIL, ISO 9001, ISO 20000, ISO/IEC 38500, eTOM, COBIT, eSCM, and CMMI to mention a few). Every organization that implements any of the aforementioned standards, however, has to pass through the bumpy road of learning its principles and details along with aligning the organization to conform to the introduced practices. The amount and complexity (and sometimes ambiguity) of such standards along with the need for reflecting them in the enterprise architecture are a starting point for searching some ‘middleware’ solution based on ontological description.

Authors of this work would like to address certain issues of developing ontology in the domain of The Information Technology Infrastructure Library (ITIL) such as competency questions, identifying sources of knowledge, building ontology and assessing economic rationale for such project. Most of these topics are not novel [9]. In

* Gdańsk University of Technology, Faculty of Management and Economics, Department of the Information Technology Applications in Management, ul. Narutowicza 11/12, 80-233 Gdańsk, Poland.

particular, research conducted in Hewlett-Packard laboratories [5] seems promising as HP's Service Manager (current version 9.21) is in compliance with ITIL V3. This software application is an example of several technologies supporting IT management that is in the range of interest of authors. Yet, the initial stage of described areas in ITIL ontology does not allow for far-reaching conclusions with more scientific work ahead.

2. ITIL AS AN ONTOLOGY DOMAIN

The Information Technology Infrastructure Library (ITIL) [7] describes best practices for service management [6]. ITIL contributes some best practices that become the frameworks for improving service quality. These best practices are codified in books (5 in the 3rd version of ITIL) that present best practices for service management in the following areas (see: Fig. 1):

- Service Strategy: addresses how to set a strategy to meet customer needs and provide value.
- Service Design: includes dealing with architecture, technology, processes, information, and organizational issues; it also encompasses collecting business requirements along with designing and developing appropriate service solutions, processes, and measurement systems.
- Service Transition: describes how to implement designed services into operations; it covers areas of the change management, service asset and configuration management, knowledge management, release and deployment management, service evaluation, and service validation and testing.
- Service Operation: describes best practices to ensure services availability that are split into five key components: event management, incident management, request fulfillment, problem management, and access management.
- Continual Service Improvement: describes how to collect and analyze data on processes performance (KPIs—Key Process Indicators) via an improvement process and provides best practices on service management and service reporting.

These five books constitute a so-called ITIL Core: best practice guidance applicable to all types of organizations who provide services to a business. The ITIL provides also a component named The ITIL Complementary Guidance which is a complementary set of publications with guidance specific to industry sectors, organization types, operating models, and technology architectures.

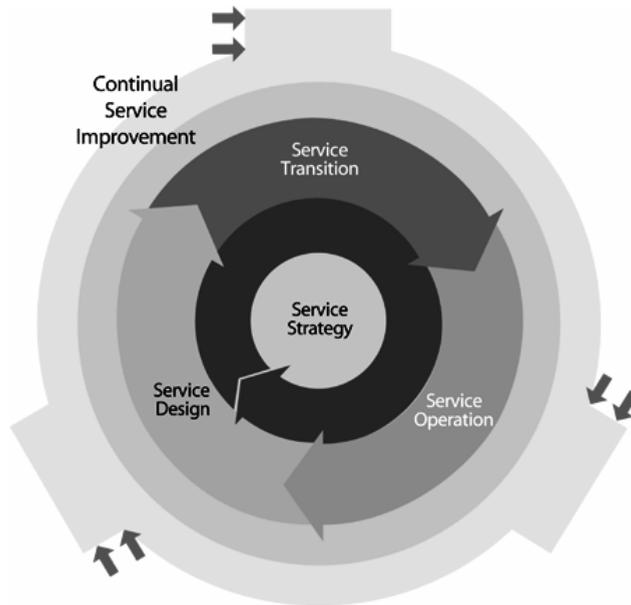


Fig. 1. The ITIL Core [7]

3. COMPETENCY QUESTIONS

The aim, scope and depth of the ontology depend on the knowledge it is supposed to store and conclude. The software engineering discipline has a requirements modeling technique of ‘use cases’ that a system should respond to. When specifying ontology a knowledge worker has a tool of ‘competency questions’ – a list of issues that ontology should (must) reply to.

For the domain of the ITIL the following competency questions can be formulated:

- What concepts/notions are used in the ITIL (vocabulary)?
- What services and processes are covered in each best practice book? / To which book does a given service or process belong to?
- What is the sequence of each process or service?
- How do processes/services relate to each other?
- What roles/responsibilities are assigned to each service/process?

These questions were chosen to illustrate the process of ontology development and can easily be expanded by adding other elements of the ITIL domain.

4. SOURCES OF KNOWLEDGE

The knowledge on ITIL (V3) is generally stored in five books mentioned earlier with almost 1700 pages in total. Each book consists mostly of the text descriptions illustrated with semi-formal figures (about 171 in the Service Strategy book) and tables (21 in the Service Strategy book). There is a glossary of 387 definitions and 99 acronyms at the end of each book.

These forms of knowledge representation can be ordered according to their expressive precision (ascending):

1. textual descriptions,
2. figures,
3. tables.

The sequence of 2 and 3 can be a bit controversial: if ITIL figures were representing formal models such as UML or BPML, they would be placed below tables as being semantically richer. However, they lack such formalisms and therefore are here listed as less precise than tables.

To shift the knowledge on ITIL on the higher level of expressive precision it can be modeled as an ontology domain. This ontology development process could be aided if a content metamodel is present. Such metamodel has been introduced in other acknowledged IT standard: The Open Group Architecture Framework (TOGAF) – in its 9th version [10]. One of TOGAF's chapters has been intended to present all main concepts and relations among them so the ontology developer can easily capture the most essential part of the framework and reflect it in the formal ontology.

5. ONTOLOGY DEVELOPMENT METHODS

A tool without knowledge on how to utilize it is virtually useless. One must add a method of operating the given tool. Those methods can be expressed as a detailed algorithm, formal standard or recommended guidelines (as, for example, these given in [8]) or best practices (bearing in mind what the ITIL really is).

When developing ontology one should be aware that there is no single correct way to model knowledge. The scope and the perception of the domain in question, the conceptualization process may all vary depending – among other factors – on the planned application of the given ontology.

A second assumption of ontology development states that it is an iterative process. It seems unlikely that one should obtain a complete, accurate model that fully meets the requirements at the first attempt. And even if, the ontology is often a part of a larger system which may – throughout its life-cycle – also enforce later changes.

Thirdly, one should to assume that the ontological model reflects the concepts from the domain of interest. Because one of the purposes for creating ontologies is to share common understanding of the domain, it is inadvisable to model the domain in the way that would be operational for software agents but completely unreadable for any human being except its originator.

Bearing in mind these three assumptions, one can proceed with creating ontology. One of the approaches divides this process into eight steps [1, p. 226]:

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.
4. Define the classes and the class hierarchy.
5. Define the properties of classes (slots).
6. Define the facets of the slots.
7. Create instances.
8. Check for anomalies.

As one can see, these steps correspond to the object-oriented approach. However, it is not identical with programmers' point of view. It puts stress on structural properties of a class rather than on the methods and operational properties which well addresses the features of the OWL.

Somewhat similar guidelines but more related to the software engineering were presented in METHONTOLOGY by Fernández, Gómez-Pérez and Juristo [3]. They put more stress on the documents such as requirements specification and the activities that can lead to the concurrent ontology development in the evolving cycle (see: Fig. 2).

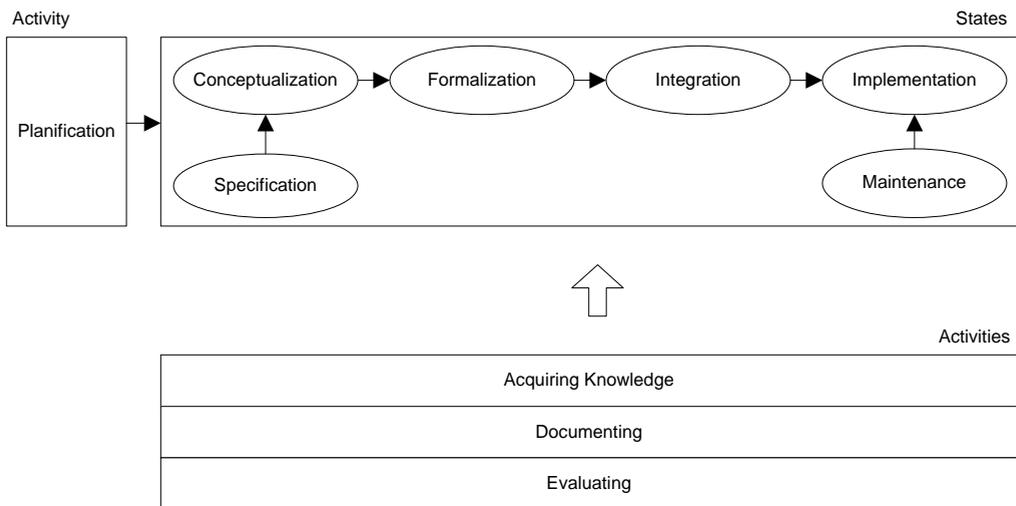


Fig. 2. METHONTOLOGY: states and activities [3]

The third approach that can be mentioned refers to the design patterns concept [4, p. 69]. In this approach one can see ontologies as a “knowledge skeletons of domains”, a specific building blocks that represent some common knowledge models and which can be reused and developed incrementally.

Gašević, Djurić and Devedžić [4, p. 177] write about ‘the gap between the knowledge of software practitioners and AI techniques’ and present a few software engineering approaches to ontology development such as UML representation and other Model-Driven Architecture-based standards that aim to adopt a standardized architecture named Ontology Definition Metamodel (ODM).

The aforementioned approaches do not exhaust the subject of methodologies for building ontologies. Fernández [2] lists (and describes) also:

- Methodology by Uschold and King: based on the experience of developing the Enterprise Ontology, an ontology for enterprise modeling processes.
- Methodology By Grüninger And Fox: based on the experience in developing the TOVE project ontology within the domain of business processes and activities modeling.
- The approach of Amaya Berneras et al.: set within the Esprit KACTUS project. One of the objectives of the KACTUS project is to investigate the feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it.
- SENSUS-Based Methodology: ontology for use in natural language processing, developed at the Information Sciences Institute (ISI) natural language group to provide a broad-based conceptual structure for developing machine translators.

To summarize this section, one can state that while the selection of the language for the ontology development seems rather obvious as the OWL is recommended for the Semantic Web, the method for conducting all required activities is still discussed and tends to focus on the life cycle or on the craft of the ontology modeling. And as it still lacks of the general and complex development framework, perhaps one can expect some specific methods/methodologies in more narrow areas, for example in the rule- or fact-based expert systems.

6. CASE STUDY: A BASIC PROCESS

A journey of a thousand miles begins with a single step, an old Chinese proverb says. So before the whole ITIL domain is covered in ontology, a simple example of ‘translating’ the documentation to the ontology will be presented.

ITIL’s Service Strategy book presents a figure 2.14 labeled ‘A basic process’ (see: Fig. 3). It consists of boxes and arrows representing entities and flows in this sample process. Three boxes representing activities are included in the ‘Process’ module.

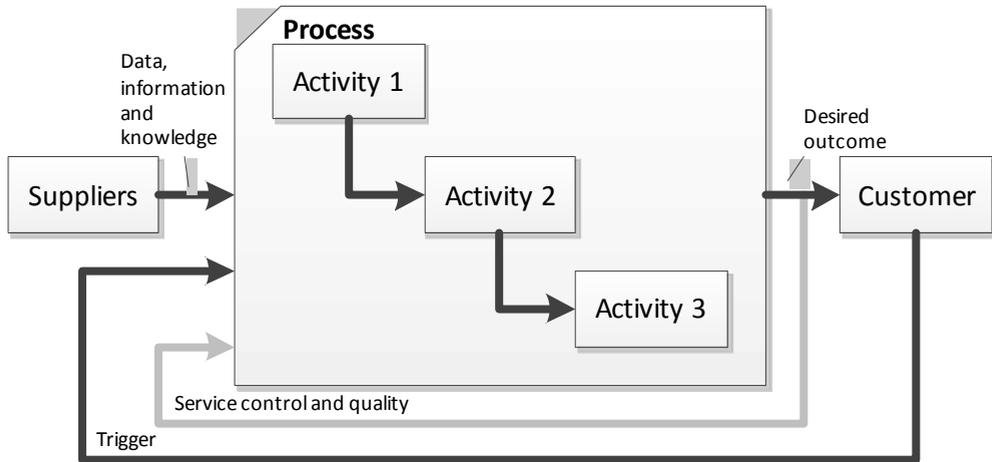


Fig. 3. A basic process [7]

The ontology engineering toolbox provides two main structures: classes and properties. The first impression, especially for the UML users, would be to model boxes from figure 3 as classes and treat labels above arrows as relations. Such approach soon happens to be misleading, because there are no verbs in arrows' descriptions. They are, in fact, also classes while names of the relations must be devised from the interpreted meaning of each arrow. All arrows outside the 'Process' compartment can be described using two object properties:

- 'supplies': links the class represented by the box on the left of the arrow with the class which name serves as an arrow's label.
- 'is_supplied_to': links the class which name serves as an arrow's label with the class represented by the box on the right of the arrow.

Arrows that point from one activity to another represent a sequence in the process, so their semantics can be read as 'precedes' when linking activities in the top-down order or 'follows' when analyzing activities in the opposite direction. This pair of object properties is the fine example of the inverse properties, i.e. if we state that Activity 1 precedes Activity 2 then it can be inferred that Activity 2 follows Activity 1.

The next concept to cover is the inclusion of activities within the process. This can also be modeled as the object property: the 'consist of' property has been inserted between 'Process' and all its activities. It may also be advisable to add the inverse property such as 'is a part of' that would enforce the semantic description of the aggregation nature of process and its activities.

Another fact that an ontology engineer can derive from the fig. 3 is that for all three activities a more general class (superclass) called 'Activity' can be created in order to maintain a "tidiness" of the ontology. Less obvious but perhaps also desired could be arranging 'Data', 'Information' and 'Knowledge' concepts into the batch under the

common name: ‘Content’. This categorization may depend on the competency questions that the given ontology should be capable of answering. In this case study such questions pertaining the content categorization have not appeared, but if there would be such requirement the monotonic nature of the ontology allows adding new facts in the next iteration of the ontology development process.

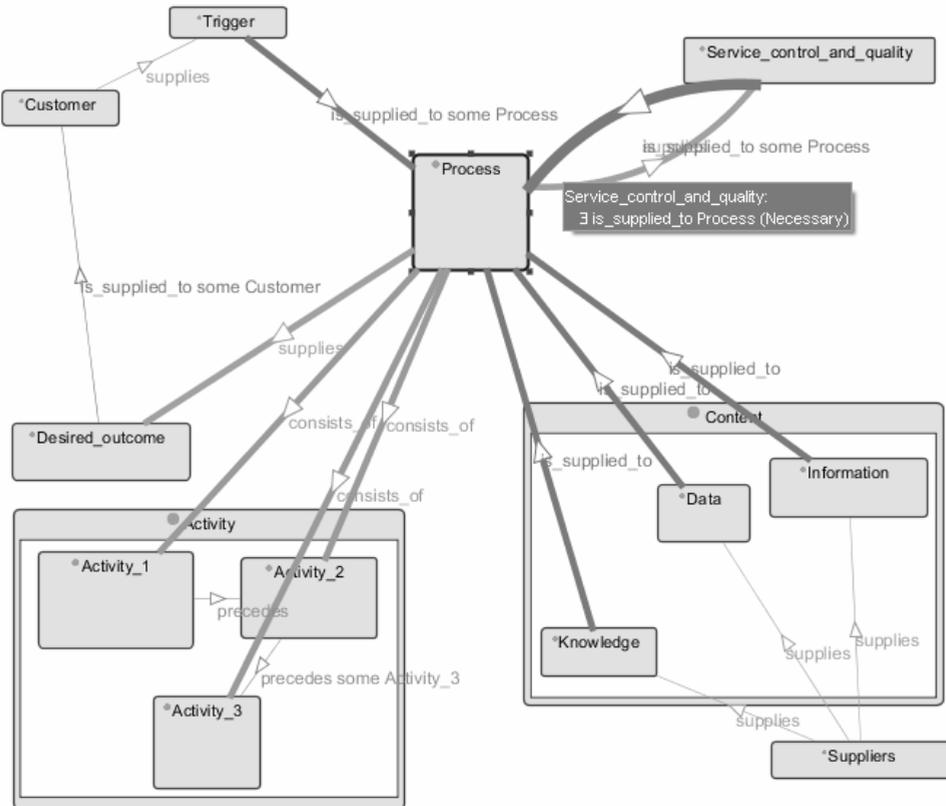


Fig. 4. A basic process—a graphical representation of the ontology (Jambalaya plug-in in Protégé)

Figure 4 depicts a graphical illustration of the ontology described above. There are 14 classes and 5 object properties which are used multiple times. Such representation may seem more complex and less clear to the person viewing both fig. 3 and 4, but in fact provides formal arrangement of the process concepts and a means to utilize this knowledge both by software agents and people (providing a proper interface).

It may be worth adding that it took one hour of the knowledge engineer to develop ontology presented on fig. 4. This single example cannot of course be a sample for assessing the total amount of time required to develop ITIL ontology. The goal of mentioning this fact is to remind an economic (cost-side) aspect of ontology development projects.

7. SUMMARY AND FUTURE WORK

The knowledge representation process issues shown in this work has shown that at least a part of the ITIL domain can be described in the formal and computable manner – ontology (with Web Ontology Language as a recommended language). Ontology allows for relatively easy knowledge classification. It also allows detecting certain inconsistencies within the documentation (which was a part of similar research on TOGAF Core metamodel). The question arises whether the effort put in the ontology development brings benefits that could not be achieved in shorter time, with fewer resources engaged and with at least the same quality.

The results of applying ontology to a very narrow scope of the only one diagram do not constitute a representative sample to draw general conclusions. However this experiment sheds some light on the path of the future research – the team involved in standard's (or best practices, or recommendations) development may use ontology as a 'back-office' model to share a common vocabulary and understanding of the standard among the people involved in that collaboration. The parallel development of the ontology in that body could support the process of the standard elaboration and revision phases.

The second party that could benefit from the ontological model of the standard such as ITIL are organizations that intend to implement such standard. Although further studies are needed, the current experience of authors suggests that such effort of developing ontology in the domain of the given IT standard on the client-side is resource-consuming and requires skills that are rare on the job market. Therefore the economic efficiency of such activities would be low. But if the standard body would provide ontology of the standard in question – as it was suggested in the preceding paragraph – along with the interface that would allow answering competency questions formulated by the client organization, the efficiency of the ontology use might increase. Authors of this chapter would like to consider it as a 'working hypothesis' that needs further examination.

Future works – aside from the aforementioned issues – should now focus on the narrower scope of ITIL: Service Operations with a stress put on the specification and development phases, and on providing end-users an interface to store and retrieve knowledge to/from ontology as the problem of querying ontology has not been here discussed.

To conclude, IT business standards can be a promising array of domains of applying ontologies that may give these standards a sound formal semantic (logic) shape that is based on the Semantic Web architecture. It allows verification and validation of the model presented in the given standard. Yet, the degree in which such ontology can support the standards' bodies and standards' users should be further studied.

REFERENCES

- [1] ANTONIOU, G., VAN HAMERLEN, F., *A Semantic Web Primer – 2nd ed.*, The MIT Press, Cambridge, Massachusetts–London, England, 2008.
- [2] FERNÁNDEZ, M., *Overview Of Methodologies For Building Ontologies*, In: IJCAI99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, Stockholm, Sweden, 1999.
- [3] FERNÁNDEZ, M., GÓMEZ-PÉREZ, A., JURISTO, N., *METHONTOLOGY: from Ontological Art towards Ontological Engineering*, In: Proceedings of the AAAI97 Spring Symposium, March 1997, AAAI Press, Menlo Park, California, USA, 1997, 33–40.
- [4] GAŠEVIĆ, D., DJURIĆ, D., DEVEDŽIĆ, V., *Model Driven Engineering and Ontology Development – Second Edition*, Springer-Verlag, Berlin/Heidelberg, Germany, 2009.
- [5] GRAUPNER, S., MOTAHARI-NEZHAD, H.R., SINGHAL, S., BASU, S., *Making Processes from Best Practice Frameworks Actionable*, In: Business Conversation Manager: Facilitating People Interactions in Outsourcing Service Engagements, Benatallah, B., Casati, F., Kappel, G., Rossi, G. (Eds.), LNCS, Volume 6189/2010, Springer, Berlin/Heidelberg, Germany, 2010, 468–481.
- [6] HURWITZ, J., BLOOR, R., KAUFMAN, M., HALPER, F., *Service Management for Dummies*, Wiley Publishing, Indianapolis, USA, 2009.
- [7] *Information Technology Infrastructure Library (ITIL), Version 3, 5 Volumes*, <http://www.itil-officialsite.com>, 2007.
- [8] NOY, F.N., MCGUINNESS, D.L., *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Lab. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford, USA, 2001.
- [9] SHANGGUAN, Z., GAO, Z., ZHU, K., *Ontology-Based Process Modeling Using eTOM and ITIL*, In: Research and practical issues of enterprise information systems II Volume 2, L. Xu, Tjoa, A., Chaudhry, S. (Eds.), IFIP, Volume 255/2008, Springer, Boston, USA, 2008, 1001–1010.
- [10] *The Open Group Architecture Framework, Version 9*, The Open Group, 2009.

*e-learning, access control,
adaptable user interfaces,
SOIL, XACML, SAML*

Adam WÓJTOWICZ*, Jakub FLOTYŃSKI*,
Dariusz RUMIŃSKI*, Krzysztof WALCZAK*

SECURING LEARNING SERVICES ACCESSIBLE WITH ADAPTABLE USER INTERFACES

E-learning gains popularity as a modern form of education, but e-learning systems have been primarily designed for use on full-size desktop computers. Recently, mobile devices such as PDAs and smartphones are increasingly often used to access the Internet. Such devices may allow students to access learning content at any time and any place, thus greatly enhancing learning efficiency. However, such devices have limitations regarding their screen size, interaction capabilities, computing power, battery lifetime, and communication bandwidth. To overcome these limitations and to enable effective use of learning content on mobile devices, adaptable user interfaces adjusted to a particular device, a particular user, and a particular context are needed. Since such interfaces are dynamically generated, it becomes critical to ensure that they reflect a consistent state of users' credentials, determined by a security policy of the system. In this work, an extension of a SOA-based learning system, called MILES 2.0, is presented. The extension assures authenticated, precisely authorized, and confidential usage of learning services. The solution builds on existing languages for defining security policies (XACML) and for creation of adaptable user interfaces (SOIL), and provides distributed SOA-based security management.

1. INTRODUCTION

E-learning platforms become an increasingly popular tool supporting education at schools and universities. Recent surveys show that more than 90% of higher education establishments use an e-learning system to support their education process [6]. Rapid development of e-learning systems is reflected by intensive research in this area con-

* Poznań University of Economics, Department of Information Technology, Mansfelda 4, 60-854 Poznań, Poland.

cerning, e.g., decision support [15], design patterns [12], or three-dimensional Web interfaces [5]. Numerous e-learning platforms are available. Examples include Moodle [7], Sakai [11], and ATutor [2].

Most of the existing e-learning platforms have been optimised for use in typical desktop-based Web environments, utilizing a Web browser as the presentation layer. However, at the present time, mobile devices such as smartphones and browser-equipped enhanced phones are increasingly often used for accessing the Internet. This trend is particularly strong in the case of young people, who want to use mobile devices to access educational resources. However, as compared to personal computers, mobile devices are usually much more limited in their capabilities, such as communication bandwidth, processing power as well as the offered presentation and interaction methods.

In [14], a SOA-based learning system, called MILES 2.0, has been presented. It has been implemented as an extension of the Web 2.0 ERC platform [13] to enable remote learning with both desktop and mobile devices. The system is equipped with adaptable user interfaces built with the use of Web Services provided by the underlying Learning Content Management System (LCMS). These interfaces can replace traditional Web-based interfaces of LCMS systems. In MILES 2.0, user interfaces are dynamically generated by ASIS – Adaptable SOA Interface System [4]. These interfaces are described by parameterized templates encoded in SOIL – Service Oriented Interface Language. Adaptation of user interfaces (UI) reflects information such as access device capabilities, user profile, and context. SOA enables building complex distributed systems, but assurance of security in such systems becomes a major challenge. The MILES 2.0 e-learning system and the ASIS adaptation framework do not provide proper security solutions.

In this work, we present an extension of the MILES 2.0 system providing access control capabilities, i.e., user authentication and fine-grained authorization of user actions, as well as confidentiality and integrity of communication. Authentication is the confirmation of a principal identity with a specified or understood level of confidence. Authorization is the process of determining whether the particular entity has the right to perform some action on some resource. Authentication and authorization are the main elements of access control, which provides protection of resources against unauthorized access [9]. Confidentiality and integrity of data transmitted over the network is assured by encryption. It may be performed using standard protocols and mechanisms such as HTTPS or VPN without significant influence on system architecture.

Implementation of access control mechanisms in MILES 2.0 requires three elements. Firstly, the process of UI generation must be bound to system security policy in order to present only the functionality which is available for a given user. For this purpose, the SOIL language has been enriched with additional commands and parameters. Secondly, a mechanism for describing security policies related to Web Services is

needed. For this purpose, an extension of the XACML language has been developed. Finally, the system architecture must be extended to include a Security Manager compatible with the remaining components.

The remainder of this work is organized as follows. In Section 2, the state of the art in security frameworks and description languages is presented. In Section 3, security requirements for SOA-based learning systems are described. In Section 4, architecture of a SOA-based learning system including the security components is explained. In Section 5, an example dataflow in the system is provided. Finally, in Section 6, conclusions and future works are presented.

2. SECURITY FRAMEWORKS AND DESCRIPTION LANGUAGES

Access control can be performed with the use of different existing frameworks such as Java Authentication and Authorization Service [10], Spring Security [3], or Apache Shiro [1]. The common drawback of these frameworks is utilization of vendor-specific approaches for the security policy description instead of standardized vendor-independent solutions.

This disadvantage does not occur in the context of two standardized security description languages. Security Assertion Markup Language (SAML) [9] is an XML dialect standardized by OASIS. SAML provides means for authorization and authentication with a strong focus on the Single-Sign-On (SSO) technique. SAML enables exchange of security attributes, named assertions, and simple authorization data between distributed system components, especially through Web Services. SAML may be utilized together with other XML-based languages such as SOAP and WSDL for Web Services, XACML for security policies, and SOIL for generation of interfaces in SOA applications.

Assertion is the basic data structure in SAML. It may be interpreted as follows: assertion A was given by supplier S at time T, in relation to service WS, provided that conditions C are fulfilled. Assertion contains statements which may inform about attributes assigned to a user, authentication, or authorization decision. For example, an assertion can state that at 10:00 AM the student identified by the identifier "Jan Nowak" has been authenticated by his password transmitted through a secure channel to access the course on computer science provided by the university.

In the data processing model, there are two sides that can be distinguished: Identity Provider (IP) and Service Provider (SP). The IP implements services responsible for authentication. The SP exchanges security attributes with IP to secure usage of services. The SAML standard supports a number of message exchange schemes (profiles) under SSO such as Web Browser SSO and Enhanced Client or

Proxy [9]. The main advantage of SAML is vendor-independence and straightforward integration with existing software solutions. SAML is a proper solution for complex authentication schemes, but does not enable definition of complex security policies.

eXtensible Access Control Markup Language (XACML) is another XML dialect standardized by OASIS [8]. XACML provides means of definition of security policies, requests, and responses. The XACML specification additionally defines architecture of security policy engine (SPE), which is responsible for enforcement and management of security policies.

Rules are basic elements of XACML. A rule is a triplet consisting of a subject, an action, and a resource for which the access may be granted. Rules may then be grouped in a policy, together with an associated target specifying a set of subjects requiring to perform actions. Finally, policies may be grouped into policy sets. An XACML request consists of a subject, a resource, and an action name. Both policy and request may contain also additional attributes which may determine the roles associated with users. An XACML response may have four values: permit, deny, indeterminate (if many policies are suitable for the request) or not applicable (if no policy suits the request).

XACML implementation architecture includes five components, namely Policy Enforcement Point, Context Handler, Policy Information Point, Policy Decision Point, and Policy Administration Point [8]. The Policy Enforcement Point (PEP) is responsible for enforcing security policies. It receives requests from a requester and communicates with the Context Handler to obtain decision about an access of a subject to a resource. The Context Handler (CH) is responsible for transforming messages received from the PEP into XACML request and sending them to the Policy Decision Point (PDP). The CH is also responsible for transforming XACML responses obtained from PDP into a response format understandable by PEP. The request sent to PDP is built according to the subject, resource, and action retrieved from the Policy Information Point. The Policy Information Point (PIP) is responsible for storing and retrieving subject, resource, action, and additional attributes in/from a database when required by the CH. The Policy Decision Point (PDP) is responsible for verifying XACML requests received from the CH against the set of security policies obtained from the Policy Administration Point. The PDP sends XACML responses back to the CH. The Policy Administration Point (PAP) is responsible for storing and retrieving policy sets in/from a database when required by the PDP.

Main advantages of XACML are vendor-independence and ability to handle complex security policies. The main disadvantages of XACML are the relative inefficiency of processing XML requests and difficulties with integrating it with existing access control mechanisms.

3. SECURITY REQUIREMENTS IN SOA-BASED LEARNING SYSTEMS

In this section, requirements for implementation of security in SOA-based learning systems are discussed. We address mainly access control to learning services, not confidentiality and integrity of communication, which is performed by standard protocols and mechanisms. The following requirements should be satisfied.

1. Due to the multiplicity of services, Single-Sign-On (SSO) method for authentication is desirable.
2. A rule should be the primary entity for making authorization decisions. It should be a triple combining: a resource, an action to be performed, and a decision which may be either to permit or to deny.
3. Rules should be grouped into roles. Several roles may be assigned to a user. Such an approach enables complex descriptions of access rights, while keeping management of these rights relatively simple.
4. Standard centralized Role-Based Access Control (RBAC) model is not sufficient because of the presence of an independent layer responsible for UI adaptation and distributed model of service provisioning and consumption. Therefore, it is necessary to extend RBAC with functionality of Attribute-Based Access Control providing additional attributes for security policies, e.g., time when an action is permitted.
5. Authorization decisions should be based on security policies (SP), which combine different roles with different conditions.
6. SP should be able to describe access rights to Web Services, hierarchical structures of resources, and parent-child dependencies between actions.
7. SP should be described by a standardized high-level, vendor-independent, machine-processed language.

Furthermore, in MILES 2.0, the subsystem responsible for access control has to cooperate with ASIS. An SP extends UI templates and is verified against a request while UI is generated. For example, user's role is a part of SP and influences the way in which a UI is adapted. Consistency of SP and the UI template must be assured "by-design". Inconsistencies can lead to inaccessibility of services or improper template generation.

4. SECURITY ARCHITECTURE OF A SOA-BASED LEARNING SYSTEM

In order to fulfill the above-described requirements, a method of securing learning services is proposed. It combines with the prior implementation of MILES 2.0 e-learning system and embraces three new elements. Firstly, the SOIL language has to be extended by elements referring to the security policies. Secondly, the XACML language must be adapted to enable control of usage of Web Services. Finally, the MILES 2.0 e-learning system has to be extended with new components providing

authentication requests and decisions. Next, ASIS-CH communicates with the Policy Enforcement Point (PEP) to invoke learning services required by the MC. To generate a UI, the UI-TR is utilized. It stores SOIL descriptions of UIs and information about bindings to security policies, e.g., roles associated with a given UI template. The UI-TR accesses security policies and information about services stored in Access Services for XACML Policy Repository (ASXPR) and Access Services for Service Register (ASSR). The main purpose of the analysis of security policies in ASIS-CH is optimization of the UI generation in order not to expose functionality in the UI that is not allowed for a given user. Enforcement of security policies is implemented in the Security Manager.

4.2. XACML EXTENSIONS

XACML is suitable for protecting static resources. However, for controlling the usage of Web Services it has to be extended (defined more precisely) in all its main elements: subject, resource, and action.

Three categories of subjects are distinguished. The first category are users with roles assigned to them for a relatively long period. An example may be a student who connects to learning repositories from a campus network. The second kind of subjects are users described by mutable sets of attributes. This category is suitable for distributed models, e.g., protection of learning services for “ad-hoc” mobile clients. The third type of subjects represent Web Services associated with sets of roles or attributes. Such subjects represent services invoking other services.

Furthermore, two types of resources: services and operations are distinguished by introducing new data types describing them. It permits to model hierarchical elements, e.g., particular operations of selected services can be protected selectively. Additionally, an attribute is added to the rule. It specifies whether a dependent (child) operation may be invoked from the inside of the parent operation. It is possible to analyze inter-operation dependencies. If an invocation of a child operation is not allowed in a given rule and a protected operation invokes such an operation, the parent operation cannot be invoked, even if all other conditions are satisfied.

The modified XACML action strictly refers to the operation and represents one of two possible activities. The Invoke action describes parameters passed within the request and their ranges of possible values. The Receive action obtains a value returned by the operation as a result of the invocation.

All presented modifications of the XACML language should be correctly handled by the PDP and the SM components.

4.3. THE SECURITY MANAGER

The Security Manager (SM) for learning services is a set of components of the extended MILES 2.0 e-learning system responsible for controlling access to learning

services. It is an adaptation of the functionality and system architectures provided by SAML and XACML specifications.

The SM consists of several components outlined with thick black line in Fig. 1. Identity Provider acts as the IP from the SAML specification. It utilizes the User Repository (UR) and the Role Repository (RR) to provide SSO authentication of access to different learning services. The IP generates SAML assertions for the ASIS-CH. In the proposed solution, the Enhanced Client or Proxy (ECP) SAML profile is used. Contrary to a regular passive client, ECP is able to implement complex logic. The interface generator (ASIS) plays a role of enhanced (“intelligent”) client – it allows to choose the desired IP or SP (if different are available) and filters services depending on the interaction context and the role of the authenticated user.

Qualifications of PEP and PDP are similar as described earlier. The PEP acts as the SP defined in SAML specification. It receives requests from the ASIS-CH and sends authorizations or access control queries to the PDP. Next, access control decisions are enforced, i.e., an MC is permitted or denied to access a particular service.

The PDP transforms requests obtained from the PEP into XACML documents, retrieves proper security policies and roles associated with users, and verifies XACML requests against the security policies. Responses are translated from XACML to the form understandable for the PEP. The PDP respects the new XACML capabilities described in Section 4.2.

Access Services for XACML Policy Repository (ASXPR), XACML Policy Repository (XPR), and Role Repository play roles similar to the Policy Information Point and Policy Administration Point combined together. They store and provide policies and roles assigned to users.

5. EXAMPLE OF DATA FLOW IN THE SYSTEM

In this section, an example of authentication and authorization for learning repositories is presented. The example concerns a student who uses a mobile client (MC) to access a course on computer science. The user Jan Nowak is represented as a subject with assigned `student` role. The course is represented by one service containing three hierarchically-dependent operations. The `GetListOfTopics` operation provides a list of all topics in the course. The `GetTopicContent` operation retrieves content of the particular topic. Both services are available only through an interface which is provided by the third – `GetCourse` operation. The security policy (`E-learning policy`) enables performing the `Invoke` and `Receive` actions on `GetCourse` and also on child operations for the user with the `student` role everyday between 8:00 AM and 9:00 PM. A user invokes `GetCourse`, which causes an immediate invocation of two child services. Interactions between particular compo-

nents of the learning system correspond to the numbers of arrows in Fig. 1.

1. An MC sends a user name (`Jan Nowak`) and the password to the ASIS-CH.
2. The ASIS-CH starts a user session and sends a SAML assertion with the user name and the password to the IP.
3. The IP authenticates the user and communicates with the UR and the RR to retrieve his/her roles. A SAML assertion, which contains authentication of information (positive), a token, and attributes describing user roles (`student`), is created and sent back to the ASIS-CH.
4. The ASIS-CH sends to the UI-TR a template request, a security policy request, the service structure (schools/courses) request, and the service request.
5. A UI template encoded in SOIL is received from the UI-TR. Along with the template there are provided service-related data obtained from the Service Register and security policies-related data from the SPR. The UI template is sent to the ASIS-CH.
6. UI representation is dynamically generated in the ASIS-CH depending on: the user role (`student`), service structure (3 dependent services), and security policy (`E-learning policy`) which binds all those data with conditions (e.g., limited time of access). The UI representation is sent back to the MC.
7. The MC sends the required operation name (`GetCourse`) to the ASIS-CH.
8. The ASIS-CH passes the request and the user role as a SAML assertion to the PEP.
9. The PEP uses SAML assertion to build a request, which contains the operation (`GetCourse`), the action (`Invoke`), and the user name (`Jan Nowak`) and sends it to the PDP.
10. The PDP requests the security policy (`E-learning policy`) from the ASXPR.
11. The ASXRP sends back `E-learning policy` and user's roles to the PDP.
12. The PDP creates an XACML request based on the operation, the action, and user's roles. It verifies the XACML request against the security policy and makes an authorization decision (`permit`). The decision is translated to the form readable for the PEP and sent to the PEP.
13. Operation invocation (`GetCourse`) is enforced by the PEP.
14. Results of the operation are sent back to the PEP.
15. The PEP sends results to the ASIS-CH.
16. The ASIS-CH returns results back to the MC.

6. CONCLUSIONS AND FUTURE WORK

In this work, an extension of a SOA based learning system, called MILES 2.0, to support access control (authentication and authorization), and confidentiality (using encryption) has been proposed. The extension encompasses three elements. Firstly, the interface template language (SOIL) is extended to support security policies and the ASIS frame-

work is modified appropriately. Secondly, the XACML language is extended to enable description of security policies related to Web Services. Finally, the Security Manager is introduced as a distributed subsystem for securing learning services.

It is a crucial issue to enable simultaneous access to the system for hundreds of users. In future work, efficiency of the solution should be tested especially in scope of SAML and XACML processing and interactions between Web Services.

A possible system extension is support for security managers based on different technologies. It is desirable to enable exchange of the XACML-based SM for an implementation in JAAS, Spring Security or any other available technology.

REFERENCES

- [1] APACHE, Apache Shiro, <http://shiro.apache.org>, retrieved 20.07.2011
- [2] ATUTOR, ATutor website, <http://atutor.ca>, retrieved 26.01.2011
- [3] BEN ALEX, Acegi Security. Reference documentation, <http://www.acegisecurity.org/reference.html>, retrieved 07.07.2011
- [4] CHMIELEWSKI J., WALCZAK K., WIZA W., *Mobile Interfaces for Building Control Surveyors*, In: Software Services for e-World, IFIP Advances in Information and Communication Technology, vol. 341, Cellary W., Estevez E. (Eds.), The 10th IFIP WG.6.11 Conference on e-Business, e-Services and e-Society I3E 2010, Buenos Aires, Argentina, Springer, 2010, 29-39
- [5] DI CERBO F., DOREDO G., PAPAELLO L., *Integrating a Web3D interface into an e-learning platform*, In: Proc. of the 15th Int. Conf. on Web 3D Technology, Los Angeles, 2010, 83-92
- [6] GREENBERG A.D., ZANETIS J., *Distance Education and e-Learning Metrics Survey 2010*, <http://www.wainhouse.com/files/edu/WR-DETech-2010-ExecSum.pdf>, retrieved 26.01.2011
- [7] MOODLE, 2011, Moodle website, <http://moodle.org>, retrieved 07.07.2011
- [8] OASIS, eXtensible Access Control Markup Language 2 (XACML) v. 2.0, 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, retrieved 07.07.2011
- [9] OASIS, Security Assertion Markup Language (SAML) V1.1, 2003, <http://www.oasis-open.org/standards#sam1v2.0>, retrieved 07.07.2011
- [10] ORACLE, Java Authentication and Authorization Service (JAAS). Reference Guide for the Java 2 SDK, Standard Edition, v. 1.4, <http://download.oracle.com/javase/1.4.2/docs/guide/security/jaas/JAASRefGuide.html>, retrieved 07.07.2011
- [11] SAKAI, Sakai Project, <http://sakaiproject.org>, retrieved 26.01.2011
- [12] RAJAM. S., CORTEZ R., VAZHENIN A., BHALLA S., *Design patterns in enterprise application integration for e-learning arena*, In: Proceedings of the 13th International Conference on Humans and Computers, Klyuev V., Cohen M., Aizu-Wakamatsu, 2010, 81-88
- [13] WEB 2.0 ERC, 2011, Web 2.0 ERC Project, <http://www.web20erc.eu/>, retrieved 07.07.2011
- [14] WALCZAK K., WIZA W., RUMIŃSKI D., CHMIELEWSKI J., WÓJTOWICZ A., *Adaptable User Interfaces for Web 2.0 Educational Resources*, In: Proc. of the 9th International Conf. "Multi-media in Business and Management", Wisła, Poland, 2011 (accepted for publication)
- [15] ZORILLA M., GARCIA D., ALVAREZ E., *A decision support system to improve e-learning environments*, In: Proceedings of the 2010 EDBT/ICDT Workshops, Daniel F., Delcambre L., Fotouhi F. (Eds.), Lausanne, Switzerland, 2010, Article 11, 8 pages

Maciej ZIĘBA*

ENSEMBLE METHODS FOR CUSTOMER CLASSIFICATION IN SERVICE ORIENTED SYSTEMS

This work concentrates on the problem of customer classification in service oriented systems. The typical classification problems observed in service oriented systems are distinguished in this publication: imbalanced data and cost-sensitive learning problems. First of the stated problems is related with disproportions of examples from different classes in training set further used to build the classifier. Second of the mentioned problems is observed, when the misclassification cost values are significantly different between classes. The ensemble classifier with switching class labels is proposed as classification method which solves two of identified problems. The proposed solution uses misclassification-based probabilities to construct artificially balanced datasets, which are further used to build base classifiers of the ensemble. Final classification using proposed solution is made by using simple majority voting method.

1. INTRODUCTION

Development of expert systems is observed in last few decades. It mostly indicated by the growing availability of data and computational resources and the need of business processes automation. Decision making components can be often identified in modern processes and the costs of executions of such processes are relatively high due to the fact, that decisions are made by human experts (medical consultants, bankers or financial specialists). They are usually supported by expert systems but the final decision is most cases made by a human. The growing accuracy of decisions made by data mining algorithms and their ability to solve data mining problems like missing values or imbalanced data make such algorithms ready to be used in the modern business process without human computer interaction. The development of decision making methods goes in parallel with growing popularity of solutions included in Service

* Wrocław University of Technology, Faculty of Computer Science and Management , Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.

Oriented Architecture (SOA) paradigm [15]. The fundamental issue of SOA is a service, which represents the encapsulated functionality, in general form described by inputs, outputs and well-structured semantic description. During the business process execution the services are called to deliver necessary functionality to the process. It can be easily observed, that the functionality representing decision making solutions should be delivered to the process by calling proper data mining services. Such solution, comparing to traditional system oriented approach, eliminates the cost of human work and makes the execution process much more flexible.

This work concentrates on special case of decision making process called pattern classification. Pattern classification is the discipline of assigning class labels to the objects described by some features. More precisely, the goal of pattern classification is to take the vector of features (also named as attributes) and assign to one of the J class labels from given set of features $\{C_1, \dots, C_J\}$ [1,9,20]. The process of classification is made using classification method called the classifier Ψ . The classifier can be constructed by some experts and represented as a set of decision rules or trees, but in most real-life problems it is built in supervised training process. To train the classifier, the set of observations represented by vectors of features values \mathbf{x}_n and corresponding classes y_n (which takes the coded class values from set $\{C_1, \dots, C_J\}$) called training set ($S_N = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$) is used.

It can be easily observed, that classifier is fully described, if two issues are defined: how to use the classifier if unlabelled object must be classified and how to train the classifier using training set S_N . Various types of classifiers are presented in the literature: parametric classifiers like Support Vector Machines or Neural Networks and nonparametric classifiers like decision trees and rules. Usually, couple of methods can be used to train one type of classifiers. For instance, decision rules can be constructed using is PRISM [5] method or RIPPER algorithm [9]. In the last few decades growing popularity of so called ensemble classifiers is observed [20]. The idea of ensemble classifiers is very simple: each of K experts make independent decision about class assignment and final decision is made according to decisions gained from the experts. More precisely, there are K base classifiers on the first stage (denoted by $\Psi_k^{(1)}$) and one fusion algorithm on the second stage (denoted by $\Psi^{(2)}$).

According to current research the ensemble classifiers are better quality classifiers than individuals, if the classification accuracy of decisions made by each of base classifiers separately is on acceptable level and diversity among base classifiers is high enough [3,11,24]. Moreover, ensemble classifiers can be successively applied to eliminate negative effects connected with typical problems with data [12,13,25,26].

As it could be easily noticed the key problem of classification is to build good quality classifier using set of examples stored in training set S_N . Unfortunately, data

included in training sets is usually not perfect. This section concentrates on three, very important problems with data which can be observed in classification field: missing values problem [13], imbalanced data problem [12] and cost-sensitive classification problem [12]. First problem occurs, when some values of examples in training set are missing. The data can be incomplete due to various circumstances: inability of making some tests because of the lack of medical equipment or inappropriate test type for certain patients medical systems (medical systems), refusal to answer some questions in questionnaire by responders or inability to read handwritten answers (bank systems). The second of considered problems is composed with the situation, when the numbers of examples from different classes in training set are imbalanced. If we consider the two class classification task the imbalanced data problem occurs, when the cardinality of examples labeled by one class (called majority class) is significantly higher than cardinality of examples labeled by the second class (called minority class). The problem with imbalanced data is often considered in parallel with cost-sensitive classification problem. In two class classification such problem can be observed, when the cost of classifying object from minority class as an object from majority class is significantly higher than the cost of classifying object from majority class as an object from minority class.

The goal of this work is to propose classification method which deals with imbalanced data problem in parallel with cost-sensitive classification problem. The ensemble classifier with switching class labels is proposed as possible solution for the two, stated above classification problems. As many authors had shown in their works [7,8,12,16] ensemble classifiers are successively applied to solve the imbalance data and cost-sensitive classification problems, mainly, because they give the possibility of generating artificiality balanced datasets used to construct base classifiers $\Psi_k^{(1)}$ on the first stage.

2. RELATED WORKS

The main disadvantage of present training algorithms is a sensitivity to disproportions in training set. Most of standard algorithms assume or expect balanced class distributions. If such algorithm is used to train the classifier with imbalanced dataset, the algorithm fails to properly represent the distributive characteristics of the data. If the cost-sensitive classification problem is considered in parallel the value empirical risk can be unacceptably low. It is obvious, that accurate techniques for dealing with imbalanced data problem must be considered during building good-quality classifiers. Various methods are used to deal with imbalanced data problem [12].

Random Oversampling and Undersampling. The idea of random oversampling is very simple. New set E composed only of examples from minority class is created by

sampling with replacement examples from imbalanced training set S_N . Balanced set is created by merging sets E and S_N . In random undersampling majority examples are eliminated from S_N to balance it. This approach can be only applied if the distribution of majority class in training set will not be changed in undersampling process.

Informed Undersampling. To save the distribution in undersampling process the procedure of examples selection must be intelligent. The idea of informed undersampling is to remove those examples, which are least needed and select only important elements from majority class. Interesting informed undersampling approach is presented in [22]. Authors of this approach present various techniques for imbalanced data problem which are based on K-NN algorithm. According to the presented results the best of proposed solutions is to select those majority examples whose average distance to the three farthest minority class examples is the smallest.

Synthetic Sampling with Data Generation. On the other hand, synthetic samples can be generated in smart way to balance minority class with majority class. Good example of such type of methods is synthetic minority oversampling technique (SMOTE) presented in [6]. This approach uses K-NN to create artificial examples. Consider example \mathbf{x}_n from minority class and set of minority examples which are nearest neighbours of the considered example. The new artificial example is going to be situated on the path between \mathbf{x}_n and selected neighbours. Various extensions of SMOTE algorithm are observed in literature: Border-SMOTE [17], which evaluates the candidates for synthetic examples which are created using SMOTE or SMOTE + Tomek links [1], which uses so called Tomek links analysis to make data clearing after applying clearing procedure.

Cluster-Based Sampling Methods. Cluster-based sampling algorithms are particularly interesting because they provide an added element of flexibility that is not available in most simple and synthetic sampling algorithms, and accordingly can be tailored to target very specific problems [12]. The example of cluster-based sampling method was proposed in [18]. Authors propose clustering solution to be applied for both majority and minority examples included in training set. K-means algorithm [2] is proposed as clustering algorithm, however, various grouping techniques can be used instead. After applying clustering algorithm two groups of clusters are achieved: group of K_{\min} clusters of examples from minority class and group of K_{maj} examples from majority class. Then, the cluster with highest cardinality C_{\max} is selected from the majority group. Next, the oversampling process is made for each of majority clusters to create the situation in which the number of examples in each majority cluster will be equal to the cardinality of the selected cluster C_{\max} . In next step the number of examples from majority class (those examples from training set together with those artificial examples created using oversampling) is divided by the number of clusters

from minority group to get the desired cardinality of each of minority clusters. The oversampling is made to achieve the desired cardinality of each minority cluster. In this strategy, the oversampling process is applied to both classes, so the imbalances between-class and inner class is eliminated in parallel.

Ensemble-based approaches. Ensemble techniques are widely applied to increase classification accuracy [20,24], to deal with missing values problem [14,25] or to construct classifiers in incremental training [19]. Ensembles are also used for imbalanced data problem [7,8,16]. One of the ensemble solutions for imbalanced problem is SMOTEBoost algorithm [7]. This method uses SMOTE sampling to generate artificial examples for minority class for each boosting iteration. In such approach each of created base classifiers concentrates more on minority class. As a consequence, the final classification decision made by ensemble classifier is more balanced. The other example of ensemble approach for imbalanced problem is DataBoost-IM method [16]. This algorithm also uses boosting approach to generate base classifiers. For each of boosting iterations hard examples are identified in current training set. The hard example, which is also called "seed" by the authors, is difficult-to-learn example. Next, each of identified hard examples is used, as a seed, to generate artificial examples. These artificial examples are added to the current training set and the boosting distribution is modified respecting newly added samples. Presented approach uses hard examples as seeds for sampling and hard examples are identified according to error rates between classes. It means, that sampling procedure is flexible and depends on the number of identified seeds. Other boosting-based approach is presented in [8]. Briefly, RAMOBoost adaptively ranks minority class instances at each learning iteration according to the sampling probability distribution that is based on the underlying data distribution, and can adaptively shift the decision boundary toward difficult-to-learn minority and majority class instances by using a hypothesis assessment procedure.

3. ENSEMBLE CLASSIFIER WITH SWITCHING CLASS LABELS

The main idea of switching class labels techniques is to increase diversity of base classifiers by changing class labels in datasets S_{N_1}, \dots, S_{N_k} , which are used to train them. More precisely, if we consider the example \mathbf{x}_n which belongs to j -th class, it will be switched to i -th class with estimated switching probability $\tilde{p}(i | j)$. It can be observed, that main problem in switching class labels techniques is to find the estimated probability values $\tilde{p}(i | j)$. Breiman in his work [4] proposes procedure of estimating the probability which uses overall switching rate the proportions of the different class labels in the original training set. Breiman highlights, that switching probabilities should be estimated in such way, that the probability distribution of random variable representing features and classes must be saved. The work on switching

class labels techniques started by Breiman, was extended by authors of [23]. They made tests for several switching probabilities estimators and concluded, that for large number of base classifiers saving mention probability distribution is not necessary to get high classification accuracy.

This dissertation presents slightly different switching class labels technique. Initial testing procedure is proposed to identify tendencies of classification incorrectness between class labels. The switching probabilities are estimated basing on error rates between classes. According to the proposed procedure, it is more probable to switch labels between classes which are difficult to separate using single classifier and less probable if the classes are almost perfectly separable. Comparing to solution presented in [4] and extended in [23] this approach does not require setting any parameters and maintaining class distribution. Despite the fact, that this approach can be successively used for balanced data problem the extended version which deals with imbalanced dataset and cost-sensitive problems is going to be presented.

The essence of presented approach is to identify tendencies of incorrect classification between class labels. To achieve this, the initial base classifier $\Psi_0^{(1)}$ is trained using complete set of examples S_N . Next, the performance of classifier is tested on the same set S_N . During testing procedure, for each pair of class labels (i, j) , the number of examples from i -th class classified as member of j -th class group is calculated. Denote this number as $n_{i,j}$ and the number of possible class labels as J . In the next step training sets S_{N_1}, \dots, S_{N_K} are generated using re-sampling with replacement as it is made in bagging algorithm. For each training set S_{N_k} switching class labels is made. The probability that class label i is replaced by class label j is equal:

$$\tilde{p}(i | j) = \frac{n_{i,j}}{\sum_{j=1}^J n_{i,j}} = \frac{n_{i,j}}{n_i} \quad (1)$$

where n_i is the number of elements from i -th class in training set. In general, if classifier has tendency to classify object from i -th class as objects from j -class (the value of $n_{i,j}$ is high) the probability of replacing class j with class i is high. Such misclassification tendencies are especially observed in imbalanced data problems. Consider two-class classification task, in which minority class is coded with j_{\min} and majority class with j_{\max} . In most imbalanced dataset problems following condition is satisfied:

$$\tilde{p}(j_{\max} | j_{\min}) < \tilde{p}(j_{\min} | j_{\max}) \quad (2)$$

As a consequence, datasets S_{N_1}, \dots, S_{N_K} used to train base classifiers $\Psi_1^{(1)}, \dots, \Psi_K^{(1)}$ become more balanced, because more examples from majority class are switched to

minority class comparing to opposite situation. It means, that there are less mistakes related with such misclassification made by such constructed ensemble classifier, comparing to the misclassification performance of single decision tree. Additionally, if cost-sensitive learning problem is observed in classification task, what means that:

$$c(j_{maj} | j_{min}) > c(j_{min} | j_{maj}) \quad (3)$$

(where $c(i | j)$ represents the cost of classifying the object from i -th class as objects from j -class) the probability $\tilde{p}(j_{maj} | j_{min})$ should be set on 0, because the fundamental aim in such case is to minimise the number of objects from minority class classified as objects from majority class ($n_{j_{min}, j_{maj}}$). Concluding, if we consider the two-class classification problem, in which imbalanced dataset is used to build the classifier in cost-sensitive learning process, following switching probabilities should be used in proposed method:

$$\tilde{p}(j_{maj} | j_{min}) = 0, \tilde{p}(j_{min} | j_{maj}) = \frac{n_{j_{min}, j_{maj}}}{n_{j_{min}}} \quad (4)$$

and, as a consequence, we can write the probabilities of saving the class labels:

$$\tilde{p}(j_{min} | j_{min}) = 1, \tilde{p}(j_{maj} | j_{maj}) = \frac{n_{j_{min}, j_{min}}}{n_{j_{min}}} \quad (5)$$

The description of pseudocode of ensemble classifier with switching class labels is presented on Figure 1. In the first step of the algorithm, classifier $\Psi_0^{(1)}$ is built on training set S_N . The classifier is not the component of ensemble structure, it is created only to identify missclassification tendencies and, as a consequence, to estimate value of probability $\tilde{p}(j_{min} | j_{maj})$, what is made in second step of the procedure. Next, base classifiers $\Psi_1^{(1)}, \dots, \Psi_K^{(1)}$ are created in the loop in following way. First, training set $S_{N_{1,k}}$ is generated by *bootstrap sampling* from the initial training set S_N . *Bootstrap sampling* is simply sampling with replacement, which is used in commonly used in many ensemble approaches like bagging algorithm. Following the procedure, duplicates are removed from the training set $S_{N_{1,k}}$. It is made to avoid the situation in which there are two the same examples and for one example the class label will be switched, so the same object will be represented by two examples from two, different classes. Reduced dataset, which is denoted by $S_{N_{2,k}}$ is transformed to dataset $S_{N_{2,k}}^{(1)}$ using switching procedure. Each object from training set $S_{N_{2,k}}$, which is member of majority class j_{maj} , is switched to minority class j_{min} with the probability $\tilde{p}(j_{min} | j_{maj})$. Gained in such way training set is used to build base classifier $\Psi_k^{(1)}$.

The presented ensemble classifier with switching class labels uses simple voting combiner formally described using equation:

$$\Psi^{(2)}(\mathbf{x}) = \arg \max_{j=1,\dots,J} (\Psi_k^{(1)}(\mathbf{x}) = j) \quad (6)$$

where the number of possible class labels is equal J and the classes are coded using integer values $1, \dots, J$. Indicator $I(\bullet)$ returns 1 if expression inside is true and 0 otherwise. It means, that new object will be classified to the class, which will be selected by majority of base classifiers.

INPUTS:

Training set: $\mathcal{S}_N = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$

Number of base classifiers: K

OUTPUTS:

Base classifiers: $\Psi_1^{(1)}, \dots, \Psi_K^{(1)}$

PROCEDURE:

1. Build classifier $\Psi_0^{(1)}$ on training set \mathcal{S}_N
 2. Estimate probability value $\tilde{p}(j_{\min} | j_{maj})$ by testing $\Psi_0^{(1)}$ on training set \mathcal{S}_N
- for k from 1 to K do**
- 3.1 Generate training set $\mathcal{S}_{N_{1,k}}$ from \mathcal{S}_N using bootstrap sampling
 - 3.2 Generate $\mathcal{S}_{N_{2,k}}$ by removing duplicates from training set $\mathcal{S}_{N_{1,k}}$
 - 3.3 Set $\mathcal{S}_{N_{2,k}}^{(0)} = \emptyset$
- for i from 1 to $N_{2,k}$ do**
- if** ($y_i = j_{maj}$)
- 3.4.1.1 Generate random value r from $[0,1]$
- if** ($r \leq \tilde{p}(j_{\min} | j_{maj})$)
- 3.4.1.1.1 Set $y_i = j_{\min}$
- end if**
- end if**
- 3.4.2 Add example (\mathbf{x}_i, y_i) to $\mathcal{S}_{N_{2,k}}^{(0)}$

```

end for
3.5 Build classifier  $\Psi_k^{(1)}$  on training set  $S_{N_{2,k}}^{(1)}$ 
end for

```

Fig. 1. Pseudocode of ensemble classifier with switching class labels.

4. CONCLUSIONS AND FUTURE WORK

The goal of this work was to propose ensemble method for classifying clients of service oriented systems, which solves two, described in introduction classification problems: imbalanced data and cost-sensitive classification problem. The goal was reached by proposing ensemble classifier with switching class labels, which uses switching probabilities to generate balanced datasets which are further used to build base classifiers on the first stage of the ensemble.

The future works concentrate on deeper evaluation of the proposal of ensemble classifier. The representative number of datasets is going to be selected and statistical test will be made to examine the difference between results achieved during testing the ensemble classifier with switching class labels and other approaches.

ACKNOWLEDGEMENT

The research presented in this work has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

REFERENCES

- [1] BATISTA G. E., PRATI R. C., MONARD M. C., *A study of the behaviour of several methods for balancing machine learning training data*, *ACM SIGKDD Explorations Newsletter*, 6, 20–29, 2004.
- [2] BISHOP C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.
- [3] BREIMAN L., *Bagging predictors*, *Machine Learning*, 24, 123–140, 1996.
- [4] BREIMAN L., *Randomizing Outputs to Increase Prediction Accuracy*, *Machine Learning*, 40, 229–242, 2000.
- [5] CENDROWSKA J., *Prism: An algorithm for inducing modular rules*, *International Journal of Man-Machine Studies*, 27, 349–370, 1987.
- [6] CHAWLA N. V., BOWYER K. W., HALL L. O., *SMOTE : Synthetic Minority Oversampling Technique*, *Artificial Intelligence*, 16, 2002.
- [7] CHAWLA N. V., LAZAREVIC A., BOWYER K. W., HALL L. O., *SMOTEBoost : Improving Prediction*. *Lecture Notes in Computer Science*, 2838, 107–119, 2003.

- [8] CHEN S., HE H., GARCIA E., *RAMOBoost: Ranked Minority Oversampling in Boosting*. IEEE transactions on neural networks, 21, 1624–42, 2010.
- [9] COHEN W. W., *Fast effective rule induction*. In Proceedings of the Twelfth International Conference on Machine Learning, 115–123, 1995.
- [10] DE SA M., *Pattern Recognition*, Springer, 2001.
- [11] FREUND Y., SCHAPIRE R. E., HILL M., *Experiments with a New Boosting Algorithm*, Machine Learning: Proceedings of the Thirteenth International Conference, 1996.
- [12] GARCIA E. A., *Learning from Imbalanced Data*, IEEE Transactions on Knowledge and Data Engineering, 21, 1263–1284, 2009.
- [13] GARCIA-LAENCINA P. J., SANCHEZ-GOMEZ J. L., FIGUEIRAS-VIDAL A. R., *Pattern classification with missing data: a review*, Neural Computing and Applications, 19, 263–282, 2009.
- [14] GHANNAD-REZAEI M., SOLTANIAN-ZADEH H., YING H., DONG M., *Selection-Fusion Approach for Classification of Datasets with Missing Values*, Pattern recognition, 43, 2340–2350, 2010.
- [15] GRZECH A. and et all (Editors). *SOA Infrastructure Tools – concepts and methods*, Poznan University of Economic Press, 2010.
- [16] GUO H., HERNA L. V., *Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach*, ACM SIGKDD Explorations Newsletter, 6, 30–39, 2004.
- [17] HUI H., WANG W., MAO B., *Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning*, Lecture Notes in Computer Science, 878– 887, 2005.
- [18] JO T., JOPKOWICZ N., *Class imbalances versus small disjuncts*, ACM SIGKDD Explorations Newsletter, 6, 40–49, 2004.
- [19] KUNCHEVA L. I., *Classifier Ensembles for Changing Environments*, Lecture Notes in Computer Science, 3077, 1–15, 2004.
- [20] KUNCHEVA L. I., *Combining Pattern Classifiers*, A John Wiley & Sons, Inc. Publication, 2004.
- [21] KURZYNSKI M., *Pattern recognition – statistic methods* (in Polish), Oficyna Wydawnicza Politechniki Wrocławskiej, 1997.
- [22] MANI J., ZHANG I., *KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction*, In Proceedings of International Conference on Machine Learning (ICML 2003), Workshop Learning from Imbalanced Data Sets, 2003.
- [23] MARTINEZ-MUNOZ G., SUAREZ A., *Switching class labels to generate classification ensembles*, Pattern Recognition, 38, 1483–1494, 2005.
- [24] MELVILLE P., MOONEY R. J., *Constructing Diverse Classifier Ensembles using Artificial Training Examples*, Proceedings of the IJCAI–2003, 505–510, 2003.
- [25] POLIKAR R., DEPASQUALE J., MOHAMMED H. S., BROWN G., KUNCHEVA L. I., *Learn++.MF: A random subspace approach for the missing feature problem*, Pattern Recognition, 1–16, 2010.
- [26] SUN Y., KAMEL M., WONG A., WANG Y., *Cost-sensitive boosting for classification of imbalanced data*. Pattern Recognition, 40, 3358–3378, 2007.

Bogumila HNATKOWSKA*, Bartosz NOWAKOWSKI*

WINDOWS AZURE CLOUD COMPUTING VERSUS CLASSICAL SOLUTIONS – COST COMPARISON

Cloud computing has become very popular recently. Windows Azure is one of the newest clouds available on the market. It became commercially available in 2010. Because of a very short history of cloud computing and even shorter of Windows Azure potential cost benefits of using them are not completely known. In the work a mathematical model to cost comparison of using a system deployed in Windows Azure and servicing by a company is presented. Based on this model several simulations were conducted in order to answer the question in which cases a cloud is economically efficient and to what extent.

1. INTRODUCTION

Recently, cloud computing services have become very popular. It is said that they can replace traditional servers in many applications regardless of the scale of these solutions [1].

There are many clouds available on the market. Clouds can be divided due to the type of services they offer or the availability of the services. Clouds can offer hardware resources with the entire infrastructure, the system supporting multiple servers running in the cloud or even software.

The main advantages of using clouds over traditional servers is a flexible payment model used in cloud computing [1, 2]. In that model a user pays only for the resources used.

The aim of the work is to compare the total cost of a system operation deployed in the Windows Azure and traditional servers. To do that a mathematical model was elaborated that was further used in several simulations.

* Wrocław University of Technology, Faculty of Informatics and Management, Institute of Informatics, Wyb. Wyspińskiego 27, 50-370 Wrocław, Poland.

The work is organized as follows. Section 2 provides a computation model to cost comparison, while Section 3 calculations done on the base of the model for a hypothetical system. Conclusions are presented in the last Section 4.

2. SIMULATION MODEL

2.1. COST FACTORS

Typically, the carrying a simulation involves following steps [3]:

1. Problem formulation.
2. Model construction.
3. Model validation.
4. Model usage.

The aim of the simulation is cost calculation and next cost comparison of a traditional solution, in which a whole software system (software, and hardware) is owned and managed by a specific company, and a modern solution, in which the system is deployed in a cloud. The assumption is that, in both cases, the performance of the system is the same in the considered period of time.

There are only a few works considering the above mentioned problem, e.g. [4, 5]. However, they usually describe the cost of cloud computing at very general level.

In our research we compare the cost of operation of a hypothetical but realistic system, assuming that it will be used for 3 years. Three years period is a typical one for which enterprise strategic plans are done. The test system is a website, due to the fact that this solution is the most common for both cases: cloud and traditional computing.

The built model is a typical symbolic, stochastic model, in which only one stochastically depended parameter exists – system load. Three different scenarios of changes in system load are considered: (a) rapid load increase (b) unpredictable load increases (c) predictable load increases, however in the work due to the limited space only two are presented. For the purposes of simulation, for each load characteristics, 1905 (365 days x 3 years) numbers have been generated, representing an average daily number of requests that the server had to process simultaneously. The numbers were generated with the use of GNU Octave environment [6], and the programs written by authors.

The rest of simulation parameters are deterministic, and easy to justify. All cost factors' values (see table 1 for Windows Azure, and table 2 for traditional servers) influencing the characteristics of the test system are determined in an arbitrary manner. However, these values are similar to typical websites.

Table 1. Cost factors for Windows Azure

Name (Symbol)	Value	Justification
Specification of hardware resources	Processor: 1,6 GHz RAM: 1,75 GB Hard disk: 225 GB	Among five server specifications provided by Microsoft [7] the <i>small</i> server has been selected. This is the default option in Windows Azure.
Number of requests which 1 server is able to serve simultaneously (RC_s)	20	The value is calculated based on traditional solution (see table 2). It is proportionally decreased taking into account the server configuration.
Administrator salary (μ)	4833 PLN	Average salary calculated on the base of salaries of 3 types of administrators: database administrator, net administrator, server administrator [8].
Number of servers (role instances) one administrator is able to manage (\mathcal{G}_a)	200	According to [9].
Number of data in GB stored in the system (D)	1000	Value set arbitrarily.
Average amount of data (in GB) that is sent after a single request by one user (ε)	402 KB = 0,000383 GB	Calculated as the average of 5 the most popular services (according to [10]) by opening 10 random pages. The calculation omits the commercials done in Flash technology.
Cost of services		
Computing power (C)	0,32 PLN for each server instance	According to [7].
Data storage (DSC) (φ)	0,40 PLN for 1 GB 0,03 PLN for 10000 transactions	According to [7].
Data transfer (γ) (σ)	Incoming: 0,27 PLN for 1 GB Outgoing: 0,40 PLN for 1 GB	

Table 2. Cost factors for traditional servers

Name (Symbol)	Value	Justification
Specification of hardware resources (π)	Processor: 2,13 GHz Hard disk: 2 × HDD 500GB RAM: 2 × 4096 MB Power supply: 500 W	The configuration recommended for Microsoft Office SharePoint Server 2007 [11]. An example server with such configuration is Actina Solar E 200 S4. The price of this server has been checked in sixteen stores. The average price is equal to 4750.62 PLN.
Cost of 1 server (K)	4750,62 PLN	
Rate of depreciation (τ)	30%	During the year the company could save thirty percent of the initial value of the hardware resources [12].
The number of requests that 1 server is able to serve simultaneously (RC_s)	60	The value is suggested for Microsoft Office SharePoint Server 2007 [11].
Administrator salary (μ)	4833PLN per month	Average salary calculated on the base of salaries of 3 types of administrators: database administrator, net administrator, server administrator [8].
The number of servers 1 administrator is able to manage (\mathcal{G}_a)	15	According to [9].
Average amount of data (in megabits) that is sent after a single request by one user (ε)	402 KB = 3,13 Mb	Calculated as the average of 5 the most popular services (according to [10]) by opening 10 random pages. The calculation omits the commercials done in Flash technology.
Electricity cost (φ)	258,81PLN for MWh	According to [9].
Microsoft Software license cost (K_w) (K_{sql})	Windows Server 2008: 1243 PLN SQL Server 2008 R2: 11598 PLN (with software assurance)	According to [13] assuming that the average course of dollar is given from [14] on 28.04.2011.
Cost of Internet connections (\mathcal{G})	30 PLN for each mb/s	

2.2. CALCULATION FORMULAS

2.2.1. WINDOWS AZURE

Total monthly cost (TC_{WA}) of the system operation in Windows Azure is calculated according to the formula given below.

$$TC_{WA} = C_{ComputationPower} + C_{DataStorage} + C_{DataTransfer} + C_{SystemManagement} \quad (1)$$

The components of the formula are defined below.

Monthly cost of computation power ($C_{ComputationPower}$) is defined as follows:

$$C_{ComputationPower} = \sum_M \text{ceil} \left(\frac{RC}{RC_S * \alpha} * \omega \right) * C \quad (2)$$

where:

- RC – number of simultaneous requests per day (generated by the program),
- α – performance increase factor (set to 0,8); when a new instance of a server is run the performance of the whole system increases on about 80%.
- M – set of daily loads in a given month (generated by the program),
- ω – margin of available computing power above current needs (set to 10 %); the margin allows to run new role instances without the decrease of performance.

The rest of symbols used in the formulas are explained in table 1.

The expression: $\text{ceil} \left(\frac{RC}{RC_S * \alpha} * \omega \right)$ is used for calculation of the necessary number of server instances.

Monthly cost of data storage ($C_{DataStorage}$) is calculated according to the formula (3).

$$C_{DataStorage} = D * DSC + \sum_M \frac{RC * \beta}{10000} * \varphi \quad (3)$$

where:

- β – number of transactions associated with one user request (generated by the program)

Monthly cost of data transfer ($C_{DataTransfer}$) is defined as follows:

$$C_{DataTransfer} = \sum_M RC * \gamma * \delta + RC * \varepsilon * \gamma \quad (4)$$

Monthly cost of system management ($C_{SystemManagement}$) is calculated as follows:

$$C_{SystemManagement} = \text{round} \left(\frac{\text{ceil} \left(\frac{RC_m}{RC_S * \alpha} \right) * \omega}{\vartheta_a} \right) * \mu \quad (5)$$

where:

- RC_m – maximum number of simultaneous requests in a given month.

2.2.2. TRADITIONAL SOLUTION

Total monthly cost (TC_{TS}) of the system operation in traditional environment is calculated according to the formula given below.

$$TC_{TS} = C_{Hardware} + C_{Electricity} + C_{InternetConnections} + C_{SystemManagement} \quad (6)$$

In addition to the total cost of the system running on traditional servers (sum of TC_{TS} calculated for 3 years), we need to add the cost of server software licenses – $C_{Licences}$ (see formula 7).

$$C_{Licences} = \text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right) * K_w + (\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right) / 10) * K_{sql} \quad (7)$$

Monthly cost of hardware ($C_{Hardware}$) is calculated according to formula (8). Explanation of the symbols used in formulas is given in table 2. The meaning of the rest of symbols, e.g. α , is the same as for Windows Azure.

$$C_{Hardware} = (\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right) + \text{ceil}\left(\frac{\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right)}{10}\right)) * K * \frac{\tau}{12} \quad (8)$$

The assumption is that a company has all servers needed. Only rate of depreciation is taken into account. For each 10 transaction servers one database server is addend. Function ceil is used to include redundant servers used in the case of failure of one of the primary servers.

Monthly cost of electricity ($C_{Electricity}$) used by servers is calculated according to the formula given below.

$$C_{Electricity} = (\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right) + \text{ceil}\left(\frac{\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right)}{10}\right)) * \pi * 24 * 30 * \varphi \quad (9)$$

Monthly cost of Internet Connections ($C_{InternetConnections}$) is defined as follows:

$$C_{InternetConnections} = \frac{\varepsilon * RC_m}{\sigma} * \vartheta \quad (10)$$

where:

- σ – maximal number of seconds to wait for a response (set to 3 s).

Monthly cost of system management ($C_{SystemManagement}$) is defined below.

$$C_{SystemManagement} = \text{round}\left(\frac{(\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right) + \text{ceil}\left(\frac{\text{ceil}\left(\frac{RC_m}{RC_S * \alpha}\right)}{10}\right))}{\vartheta_a}\right) * \mu \quad (11)$$

The model was built by one co-author and validated by the second one. The only element needed examination was the logic of the formulas, and functions generating

data. The output validity was not checked as we had no real system with the known costs of operation.

3. SIMULATIONS

3.1. RAPID INCREASE OF SYSTEM LOAD

Scenario of rapid increase assumes continued growth in system load throughout the time. According to the generated data the load of the system begins with thirty requests per second. In the three years system load increases about seven times to about two hundred and twenty requests per second.

3.1.1. WINDOWS AZURE

Monthly costs of system operation throughout the test period ranged is shown in Fig. 1. All obtained calculations and results visualizations were done in Excel.

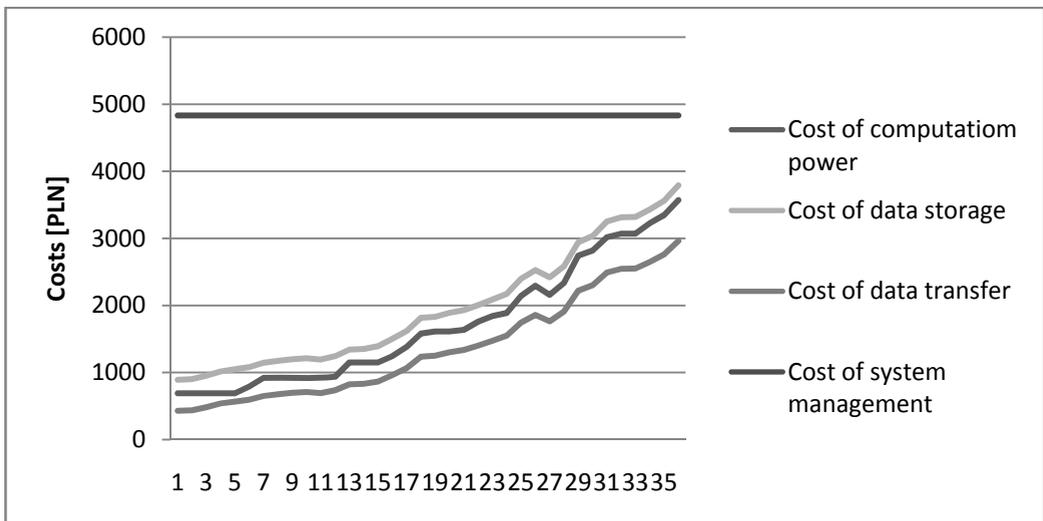


Fig. 1. Monthly costs of system operation in Windows Azure – rapid increase scenario

Costs of system management are constant. This is due to the assumption that there is a necessity of at least one administrator employed for full time. The rest of cost components strongly depend on system load. The most expensive is to store the data in the cloud.

3.1.2. TRADITIONAL SERVERS

Monthly cost of system maintenance of traditional servers is constant throughout the test period and is equal to about 14 000 PLN. The amount does not include the cost of air conditioning, which may vary depending on the type or location of the building in which servers are running.

Additionally, costs of software licenses must be taken into account. The cost of the licenses is about 21 500 PLN (licenses for operating system for all servers and one SQL Server license).

3.1.3. COMPARISON FOR RAPID INCREASE OF SYSTEM LOAD SCENARIO

The total costs of system operation in the case of Windows Azure (355 289 PLN) is approximately 34% percent lower than for traditional servers (540 596 PLN). However, monthly costs in the case of Windows Azure in the final stage exceeded the cost of traditional servers what may suggest that the solution using traditional servers could be more profitable if the load level such as in the final stage of the period, will keep for a long time.

3.2. UNPREDICTED INCREASE OF SYSTEM LOAD

Fig. 2 presents data generated by the program, representing unpredicted increase of system load. Twenty increases have occurred, lasting ten days. The load in the period between increases remained stable with slight fluctuations.

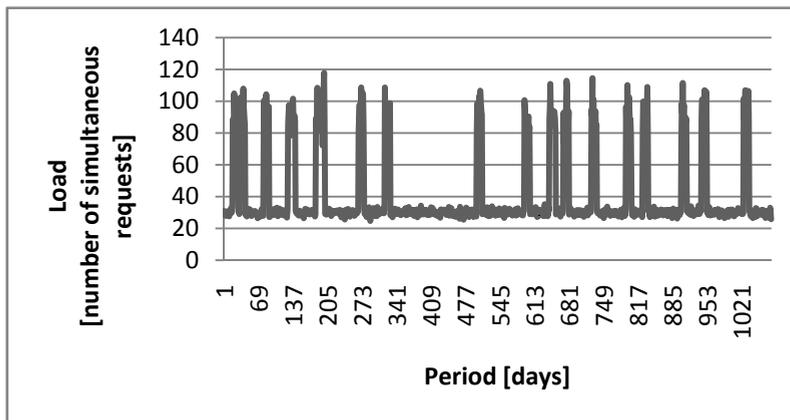


Fig. 2. Presumed number of simultaneous requests for unpredicted increase scenario

3.2.1. WINDOWS AZURE

Monthly costs of system operation throughout the test period ranged from about 6000 PLN to 7500 PLN. The cost of cloud computing services ranged from about 1400 PLN to 2500 PLN per month (see Fig. 3).

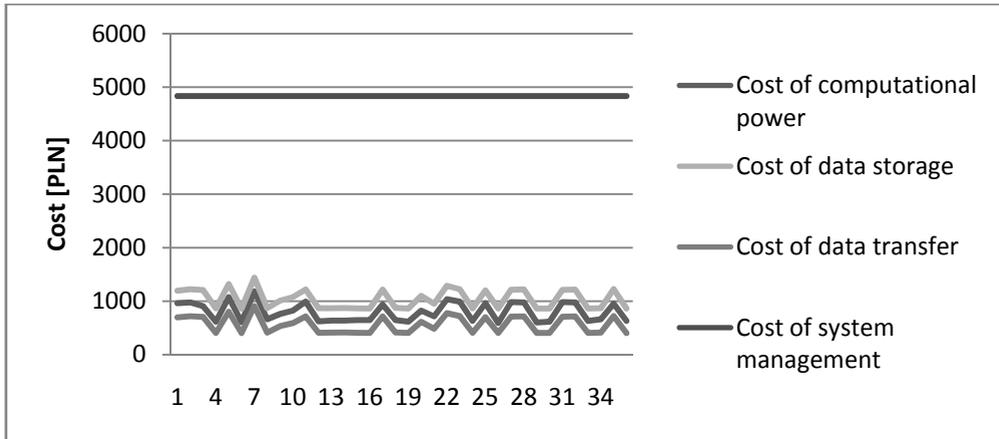


Fig. 4. Monthly costs of system operation in Windows Azure – unpredicted increase scenario

3.2.2. TRADITIONAL SOLUTIONS

The costs of system operation includes the cost of electricity, calls and system management. These costs are the same (about 10000 PLN) throughout the test period. Therefore, the total cost of system operation is also constant. The biggest part of this amount is paid to administrator. Another big cost component is the cost of Internet connections. Much less are the monthly costs of hardware maintenance and power costs. They do not exceed 1000 per month.

3.2.3. COMPARISON FOR UNPREDICTED INCREASE OF SYSTEM LOAD SCENARIO

In the case of unpredicted increases of load distribution the differences in monthly costs of system operation are significant. They range from about 3000 to 4000 PLN month (see Fig. 5).

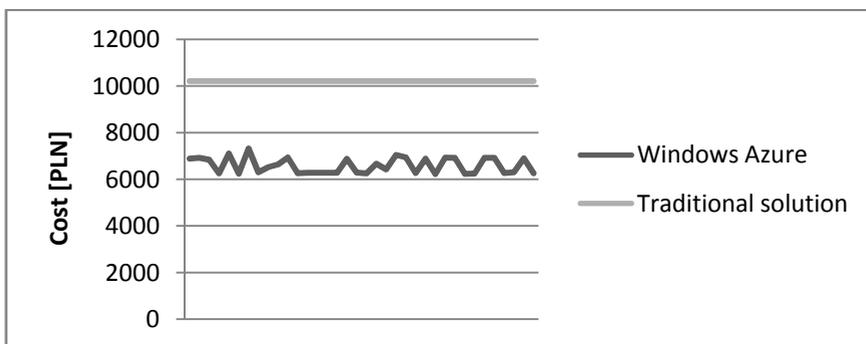


Fig. 5. Comparison of monthly costs of system operation – unpredicted increase scenario

Despite the jumps, during which system load was even three times bigger than the normal load, the monthly cost of service Windows Azure increased up to twofold. This is due to small segments of time in which these increases occur.

4. CONCLUSIONS

The work examines the costs of system operation for two types of solutions, based on the Windows Azure cloud and traditional servers. The simulation was conducted based on previously defined mathematical model.

The results of the simulations in all test cases suggest that the use of Windows Azure is more profitable than the use of traditional servers. Savings achieved through the use of Windows Azure is about 35% compared to the costs of using traditional servers.

In all considered scenarios of system load change the total cost of system operation is lower for Windows Azure throughout the whole period considered. However, for rapid increase scenario, the cost of the system deployed in Windows Azure is growing throughout the whole period considered, finally exceeding the operation costs for the traditional servers. Operation cost in Windows Azure exceeds the cost of operation for traditional servers when the average system load approaches the maximum permissible load of traditional servers. The main reason for lower costs in the case of Windows Azure is the ability to dynamically acquire and release hardware resources. This suggests that Windows Azure is a cheaper solution in all cases where the load is not stable or not close to the maximum permissible load of the servers.

The calculations presented in the work were done for only one dataset representing system load. The calculations were repeated for another sets, but the results obtained at the beginning were representative ones.

The simulations were carried out on the basis of values determined for a hypothetical system. However, the model used in simulation can be used to estimate the costs of maintenance of other systems.

We are going to check the accuracy of the model for a real system. The model could be also extended to include parameters of other clouds than Windows Azure.

REFERENCES

- [1] BOCEWICZ G., BANASZAK Z., WÓJCIK R., *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach*, In: Agent and Multi-Agent Systems: Technologies and Applications, Nguyen N.T., Grzech A., Howlett R.J., Jain L.C. (Eds.), Berlin, Springer-Verlag, 2007, 578–587.
- [2] NOWACKI W., *Plasticity of polycrystal*, Warszawa, PWN, 1987, 687–704.

- [3] PERROS H., *Computer simulation techniques. The definitive introduction*. Computer Science Department, NC State University, Raleigh, NC, 2009, available at: <http://www4.ncsu.edu/~hp/simulation.pdf>
- [4] ASHIMOV A., BOROVSKIY Yu., ASHIMOV As., *Parametrical Regulation Methods of the Market Economy Mechanisms*, Systems Science, Vol. 35, No. 1, 2005, 89–103.
- [5] QIAN L., ZHIGUO L., YUJIAN D., LEITAO G. *Cloud Computing: An Overview.*, In: Cloud Computing: First International Conference, CloudCom 2009. Beijing, Springer-Verlag, 2009.
- [6] PETCU D., *Identifying Cloud computing usage patterns*. IEEE Xplore. Research Institute e-Austria Timisoara, Computer Science Department, West University of Timisoara, 2010. <http://ieeexplore.ieee.org/iel5/5606060/5613076/05613106.pdf> (march 2011).
- [7] LI X., LI Y., TIANCHENG L., JIE Q., FENGCHUN W. *The Method and Tool of Cost Analysis for Cloud Computing*. Bangalore: Res. Lab., IBM, 2009.
- [8] WALKER, E. *The Real Cost of a CPU Hour*. IEEE Computer, 2009, 35-41.
- [9] EATON J. W. *Octave*. <http://www.gnu.org/software/octave/> (May 2011).
- [10] Microsoft Corporation. *Windows Azure Platform Consumption*. Windows Azure Platform Consumption, <http://www.microsoft.com/windowsazure/offers/popup/popup.aspx?lang=en&locale=en-us&offer=MS-AZR-0003P> (May 2011).
- [11] SEDLAK & SEDLAK. *Moja placa*. http://www.wynagrodzenia.pl/moja_placa.php (April 2011).
- [12] ALINEAN Inc. *Microsoft Windows Azure Platform TCO Calculator.*, <https://roianalyst.alinean.com/msft/AutoLogin.doc?d=176318219048082115> (October 2010).
- [13] ALEXA INTERNET Inc. *Alexa Top 500 Global Sites*. <http://www.alexa.com/topsites> (April 2011).
- [14] MICROSOFT CORPORATION. *Estimate performance and capacity requirements for search environments*. Microsoft Corporation. Maj 21, 2009. [http://technet.microsoft.com/en-us/library/cc262574\(office.12\).aspx](http://technet.microsoft.com/en-us/library/cc262574(office.12).aspx) (May 2011).
- [15] Dziennik Ustaw, Nr 14, Poz. 176, Ustawa z dnia 26 lipca 1991 r. o podatku dochodowym od osób fizycznych, 2000.
- [16] MICROSOFT CORPORATION. *SQL Server 2008 R2 Pricing*. <http://www.microsoft.com/sqlserver/2008/en/us/pricing.aspx> (April 2011).
- [17] Narodowy Bank Polski. *Kursy walut*. Narodowy Bank Polski. Kwiecień 16, 2011. <http://www.nbp.pl/home.aspx?f=/kursy/kursya.html> (April 2011).

PART III

**MODELING AND MEASURING QUALITY
OF KNOWLEDGE AND SERVICES**

Leonardo MANCILLA AMAYA*,
Cesar SANIN*, Edward SZCZERBICKI*

AN AGENT-BASED APPROACH TO MEASURE KNOWLEDGE QUALITY

The increasing importance of knowledge as an organizational asset has encouraged researchers to explore new and diverse mechanisms to measure quality of knowledge. The high complexity of this topic and relatively scarce literature in this area, present a new opportunity for further development and improvement. Some existing solutions use elements from Total Quality Management or criteria from data and information quality, with the aim of assessing knowledge quality. In general, the approaches presented in literature require a high degree of human intervention; thus, the workload is increased along with the risk of biased opinions influencing the final quality measures. This chapter presents the design and implementation of a new agent-based approach to measure knowledge quality. User feedback and automated agent calculations are taken into account to obtain a percentage, which represents an individual's knowledge quality. This approach is used by the e-Decisional Community, an integrated knowledge sharing platform that aims at the creation of markets where knowledge is provided as a service.

1. INTRODUCTION

Quality has been traditionally regarded as an important attribute of manufactured goods, and it has become a key factor to assure an organization's success in global economy. However, literature has failed to provide a unified definition of quality, and probably it is impossible to reach an agreement on this issue, because quality can be understood depending on the context in which it is used. For instance, Seawright and Young [12] present a variety of definitions about quality and the relationships between them, classified in seven categories as follows: strategic, transcendent, multidimensional, manufacturing-based, value-based, product-based and user-based; according to Seawright and Young [12], the understanding of these associations can help an or-

* School of Engineering, Faculty of Engineering and Built environment, The University of Newcastle, Callaghan, NSW, 2308, Australia.

ganization compete in a better way. Reeves and Bednar [12] describe the advantages and disadvantages of different definitions of quality, and state that each one is appropriate under different situations. Some definitions of quality include the concepts of quality as excellence, as value, as conforming to specifications, as a way to meet/exceed expectations, and quality from the customer's point of view. These definitions of quality are related also to the categories defined in [12], but still, a unique definition of quality seems to be hard to find.

Given the nature of today's markets, intangible goods and services are also subject to quality assessment, mainly because consumers have become more demanding and knowledgeable. Knowledge is probably the most valuable intangible asset for an organization, which helps them meet new market demands. Managers face decision-making situations on daily-basis and they have to use their previous experience to support their current decisions. This process involves assessing the quality of existing solutions to maximize the benefits for the organization.

It can be understood that measuring knowledge quality is a new kind of task that is required in order to increase the effectiveness of decisions and overall competitive advantage. This topic has attracted the attention of the research community given its complexity and the lack of consensus around related concepts such as quality. Different approaches have been developed aimed at measuring knowledge capital in organizations, knowledge held by individuals or embedded in processes [1, 5, 7]. Nonetheless, more clarity on the specifics of how to assess quality and quantity of knowledge is required. Steedman [13] questions different proposals on knowledge measurement from the perspective of economics arguing that the "stock of knowledge" might not be cardinally measured, and only when theory produces clear indicators it will be possible to identify magnitudes and measure knowledge accurately [13]. Also, Hofer-Alfeis [4] presents an approach to measure knowledge in organizations, but makes it clear that the measurements for quality and value of knowledge is an issue that is not solved in his approach.

In this chapter, a different approach to measure explicit knowledge quality is presented. This proposal aims at creating a semi-automatic way of assessing quality, and increase the effectiveness of the decisions made by organizations. The proposed approach looks at quality from the value point of view; value-based quality is an extension of user-based quality, in which a product satisfies users' needs as defined in [12]. Knowledge and experience are assets that provide a company with the means to adapt, and this can be understood as providing added value. The concepts hereby presented are applied in the e-Decisional Community [9], an agent-based platform for sharing experiential knowledge across different organizational levels. This platform uses a standard knowledge representation called Set of Experience Knowledge Structure (SOEKS), which constitutes Decisional DNA (DDNA) [11]. Decisional DNA is proposed as a unique and single structure for capturing, storing, improving and reusing decisional experience. Its name is a metaphor related to human DNA and the way it

transmits genetic information among individuals through time. The following sections present the conceptual background that supports the proposed approach and details of the knowledge quality assessment mechanism. The experimental prototype is also included for illustrative purposes. Finally, a conclusion and future work are presented.

2. PROPOSED QUALITY MEASUREMENT MECHANISM

2.1. KNOWLEDGE QUALITY ATTRIBUTES

Knowledge quality measurement in the e-Decisional Community is based on a set of nine attributes, extracted from existing literature on data and information quality. The main reason to base the proposal presented in this section on data and information, is that they play an important role in the creation of knowledge, as described in [2]. Also, since the process of quality assessment is meant to be semi-automatic, the best approach is to define a set of items that the agents participating in the e-Decisional Community can measure.

The proposed quality attributes were selected based on their number of appearances in literature. This indicates that there is a consensus about role in quality measurement. The selection process was based on a Pareto analysis. After obtaining the preliminary attributes, an individual analysis of each one of them was performed to refine the list, and determine the viability of their implementation in a knowledge-oriented context using Decisional DNA and SOEKS. The amount of knowledge was included as a key element in this proposal for a main reason: measuring the amount of knowledge in the e-Decisional Community will allow the platform to provide an estimate on the depth of an agent's knowledge. Research on how to measure quantity of knowledge is an ongoing task of the Knowledge Engineering Research Team (KERT), at The University of Newcastle, Australia. Table 1 presents the final attributes that were selected after the depuration process, along with their adapted definitions from literature.

Table 1. Proposed Knowledge Quality Attributes

Indicator	Definition
Accuracy	Degree of closeness of its value v to some value v' , considered correct for an entity and an attribute. (Sometimes v' is referred to as the standard.) [3].
Timeliness	The extent to which the knowledge is up-to-date for the task at hand [10].
Completeness	Knowledge is sufficient and not missing in order to complete a task [10].
Relevance	Relevance is concerned with whether acquired knowledge can be applied in a user's task [6].
Understandability	The level of expressiveness that allows for the meaning of knowledge to be understood easily [6].
Reputation	Knowledge highly regarded in terms of its source or content [10].
Believability	The extent to which knowledge is regarded as true or credible [3].

Objectivity	Knowledge is unbiased [10].
Amount	-The level of appropriateness for quantity of provided knowledge to be used in current affairs [6]. -The extend to which the volume of knowledge is appropriate for the task at hand [10].

2.2. OBTAINING VALUES FOR THE ATTRIBUTES

Attributes were grouped depending on how their values can be obtained; three different ways were identified: user, agent, or the Smart Knowledge Management System – SKMS [11]. All the values have a range from zero to one ($[0, 1]$). The values that can be automatically calculated by the agents and the SKMS contribute to the automation of the measurement process, and the user category contains the attributes that obtained from user feedback. This approach allows the system to receive inputs from the real world and adjust its behavior accordingly. The grouping and additional remarks about each attribute are as follows:

- User Category (Values obtained from user feedback): Timeliness, Relevance, Objectivity, and Understandability.
- Agent Category (Values automatically calculated by agents): Amount, Completeness and Reputation. The latter is based on the result of previous knowledge interactions, as presented in [8].
- SKMS Category (Values extracted from the SOEKS): Believability and Accuracy. These attributes are mapped to the truth and precision values of the SOEKS respectively. Their initial values are defined by the Prognosis Macro-Process. See [11] for more information.

2.3. MODEL FOR KNOWLEDGE QUALITY ASSESSMENT

First of all, it is important to recall that the knowledge measurements hereby presented are approximations of the actual quality of knowledge held by individuals and agents. Based on the elements presented throughout this section, quality inside the e-Decisional Community is measured in two steps: firstly, the quality of each individual experience belonging to an agent is calculated. Secondly, the quality of all the experiences of an entity is calculated.

Quality for individual SOEKS is defined as the average of all the quality attributes values. All attributes have the same weight because it is considered that an agent's expertise is as important as user feedback. Reputation is used in the final stage of the process, which is described in the next section. Let us define:

$$A = \{a_1, a_2, \dots, a_n\}: \text{The set of agents in the system, where } n \text{ is the total number of agents.} \quad (1)$$

$$S(a_i) = \{S(a_i)_1, S(a_i)_2, \dots, S(a_i)_m\}: \text{The set of SOEKS of agent } a_i \in A, \\ \text{where } m \geq 0 \text{ is the total number of SOEKS for that agent.} \quad (2)$$

$$QA(S(a_i)_j) = \left\{ \begin{array}{l} QA_{accuracy}, QA_{timeliness}, QA_{complete}, QA_{relevance}, QA_{understand}, \\ QA_{believe}, QA_{objectivity}, QA_{amount} \end{array} \right\}: \\ \text{The set of quality attributes} \quad (3)$$

Therefore, the quality measure Q of an individual SOEKS in the e-Decisional Community is defined as:

$$Q(S(a_i)_j) = \frac{1}{8} \sum_{k=1}^8 QA(S(a_i)_j)_k ; 1 \leq k \leq 8 \quad (4)$$

The values of quality measures for individual SOEKS can be distributed in several ways; there is not a standard model that can be used to predict a specific behavior in the values. Therefore, overall quality calculations are performed using regression analysis, offering the possibility to discover the equation that best fits a set of data samples (i.e. individual quality measures). Consequently, the total quality can be understood as the area under the best fitting curve: as the area increases so does the final quality. The overall quality for an agent in the system, $Q_{Overall}(a_i)$, is obtained by integrating the best fit equation, as follows:

$$Q_{Overall}(a_i) = \int_1^n fit(x) \cdot dx \quad (5)$$

Where: n is the total number of individual SOEKS quality measures, and $fit(x)$ is the best fit equation for the data samples. Since there is not a standard unit of measurement for knowledge, the final value is given as a percentage with respect to the possible maximum area under the curve. For example, let us assume that an agent has a total of fifty experiences in its knowledge repository. In this scenario, the agent is a “guru” and the individual knowledge quality for each one of its SOEKS is 1, which results in a best fit equation in the form $y=mx+b$, with $m=0$ and $b=1$. Consequently, the area under the curve is given by the area of the rectangle with length=50 (the total number of samples) and width=1; as a result, we have an area of 50, which is equivalent to a 100% knowledge quality.

The overall quality measure is used by agents in the e-Decisional Community as a way to evaluate other peers at the time of conforming Knowledge-Based Virtual Organizations [8]. When engaging in cooperative tasks, the definitive quality measure is calculated using an agent’s reputation $R(a_i)$ [8]. This approach helps in the process of

determining which entities are more likely to deliver a good result, and also as a “tie-breaker” when there are several agents with similar $Q_{Overall}$ values. Therefore, the quality measure used to decide an agent’s suitability for cooperation is given by the following formula:

$$Q_{Total}(a_i) = R(a_i) \cdot Q_{Overall}(a_i) \tag{6}$$

3. EXPERIMENTAL PROTOTYPE

A prototype implementing the functionality described in the previous section was developed, with the intention of performing several independent trials to evaluate how quality measures influence the possibility of an agent being selected for cooperation. The agent system is comprised of ten agents; one agent sends a group creation request, the system evaluates it, calculates overall quality values, and returns a list of the “most knowledgeable” agents. The experiment is repeated one hundred times. Quality measures for individual SOEKS were generated using a random number generator, and are recalculated to simulate the process of user feedback through time and its effects on quality. Each agent has two hundred SOEKS in their respective repositories.

3.1. PROTOTYPE IMPLEMENTATION

The prototype was implemented using Java 6, JADE 4.0.1 [15], Symja 0.0.7a [14] and Statcato 0.9.2 [16]. The classes that contain the proposed functionality are `QualityAttributes`, `QualityAttributesCache` and `QualityFileManager`. The features for quality measurement were implemented in the Individual Management Layer (IML) defined for the e-Decisional Community [9]. Figure 1 presents the simplified class diagram for IML package.

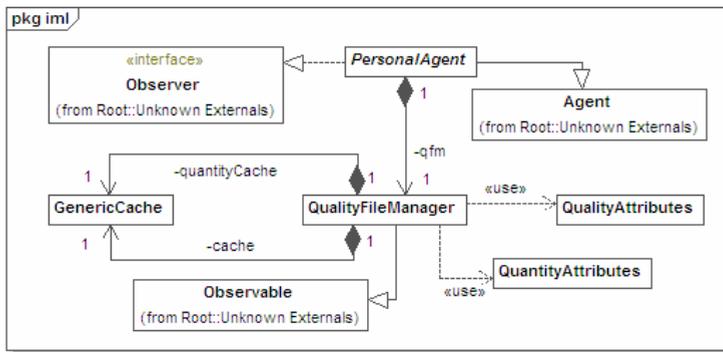


Fig. 1. Simplified class diagram for the experimental prototype.

The `QualityAttributes` holds the quality attributes for an individual SOEKS, and it is also in charge of calculating the average quality according to equation 4. The `QualityFileManager` administers the Master Quality File (MQF) for each agent; the MQF is a file that stores the information about individual quality measures for several SOEKS, which are provided by the `QualityAttributes` class. The MQF has as many entries as SOEKS exist for an agent, and each agent has one observable MQF. The `QualityAttributesCache` is an exact copy of the MQF in memory, in order to increase the performance of the system.

The prototype relies on a global Knowledge Quality Service (KQS). This service is an agent that is in charge of: *i*) Performing the regression analysis; *ii*) Calculating $Q_{Overall}$ as described in equation 5; *iii*) Keeping record of the quality for all agents; and *iv*) Providing information for group formation processes.

When an agent enters the system, it sends a message to the KQS asking to be registered. Following this, the KQS performs all the required calculations and creates a new entry in its registry. When an agent's MQF is modified, it sends a message to the KQS, which will recalculate quality based on the new values and overwrite the existing registry entry. Entries in the KQS are indexed by the agent's name, area, and subarea of knowledge. Therefore, an agent may have many entries in the KQS registry under its name, but each one of them will belong to the quality of knowledge in different topics.

In order to perform the regression analysis, some classes from the Statcato [16] project were used. These classes are: `BasicStatistics`, `CorrelationRegression`, `MultipleRegression2`, and `HelperFunctions`. As a result, the prototype is able to support seven different types of regression: linear, quadratic, cubic, logarithmic, power, exponential, and polynomial. The polynomial regression is calculated up to $n-1$ degrees, with n being the total number of samples. Although it is considered that a seventh degree polynomial is sufficient in most cases, in this prototype's scenario, it is desired that at least the polynomial model provides a high fit when all others have failed to do so. This approach assures a higher precision in the assessment of quality with a large number of data samples.

When the KQS executes the regression, it evaluates all the previously mentioned possibilities and uses the coefficient of determination R^2 to choose the best fit. Once the best fit is chosen, the following step is to calculate the area under the curve. This is achieved by using the Symja [14] library. In this process, the lower bound for the integral will always be 1 (at point 0 no experience is assumed), and the upper bound is given by the number of data samples that are provided by an agent. Then, simply by using the command `Integrate`, the KQS is able to determine the overall quality of knowledge.

Each time an agent requests the creation of dynamic knowledge-based virtual organization, the first message is sent to the KQS, which returns a list of the highest ranked agents. Then, the initiator agent queries the reputation service and obtains the

reputation values for the candidates. Then, Q_{Total} is calculated for each nominee as described in equation 6. With these values, the initiator sends messages to all the selected agents to initiate the cooperative work.

3.2. EXPERIMENTAL RESULTS

Figure 2 shows the experimental results obtained after measuring the number of times agents are selected as part of a knowledge-based virtual organization, based only on the quality of their knowledge Q_{Total} . Because of the random values generated between trials, all the agents were selected the same number of times on average. This situation helps illustrate a scenario where all of the organizational members are knowledgeable in a particular topic, and can contribute and learn as equals reducing the dependency on “expert” agents.

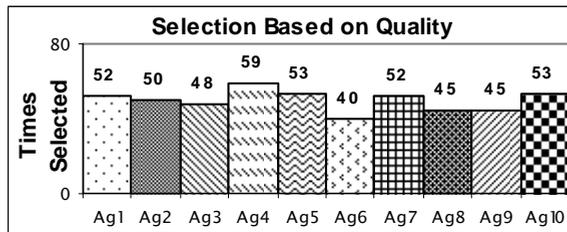


Fig. 2. Selection based on quality.

Figures 3 and 4 present a detailed analysis of the quality assessment process for agents 4 and 6. These agents were selected because they represent opposite cases in the selection process described by previously. The graphics show the relationship between overall knowledge, reputation and the total quality Q_{Total} for each one of the aforementioned agents.

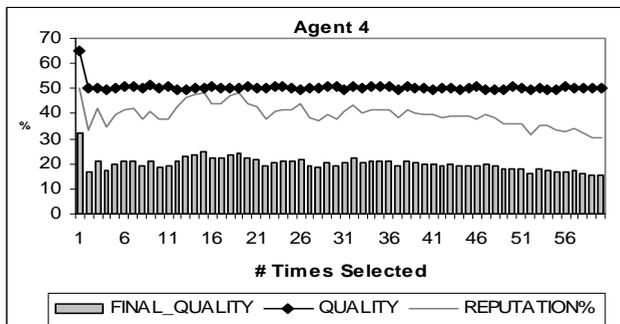


Fig. 3. Effect of reputation over quality for Agent 4.

During the experiments it can be observed that both agents have individual quality values around 50% for their SOEKS. Reputation for agent 6 had values between 30% and 40%, while agent 4 maintained a reputation around 40% for the majority of the iterations. Reputation has a great impact on the final quality values, Q_{Total} , and it is an effective method for dealing with duality in results, and an adequate way of making informed decisions about agents in the community. Also, the use of reputation enhances the platform with a basic ability to resemble social human behavior in group environments. People usually ask for advices from others who can provide new knowledge and are highly regarded inside a group.

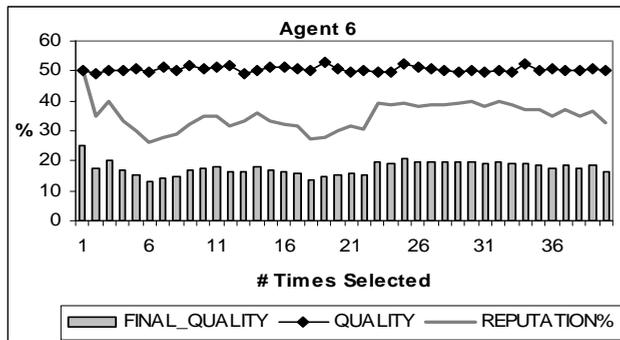


Fig. 4. Effect of reputation over quality for Agent 6.

Reputation values for agent 6 illustrate a condition that can be encountered in real life, and that is misinformation. This scenario is illustrated by Hunt [5], who states that people can strongly believe that they are correct even when they are not, and might use erroneous beliefs to make decisions. Hunt [5] makes the following remark: “A *sure-but-wrong belief, used confidently as a basis for making decisions and taking actions, may lead to surprising errors in performance – sometimes with tragic results*”. In the e-Decisional Community, this means that a user gives a high score to its SOEKS’ quality attributes based on a mistaken personal conviction; therefore, an agent will have high SOEKS quality and will probably be selected for cooperation. After several interactions other agents/users might realize that the knowledge provided by the misinformed entity is incorrect. Consequently, trust levels towards that agent must be adjusted, affecting its global reputation and reducing the number of times it is selected for cooperation, as illustrated by the experiments.

4. CONCLUSION

The model presented in this chapter, represents a new effort towards providing more accuracy in the process of quality measurement. The assessments obtained from

this evaluation, help users and agents increase the effectiveness of decision making processes. Also, quality measures in the e-Decisional Community can be used as a way to enforce Service Level Agreements, or as a way for organizations to evaluate possible peers. However, the proposed technique has not been evaluated in a real life operational context, and feedback from different industrial and academic sectors will help in the process of refining the proposed approach. Future work includes the development of historical measures, as a way to keep track of an agent's knowledge quality over time. As well, the development of a metric to quantify the knowledge possessed by an agent needs to be implemented.

REFERENCES

- [1] BONTIS, N., *Assessing knowledge assets: a review of the models used to measure intellectual capital*, International Journal of Management Reviews, Vol. 3, No. 1, 2001, 41–60.
- [2] DAVENPORT, T. H., PRUSAK, L., *Working knowledge: How organizations manage what they know*, Harvard Business Press, Boston, MA, 1998.
- [3] FOX, C., LEVITIN, A., THOMAS, R., *The notion of data and its quality dimensions*, Inf. Process. Manage., Vol. 30, No. 1, 1994, 9–19.
- [4] HOFER-ALFEIS, J., *Measuring Knowledge and KM in an organization with an explicit top-down Knowledge Strategy*, in U. Reimer, A. Abecker, S. Staab and G. Stumme, eds., WM2003: Professionelles Wissensmanagement – Erfahrungen und Visionen, Gesellschaft für Informatik, Lucerne, Switzerland, 2003, 433–442.
- [5] HUNT, D. P., *The concept of knowledge and how to measure it*, Journal of intellectual capital, Vol. 4, No. 1, 2003, 100–113.
- [6] LEE, J., LEE, Y., RYU, Y., KANG, T. H., *Information Quality Drivers of KMS*, Convergence Information Technology, 2007. International Conference on, 2007, 1494–1499.
- [7] LIST, B., SCHIEFER, J., BRUCKNER, R. M., *Measuring knowledge with workflow management systems*, Proceedings. 12th International Workshop on Database and Expert Systems Applications, 2001., 2001, 467–471.
- [8] MANCILLA-AMAYA, L., SANÍN, C., SZCZERBICKI, E., *Knowledge-Based Virtual Organizations for the E-Decisional Community*, in R. Setchi, I. Jordanov, R. Howlett and L. Jain, eds., *Knowledge-Based and Intelligent Information and Engineering Systems*, Springer Berlin / Heidelberg, 2010, 553–562.
- [9] MANCILLA-AMAYA, L., SANÍN, C., SZCZERBICKI, E., *Smart Knowledge-Sharing Platform For E-Decisional Community*, Cybernetics and Systems: An International Journal, Vol. 41, No. 1, 2010, 17–30.
- [10] PIPINO, L. L., LEE, Y. W., WANG, R. Y., *Data quality assessment*, Commun. ACM, Vol. 45, No. 4, 2002, 211–218.
- [11] SANIN, C., SZCZERBICKI, E., *Decisional DNA and the Smart Knowledge Management System: A process of transforming information into knowledge*, in A. Gunasekaran, ed., *Techniques and Tools for the Design and Implementation of Enterprise Information Systems*, IGI, New York, 2008, 149–175.

- [12] SEAWRIGHT, K. W., YOUNG, S. T., *A Quality Definition Continuum*, Interfaces, Vol. 26, No. 3, 1996, 107–113.
- [13] STEEDMAN, I., *On 'Measuring' Knowledge in New (Endogenous) Growth Theory*, Old and New Growth Theories: An Assessment, 2003, 127–133.
- [14] SYMJA, *symja: A Java computer algebra system* Available at: <http://code.google.com/p/symja/>, 2011, Last Access: April 5 2011
- [15] TELECOM-ITALIA, *Java Agent DEvelopment Framework*. Available at: <http://jade.tilab.com/>, 2011, Last Access: April 4 2011
- [16] YAU, M., *Statcato: Free Software for Elementary Statistics* Available at: <http://www.statcato.org/>, 2011, Last Access: April 5 2011

Piotr CZAPIEWSKI*

STRATEGY FOR A ROBUST AGGREGATION AGENT IN A MULTI-AGENT AUTOMATED TRADING ENVIRONMENT

The work concerns the problem of building a multi-agent automated trading environment, where multiple agent types will participate in many aspects of decision support for trading financial instruments (e.g. stocks, stock indices, currency pairs, CFDs). A problem of decision aggregation occurs, when several decision support agents are to be used simultaneously, while only one decision recommendation should be given by the system and pursued by a trader (or by an automated trade execution agent). Rationale for aggregating multiple decision support agents is discussed in the work, as well as the requirements that must be met by agents in order to apply the aggregation, and the general means of performing such a procedure. Given the distributed and heterogeneous characteristics of an automated trading environment, dealing properly with communication failures, missing data, and other unexpected behaviour, is a crucial issue. Therefore a robust strategy is proposed in the work for performing the aggregation in the presence of uncertainty and in case of unreliability of communication medium or instability of participating agents.

1. MULTI-AGENT AUTOMATED TRADING ENVIRONMENT

Automated trading is a discipline concerned with performing computerized transactions on certain financial markets (usually stocks, stock indices, currency pairs or CFDs), where all aspects of trading activities are supported by software, with some level of intelligence and autonomy. These aspects include real-time monitoring of market data, on-line data acquisition (quotes data, market-related news, transaction signals), data analysis and making decision regarding transactions, and finally performing transactions without human supervision.

* West Pomeranian University of Technology, Faculty of Computer Science and Information Technology, Żołnierska 49, 71-210 Szczecin, Poland.

The software aimed at supporting all or some of the above aspects meets the definition of an intelligent software agent, or at least the most important and generally agreed on aspects of agents characteristics [5, 6, 9]:

- it acts within a particular environment (here – online market), monitoring it and interacting with it, communicating with other entities (other trading agents, brokerage companies, market signal providers, news agencies, etc.) and responding to perceived events (incoming quote data, market signals, market reports, market-related news, etc.);
- it is autonomous – given a certain set of guidelines, it acts further on without requiring human participation;
- it is proactive and acts on behalf of its user (in this case – on behalf of the trader), pursuing to accomplish a predefined goal (in this case – to gain profit from transactions).

Many ready to use software applications exist, that can support the trader and perform automated trading (e.g. TradeStation, MetaStock, MetaTrader, VTtrader). All these products, as far as their automated trading capabilities are concerned, fall into one category, that could be characterized as follows:

- there is one software component acting as trader's agent, monitoring the market, making decisions, and performing transactions by communicating with on-line brokerage platform,
- the software is strictly bound to one particular brokerage company (regarding both data acquisition and trade execution),
- the model for making decisions is almost universally based on technical analysis rules,
- the possibility of integrating with external custom models (e.g. based on scientific computational environments) either does not exist, or is very limited and cumbersome in development.

Although it is sometimes possible to invoke custom model based on artificial intelligence and/or pattern recognition (e.g. in MetaTrader platform by using proprietary code in external DLL), doing so is a particularly difficult task and leads to a very hard to maintain software environment, tightly coupled with one particular software platform.

In comparison, a multi-agent automated trading environment aims at delivering a comprehensive software system, in which many diverse components could be freely integrated, with maintaining a loose coupling between them and without permanent coupling with any third party (e.g. brokerage platform, market data supplier, stock broker, etc.). These components could include for example scientific computational engines, various data sources, external artificial intelligence modules, GUI front-ends for interacting with the trader, or external models for market prediction, risk management, wallet construction etc.

To achieve the above goals, several categories of agents could be defined, each of them serving a particular purpose. All-in-one decision support and trading agent, as implemented by generic software, should be generally avoided, in order to maintain interoperability and possibility of composing more complex structures based on elementary components. The following basic set of agents could be defined:

- data collecting agent – connected to one or several data sources over Internet, such as Forex brokerage platform, stock data provider, news feed or market signals distributor;
- decision support agent (later on referred to as DSA) – implementing a particular forecasting model, e.g. based on any mixture of technical analysis, artificial intelligence or statistical pattern recognition, issuing recommendations for performing transactions;
- decision aggregation agent (later on referred to as DAA) – consuming recommendations from several decision agents and outputting one decision regarding transaction;
- trade execution agent – interacting with one particular brokerage company and forwarding the order to perform the transactions suggested by DSA or DAA agents;
- wallet management agent – further filtering the decisions suggested by DSA or DAA agents, based on such factors as current account balance, structure of currently hold assets, previous transactions history, or trader's preferences regarding the risk;
- performance monitoring agent – gathering information on DSA and DAA performance, i.e. some statistics regarding recommendations, percentage of profitable ones, including changes in time;
- user interaction agent – allowing the user to monitor and manage the whole system, as well as indicating personal preferences regarding transactions, risk, wallet composition, etc.

Obviously, the possibility to easily incorporate more agents would be essential, including:

- new instances of existing agent types (e.g. new model implementations with different input data, connectors to other data sources or trading platforms),
- new types of agents.

The general sketch of suggested multi-agent trading system architecture is presented in Fig. 1. More detailed description of system's architecture and other types of agents is beyond the scope of this work and will be omitted.

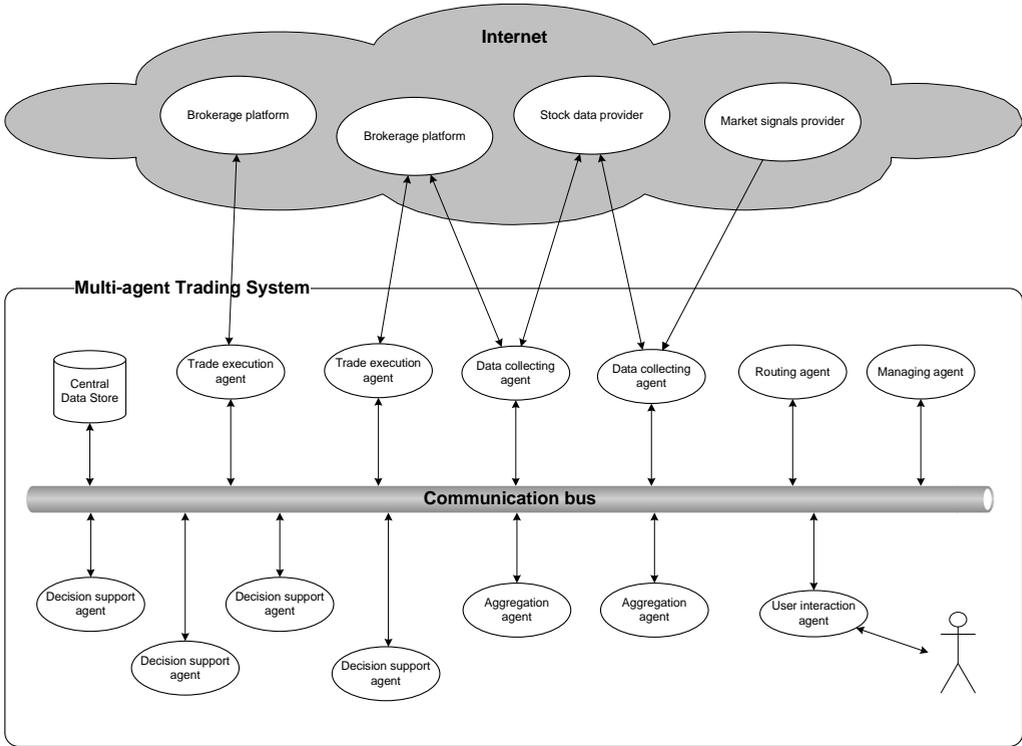


Fig. 1. Components of a multi-agent automated trading environment

2. DECISION AGGREGATION IN TRADING ENVIRONMENT

2.1. RATIONALE FOR DECISION AGGREGATION

In the context of automated trading environment, decision aggregation consists in making a final decision based on several partial recommendations. Output of several decision support agents (DSA) is fed into a decision aggregation agent (DAA), which applies some kind of aggregation scheme and outputs a final decision regarding the transaction – buying or selling an asset. Examples of architecture implementing this concept are presented in Fig. 2.

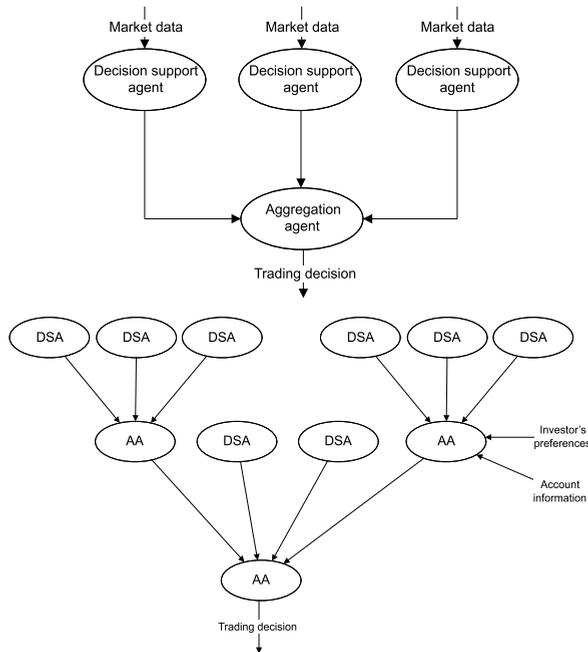


Fig. 2. Example of agents' decision aggregation organization (simple and hierarchical aggregation)

Generally two distinct and intrinsically different sets of motives for aggregating recommendations given by decision support agents can be formulated. The first one stems from the domain of statistical modelling and pattern recognition. There are several typical reasons for aggregating many sub-models into one complex model, shared by numerous common classification and regression tasks, which will not be covered here. The detailed discussion of such problems can be found in [3, 4, 7, 8, 9, 10].

The second group of reasons is inherent to the area of automated trading environments. First of all, a significant uncertainty is typical for trading decision support – none of the DSA agents can perform well during the whole operational period, whatever the current market conditions might be. Introducing aggregation agents forms a second layer of decision support, and thus allows for greater adaptability, e.g. by adjusting influence of particular decision agents, and even disabling some of them, should they stop to perform well. Secondly, some kind of black-box agents, developed by a third party, could be incorporated into the system. Since the exact nature of their forecasting model, performance, and – as a result – their credibility would be a priori unknown, the need to control them and confront with other agents naturally occurs. The overall performance of trading decision support, in terms of both profit gained and risk involved, could in great extent benefit from such a composition of diverse agents, therefore it is crucial to handle them properly.

Finally, such an aggregation enables the possibility to take trader's preferences into account, even when dealing with ready-made DSA agents. Some kind of risk measure could be incorporated in the evaluation of particular DSA agents, the investor would be asked to declare their inclination towards either high-profit or low-risk trading, and then aggregating agent would differentiate the role of particular decision agents in the decision making process.

2.2. REQUIREMENTS FOR DECISION AGGREGATION

Obviously, not all DSA agents can be mixed freely in order to form an aggregated model. A highly heterogeneous environment could contain agents differing in terms of transaction horizon (e.g. couple of hours, days or months), type of a predicted financial instrument (e.g. agent specialized in EUR/USD vs. general agent for forecasting any kind of CFD), type of recommendation (market buy/sell signal or stop/limit order, with or without protective stop loss/take profit orders) and many more. In order to accommodate such diverse mixture of agents, each of them should provide the system with a comprehensive set of metadata, characterizing agents predictive activities. Only then the system will be capable of clustering the agents and properly aggregating their outcomes.

The experiment described in this work deals with aggregating recommendations for trading EUR/USD currency pair. The participating agents were built upon simple decision rules, each of them gave discrete output values $\{-1, 1, 0\}$, denoting respectively: recommendation of selling an asset, buying an asset, and no recommendation from the given agent at all. All the agents assume roughly the same, short trading horizon of one to several hours. The agents indications suggest taking immediate action of placing a market order, without any protective stop/limit orders.

2.3. AGGREGATION OPERATORS

In general terms the aggregation of DSA agents recommendation consists in taking a set of decisions and applying an aggregating operator to them, which results in one concrete decision recommendation. Various aggregating operators can be used to achieve this goal, the most important and most widely used of them will be shortly described here.

Majority voting operator used in classifiers consists in assigning an object to the class indicated by the majority of the base models. Given M base models D_m and J classes C_j , the aggregated model D^* could be defined as [3]:

$$D^*(\mathbf{x}_i) = \arg \max_j \sum_{m=1}^M I(D_m(\mathbf{x}_i) = C_j) \quad (1)$$

where \mathbf{x}_i – classified observation, D_m – m -th base model, C_j – j -th class, M – number of models, J – number of classes, I – indicator function.

In the case of multi-agent trading environment, the above comes down to selecting transaction recommendation indicated by the majority of participating DSA agents – should most of them suggest buying an asset, such a decision will be pursued. Such a solution seems reasonable under two assumptions: that all the agents are equally credible, and that no trader’s preferences are to be taken into consideration during aggregation procedure. In most situations this is not the case, hence the need to apply some means to differentiate between agents.

In order to address the above concerns, weighted voting could be used, which is a simple, but arguably the most usable aggregation method taking into account base models’ significance. Assuming the same notation as above, the aggregated model D^* can be defined as [3]:

$$D^*(\mathbf{x}_i) = \arg \max_j \sum_{m=1}^M w_m I(D_m(\mathbf{x}_i) = C_j) \quad (2)$$

where w_m denotes the weight assigned to m -th base model.

When applying weighted voting, selecting the scheme for determining the weights is a crucial task. Two weighting schemes were proposed in [1] – WSE and WSP indicators – both of based on the current performance of each participating models or agents. In [2] WSE, WSP and ordered weighted averaging (OWA operator [4, 8]) were compared in terms of aggregated model’s performance. Although WSE proved to be superior, the advantage over WSP was minimal. Hence, the WSP being conceptually simpler and easier to calculate, it is advisable to utilize the latter. A momentary value of WSP indicator can be calculated as the percentage of profitable decisions over the last N_{ws} given recommendations:

$$WSP_{i,j} = \frac{n_{i,j}^+ |_{N_{ws}}}{N_{ws}} \quad (3)$$

where: $n_{i,j}^+$ – number of profitable recommendations, N_{ws} – number of past recommendations to be considered, j – index of the model.

To perform weighted voting, the weights for each of the participating agents should be proportional to momentary value of WSP. Both WSP and the weights should be recalculated at each data point.

3. STRATEGY FOR A ROBUST AGGREGATION AGENT

The above described procedures assume, that all the decision recommendations are present at the moment of performing the aggregation. In a real life scenario, there are multitude of reasons for a particular DSA agent not to be able to deliver a decision recommendation, when it is needed, such as:

- a communication failure or delay, caused by network issues, either between a DSA and a DAA, or between a DSA and external data provider; in both cases the recommendation cannot reach a DAA;
- data inconsistency caused by erroneous data sent by an external data provider (e.g. quote value out of range), which renders DSA unable to generate recommendation;
- instability of DSA implementation – a previously undetected software malfunction may cause a crash or hangout;
- algorithm divergence – a heuristic algorithm used in DAA may meet a previously unpredicted edge case, which leads to an infinite loop;
- exceeding computation time – for algorithms without time constraint, a particular edge case may lengthen the computation time, causing DAA to wait for an unpredictably long period.

As a result of such problems, two major flaws may impact the system. First, there might occur some technical stability issues. The overall system stability should never be influenced by the instability of participating agents. The whole environment should be loosely coupled and continue to function even with some agents inactive or damaged.

Second, the performance in terms of obtained financial income can be severely deteriorated. One of the reasons for introducing and aggregating multiple agents (and multiple forecasting models) is to reduce the impact of one particular agent on trading decisions, thus reducing the risk involved with trading. Without proper handling of above described problems, a technical failure in any one of participating agents will prevent the system from generating recommendations, causing it to miss profitable opportunities (loss of a potential profit) or a moment to withdraw from the market (an actual money loss). Furthermore, the risk of instability increases with the number of participating numbers.

In order to protect the system from the above described failures, the decision aggregation process can be split into two phases:

- data collecting phase – a DAA agent awaits for data from external sources, mostly from DSA agents;
- decision aggregation phase – a DAA agent analyses the input data and generates a recommendation.

To make the whole procedure robust, the following means could be facilitated:

- deadline for data collection – in each step of data processing (e.g. at each stock data point) a DAA agent sets up a deadline, after which no further data will be collected; should any DSA agent supply its recommendation later, it will not be taken into consideration; immediately after the deadline, a DAA proceeds to the next phase;
- data significance threshold – after collecting the partial recommendation, a DAA calculates the significance of obtained data; only upon exceeding a predefined threshold, the final recommendation will be generated and passed further on to executing agents.

The data significance threshold requires further explanation. In the simplest form this could mean, that a predefined number of DSA agents are required to supply a valid recommendation (e.g. 6 of 10 agents) in order for the DAA to proceed. In the more sophisticated scenario, metadata for DAA could be defined, specifying the following:

- input DSA agents, that are unconditionally required (without input from such agents no further action will be taken);
- importance weights for particular input agents;
- threshold for aggregation execution.

Each time a DAA agent enters the aggregation phase, it should proceed as follows:

- check the formal validity of all the supplied recommendations and discard invalid ones;
- check, if all the required DSA agents supplied recommendations; if not, abort the execution;
- calculate the significance indicator by summing up the importance weights for all those DSA agents, that managed to supply recommendations on time;
- check, if the calculated significance indicator exceeds a predefined threshold; if not, abort the execution;
- calculate WSP indicator for all participating agents, perform the aggregation and forward the resulting recommendation.

Such a procedure, following the above recommendations, will allow the system to function properly and bring profits even in case of particular agents failures.

CONCLUSIONS

One of the main goals of a multi-agent automated trading environment is to reduce risk by facilitating many diverse forecasting models simultaneously. As indicated by research results, decision aggregation may greatly increase the profits generated by decision support agents, while reducing the risk at the same time. However, in such a strongly distributed and heterogeneous environment, aggregation of agents' recom-

mendations must be handled properly. A procedure described in the work should prevent the system from instability and from influence of malfunctioning agents, both in terms of technical issues, and model deficiencies.

REFERENCES

- [1] CZAPIEWSKI P., *Adaptacyjna metoda indukcji reguł w inwestycyjnym procesie decyzyjnym z zastosowaniem agenta*. Doctoral dissertation, West Pomeranian University of Technology, Szczecin, 2009.
- [2] CZAPIEWSKI P., Decision aggregation in a multi-agent automated trading environment, in: *Applied Informatics. Selected Issues*, Łatuszyńska M., Radliński Ł. (eds.), Szczecin 2010, pp. 83–94
- [3] GATNAR E., *Podejście wielomodelowe w zagadnieniach dyskryminacji i regresji*. PWN 2008.
- [4] FULLER R., *OWA Operators in Decision Making*, in: Carlsson C. (ed.), *Exploring the Limits of Support Systems*. TUCS General Publications, No. 3, Turku Centre for Computer Science, pp. 85–104, 1996.
- [5] OBJECT MANAGEMENT GROUP. *Agent Technology Green Paper* [online]. <http://www.objs.com/agent/index.html>
- [6] WOOLDRIDGE M., JENNINGS N., *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review, vol. 10, 2, 1995.
- [7] WOŹNIAK M., *Metody fuzji informacji dla komputerowych systemów rozpoznawania*. Oficyna Wydawnicza Politechniki Wrocławskiej, 2006.
- [8] YAGER R., *On ordered weighted averaging aggregation operators in multi-criteria decision making*. IEEE Transactions on Systems, Man and Cybernetics 18, pp. 183–190, 1988.
- [9] ZHANG Z., ZHANG Ch., *Agent-Based Hybrid Intelligent Systems*. Springer 2004.
- [10] ZHANG Z., *Decision Aggregation in an Agent-Based Financial Investment Planning System*, in: *Modeling Decisions for Artificial Intelligence*, Torra V., Narukawa Y., Valls A., Domingo-Ferrer J. (eds.), pp. 179–190, Springer 2006.

Katarzyna MICHALSKA*, Tomasz WALKOWIAK*

ANALYTICAL AND EXPERIMENTAL DEPENDABILITY METRICS FOR SERVICE-BASED INFORMATION SYSTEM

Work presents several dependability metrics of service based information systems. Methods of metric calculation are also taken into consideration. Two general groups of metrics are analysed: analytical and experimental. Following analytical metrics are analysed: network cohesion in the system, number of services involved in the compound service, absolute importance, dependence and criticality of the service and overall reliability of the compound service. Experimental metrics, i.e. task response time and service component availability, are calculated by developed by author's simulation tool. Numerical experiments were performed on a test case scenario.

1. INTRODUCTION

In today's business and service-based information systems knowledge about measurements and metrics of system parts is not only connected with better understanding these systems, but also as a set of means about business values. No, wonder then, that this approach is still widely used in almost all present-day domains and primarily has been adapted to industry as well as hi-tech field. Furthermore, measurement started to benefit from as basis for control systems, which at present, often are compound computer aided platforms for managing other systems.

In computer science, metrics are common and widely used, still in case of service-based information systems this area is still under development [4,6,7,9]. Researchers are trying to invent or improve metrics in case of better system evaluation. Such situation can be especially seen in the dependability area. In this paper, we present some of the metrics focusing on two (analytical and experimental)

* Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, ul. Janiszewskiego 11/17, 50-372 Wrocław, Poland.

methods of their calculation. As a result, we try to look to a comparison of their results and answer the question about their utilization in a service-based information system area.

Since Information Systems must be proved experimentally on either real systems or simulations based on traces from real systems in section 2 we present short description of simulation as a methods of system analysis. Next, we propose some defined dependability metrics (section 3) that will be used in examined exemplary case scenario (section 4). Finally, there are conclusions and plans for further work.

2. SIMULATION

Simulation is an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works [4,5]. Going further, we can say that simulator is a tool that tries to mimic the behaviour of a system. In literature, two main types of simulators can be found: a continuous time and discrete event based [5].

In the area of simulation issues, there main tasks/processes can be named:

1. Designing of simulation model;
2. Calculation of the model (using simulation algorithms)
3. Results analysis

Simulation of a physical reality requires creating mathematical model that represents its nature. It can be represented in various ways starting from declarative form, functional, spatial and multi-model. After proposing suitable form of the representation simulation can be done using one or more units (depends of a dispersion of large-scale models). It should be noted, that simulation can be done on various levels of abstraction considering user needs, and data needed for results. The more precise models, the more precise results that is way a suitable model should be chosen from a set of models in a set of: stochastical (using random numbers generators); deterministic (continuous time or discrete event based); distributed (realized as a network) and agent based (agents are represented as a part of the system working as a executable threads).

2.1. MS CLASSIFICATION

For modelling and simulation technique (MS) [5] large number of tools are available. Moreover this number is still growing, since the requirements become larger and larger in all four groups of its main classification, that is:

1. Analytical tools – help to design a network model and calculate different factors (e.g. reliability) of the model.

2. Simulation tools – simulate dynamic behaviours (e.g. link failure) of a network model besides modelling
3. Topology discovery tools – extract the actual network information from an existing system and map them graphically and/or in text format
4. Topology generation tools – help to generate small as well as large topologies based on different algorithms.

Figure 1 [5] shows farther classification of the tools with respect to their commercial and educational classification.

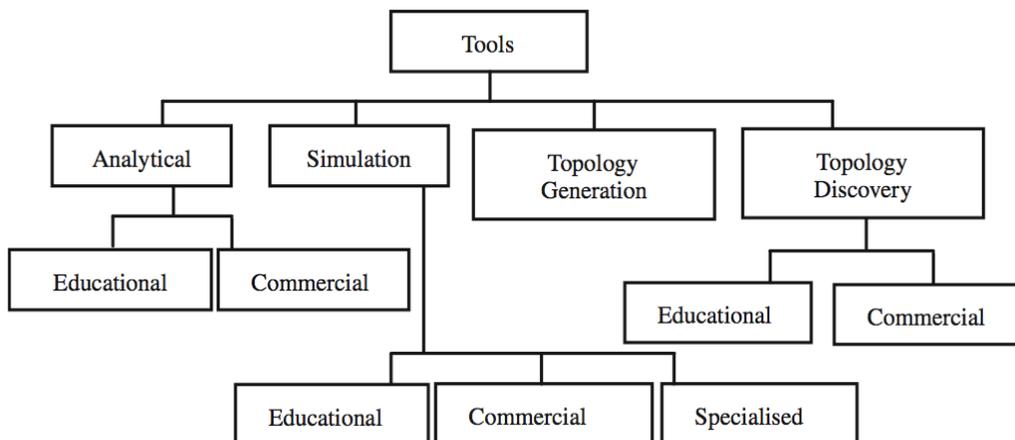


Fig. 1. Classification of the network design and simulation tools

2.2. SSFNET – A TOOL FOR ANALYSING SERVICE-BASED INFORMATION SYSTEM

The event-simulation used in this paper is based on the Scalable Simulation Framework (SSF) [4] which is a used for SSFNet [4] computer network simulator. For the purpose of simulating service-based systems we have used Parallel Real-time Immersive Modelling Environment (PRIME) [5,8] implementation of SSF due to a much better documentation then available for the original SSF. We have developed a generic class derived from SSF Entity which is a base of classes modelling system objects. It is important to notice that the Monte-Carlo approach is used. The original SSF was not designed for this purpose so some changes in SSF core were done to allow to restart the simulation from time zero several times within one run of simulation program [8].

The statistical analysis of the system behaviour requires a very large number of simulation repetition, therefore the time performance of developed simulator is very important.

3. NETWORK DEPENDABILITY METRICS

In case of further metrics definitions, some algebraic operation must be noted [6, 7]:

- Selection $\sigma_{\text{condition},...}(\mathbf{R})$ – selection of row in relation \mathbf{R} that correspond to a given condition set;
- Projection $\pi_{\text{attribute_id},...}(\mathbf{R})$ – selection of columns in relation \mathbf{R} that eliminates duplicates.

Using this operation, we can define and calculate metrics that will be presented below.

3.1. ANALYTICAL METRICS

One of the metrics is called **Network Cohesion in the System (NCY)**, understood as a number of direct (unidirectional) links between system nodes. Metric is calculated as:

$$\text{NCY} = |\pi_{\text{InvokerNode,ServiceNode}}(\sigma_{\text{InvokerNode} \neq \text{ServiceNode}}(\mathbf{A}))|$$

where:

$\pi_{\text{atrybut}}(\mathbf{R})$ – is a projection of the set;

$\sigma_{\text{warunek}}(\mathbf{R})$ – is a selection of the set with respect to conditions of \mathbf{R} relation;

Unidirectional link between node $\mathbf{N1}$ and $\mathbf{N2}$ means that induction of the service provided by node $\mathbf{N2}$ from the client is located on the node $\mathbf{N1}$.

Number of Services Involved in the Compound Service (NSIC [c]) – can be understood as a number of services related to complex services both directly and indirectly:

$$\text{NSIC}[c] = |\mathbf{S}^{\mathbf{R}}[c]|$$

This metrics is the simplest indicator of service complexity.

Absolute Importance of the Service (AIS [s]) – is a number of clients, that are part of the set s using their methods. It should be noted that number of clients in a node n is not taken into consideration (not counted), because communication between software on the same node is not causing additional load of network. Moreover, possibility of other service availability at the same node is almost equal to zero. For these reasons, Absolute Importance of the Service s can be calculated as:

$$\text{AIS}[s] = |\pi_{\text{Invoker}}(\sigma_{\text{InvokerNode}=n, \text{Service}=s}(\mathbf{A}))|$$

Absolute Dependence of the Service (ADS [s]) – is defined as a number of services that service s is depended on:

$$\text{ADS}[s] = |\pi_{\text{Service}}(\sigma_{\text{Invoker}=s, \text{ServiceNode}=n}(\mathbf{A}))|$$

Value of this metric is inversely proportional to self-sufficiency of the service, in consequence to a level of its potential autonomy and usability in other environment.

Absolute Criticality of the Service (ACS [s]) – is a the product of its absolute importance and absolute dependence:

$$ACS[s] = AIS[s] \times ADS[s]$$

It represents a point of attention that the creator of a service-oriented system must devote to service s .

Neither important services that do not depend on a number of other services, nor insignificant services that depend on many other services are not as critical for the system, as services that are both very important and have many dependencies.

Overall Reliability of the Compound Service (RC [c]) – is the reliability of a complex service c that is inversely proportional to the weakest cell in its chain (reliability of the most important services in complex service c):

$$RC[c] = \frac{1}{\max(\{ACS[s] | s \in S[c]\})}$$

This metrics can be seen as a main parameter/metrics for dependability of complex services.

3.2. EXPERIMENTAL METRICS

Task Response Time (TRT) – This metric is a numerical representation of clients perception of particular business service quality. It is calculated as an average delay between the starting time of user response ($t_{i_request}$) and getting answer ($t_{i_response}$) from the business service (i.e. only requests that were properly answered are taken into account).

$$TRT = \frac{1}{N_requests} \sum_{i=1}^{N_requests} t_{i_response} - t_{i_request}$$

It is important to state that it is calculated for all properly answered requests ($N_requests$) in all simulation runs.

Service Component Availability (SCA) – defined as a ratio of the expected value of the uptime of a system to the observation time. It means that information system is working in a business sense (considering all sub-services that is service components). We proposed [8] to estimate the availability as a ratio of properly answered requests ($N_{correct}$) over all requests ($N_{reuests}$):

$$SCA = \frac{N_{correct}}{N_{requests}} 100\%$$

4. TEST CASE SCENARIO

As an exemplary case study for our solution, we propose simple service-based information system.

The testbed is build of several elements:

- client network that represents clients hosts
- network security and communication devices: *Firewall_1* and *Firewall_2*,
- server farm that includes hosts: *WebServer*, *DNSServer*, *DBServer*, *AuthServer*, *DataTransationServer*.

Given example is thought to be an ticketing system where the main service scenario is presented on Figure 2.

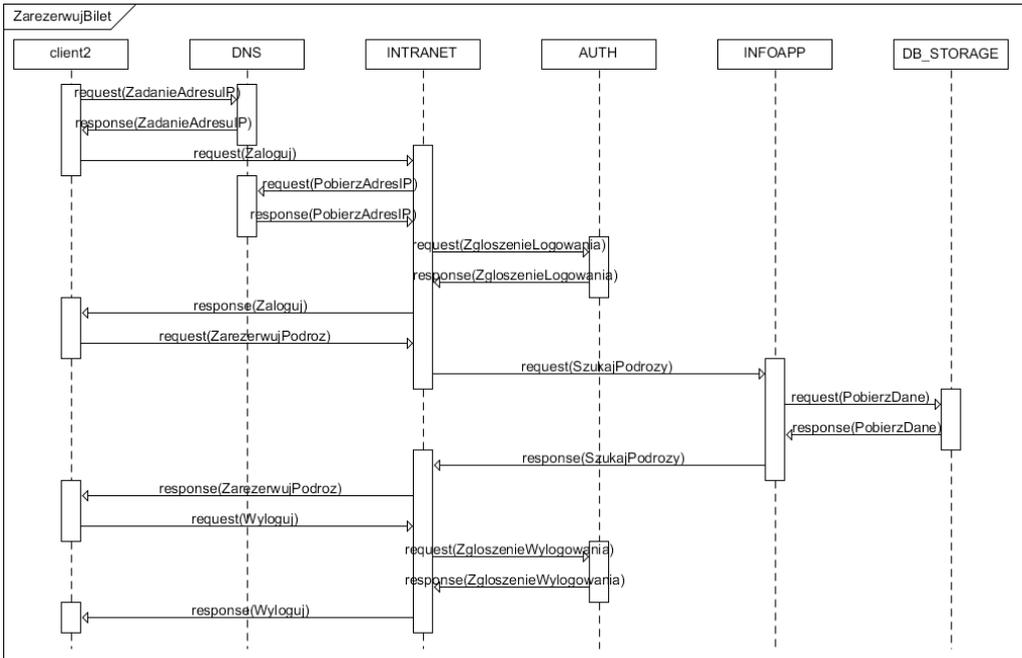


Fig. 2. System infrastructure overview

For the purpose of further analysis, each service is located on only one machine and its failure is no longer than a half of the simulation time.

Metrics proposed in section 3 where calculated at the specified test case scenario (Fig. 2).

Number of Services Involved in the Compound Service (NSIC):

client2 = 5 [DNS, INTRANET, AUTH, INFOAPP, DB_STORAGE]

Network Cohesion in the System (NCS) is: 11

Absolute Dependence of the Service (ADS):

INTRANET=3 [DNS, AUTH, INFOAPP]

DB_STORAGE=0 []

AUTH=0 []

DNS=0 []

INFOAPP=1 [DB_STORAGE]

CLIENT=0 []

Absolute Importance of the Service (AIS):

INTRANET=1 [client2]

DB_STORAGE=1 [INFOAPP]

AUTH=1 [INTRANET]

DNS=2 [client2, INTRANET]

INFOAPP=1 [INTRANET]

CLIENT=0 []

Absolute Criticality of the Service (ACS):

INTRANET=3

DB_STORAGE=0

AUTH=0

DNS=0

INFOAPP=1

CLIENT=0

Overall Reliability of the Compound Service: 0.3333333333333333

From the results given below, we can see that the client2 uses 5 services which have various criticality for the whole service. The most dependent service is INTERNET, next (also worth to analyze) is an INFOAPP. Because of the low values of ACS and AIS, Overall reliability of Compound Service is not reaching value 1. It means that the system can be seen as dependable.

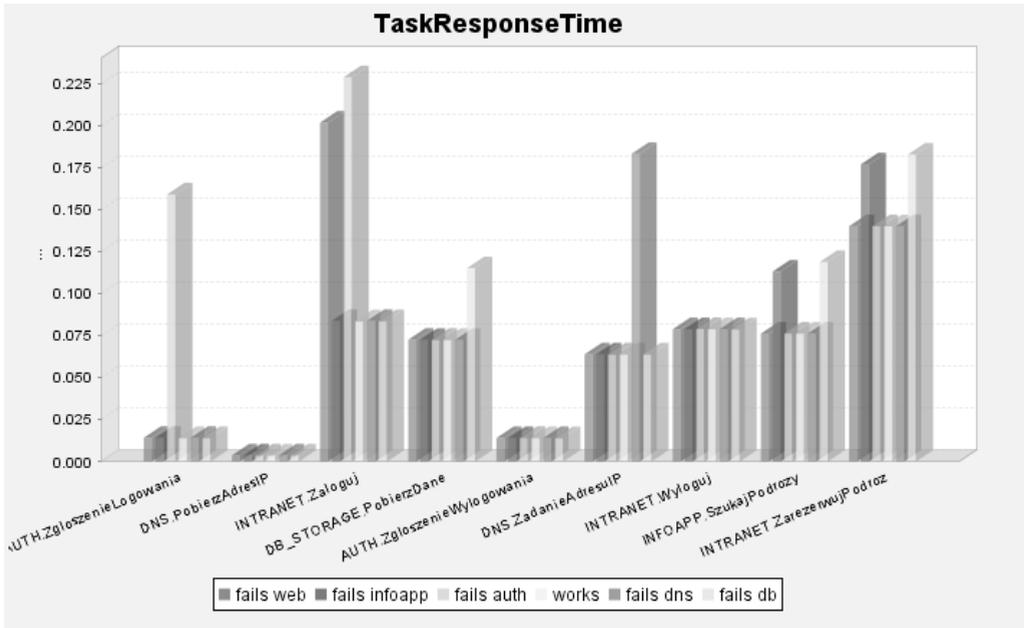


Fig. 3. Task Response Time (TRT)

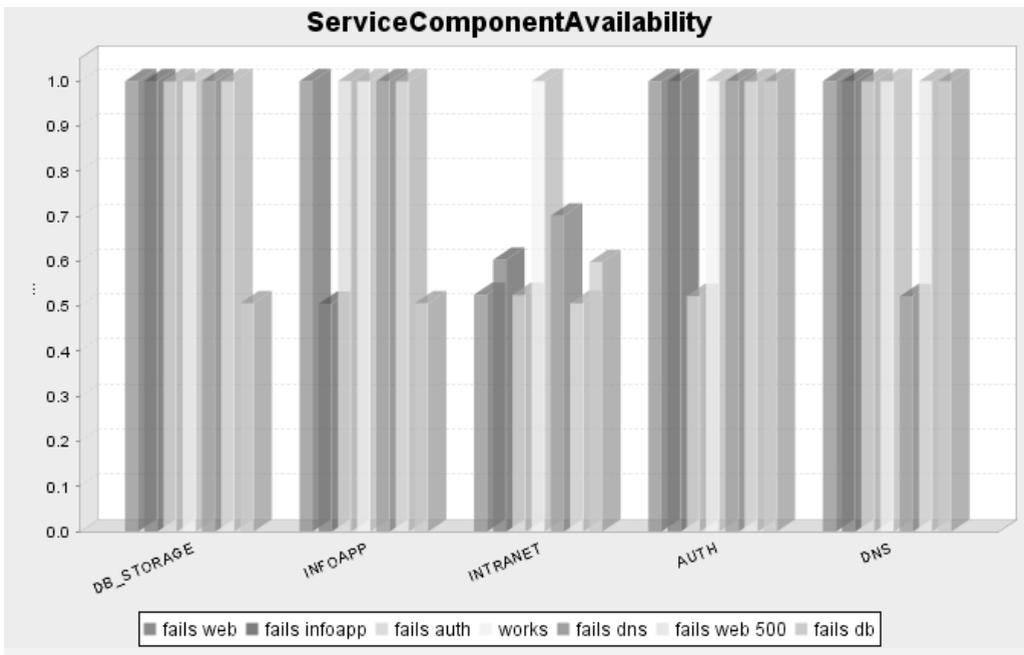


Fig. 4. Service Component Availability (SCA)

Comparing results of SCA metric with ADS metric, we can see that INTERNET service is crucial for whole system, since failure of this element influence other parts. Moreover, since (as shown on Fig. 2) INTERNET is using INFOAPP and depends on DB_STORAGE than based on mathematical transitivity it will influence INTERNET.

On the other hand, looking at the results of TRT in all faulty cases we can observe that if database server (and a service located on the node – DB_STORAGE) is unavailable than user will wait for a longer time for a proper data and may cause to a significant INTERNET delays. These results are not convergent with ADS, AIS and ACS metrics.

5. CONCLUSIONS

We have presented a set of analytical and experimental dependability metrics of service based information systems. The analytical set of metrics included: network cohesion in the system, number of services involved in the compound service, absolute importance, dependence and criticality of the service and overall reliability of the compound service. Moreover, two experimental set of metrics, i.e. task response time and service component availability are presented.

Authors developed a computer tool (with task level simulator developed by authors [3]) that allows to model [8] an information system and calculate automatically all presented metrics.

The proposed method of analyzing service based information systems shows its effectiveness. Integrating whole modelling and analysis process makes the tool useful for information system designers and administrators.

REFERENCES

- [1] AVIŽIENIS A., LAPRIE J., RANDELL B., *Fundamental Concepts of Dependability*, 3rd Information Survivability Workshop (ISW-2000), Boston, Massachusetts, USA, 2000.
- [2] EUSGELD I., FREILING F.C., REUSSNER R., *Dependability Metrics*, LNCS 4909, Springer-Verlag, 2008, pp. 1–4.
- [3] MICHALSKA K., WALKOWIAK T., *Fuzzy Reliability Analysis of Simulated Web Systems*; Springer LNCS/LNAI series; 2011.
- [4] NICOL, D., LIU, J., LILJENSTAM, M., GUANHUA Y., *Simulation of large scale networks using SSF*, Proceedings of the 2003 Winter Simulation Conference (2003). Volume 1, 7–10 December 2003, pp. 650–657.
- [5] NIKOLAIDOU M., ANAGNOSTOPOULOS D., *A distribution system simulation modelling approach*, Simulation modelling Practice and theory, vol. 11, pp. 251–267, 2003.
- [6] RUD, D., SCHMIETENDORF, A., DUMKE, R., *Product metrics for service-oriented infrastructures*. In: Applied Software Measurement. Proc. of the International Workshop on Software Metrics

- and DASMA Software Metrik Kongress (IWSM/MetriKon 2006). Magdeburger Schriften zum Empirischen Software Engineering, Potsdam, Germany, Hasso-Plattner-Institut, Shaker Verlag, November 2006, pp. 161–174
- [7] RUD, D., SCHMIETENDORF, A., DUMKE, R., *Resource metrics for service-oriented infrastructures*, In Proc. SEMSOA 2007, pp. 90–98, May 10–11, 2007, Hannover, Germany
- [8] WALKOWIAK T., MICHALSKA K., *Simulation approach to performance analysis information systems with load balancer*, W: Information systems architecture and technology : advances in Web-Age Information Systems; Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2009. s. 269–278.
- [9] ZHANG, J., ZHANG, L.-J., *Criteria analysis and validation of the reliability of Web services-oriented systems*, Web Services, IEEE International Conference on Web Services (ICWS'05), 2005, pp. 621–628

Adam CZYSZCZON*, Aleksander ZGRZYWA*

AN ARTIFICIAL NEURAL NETWORK APPROACH TO RESTFUL WEB SERVICES IDENTIFICATION

Recent studies that investigate the state of Web services show that their number is quickly growing. However, Web services are meaningful only if potential users can find sufficient information in order to use them. This means that as long as they remain unidentified in private and isolated directories of service providers and as long as there are no tools which help programmers find desired services, they are useless. The first step to change it is to propose a solution to identify services and then to create discovery tools such as Web service search engine dedicated for instance for web developers. Aimed at solving mentioned problems this work introduces a method for RESTful Web services identification. Presented approach is preceded by the definition of REST based Web service URI structure. With the usage of artificial neural network models presented method allows the classification of RESTful Web services based on their link structure patterns. Introduced research includes the analysis of service's resources and their variables in order to create a generic description of particular Web service. This work also introduces a running engine implemented in order to apply presented approach.

1. INTRODUCTION

Despite the fact that the market for Internet services is just beginning its development recent studies that investigate the state of Web services show that their number is quickly growing [2, 5]. The REST based approach into services provides a simple and lightweight solution which is followed by many Web service providers. In recent years, many APIs (Application Programming Interfaces) have been created for Web services including Amazon, Google, Facebook, Flickr, Twitter, Yahoo, YouTube and many others. However, Web services are meaningful only if potential users can find

* Wrocław University of Technology, Faculty of Computer Science and Management , Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.

sufficient information in order to use them. Thus, as long as they remain unidentified in private and isolated directories of service providers, potential developers have limited use of them. Additionally, currently there are no methods of RESTful Web services identification as well as there are no techniques of their discovery. Moreover, adopted so far attempts to their standardisation bring little benefits because they are not widely accepted. For example, solutions proposed in order to standardize interface definition like Web Application Description Language (WADL) [7] and WS-BPEL for REST [4] still are not accepted as standards, therefore they are not commonly used among RESTful Web services development community. The possibility to identify services is the first step to create discovery tools such as Web service search engine dedicated for instance for web developers.

On the Internet there are many useful RESTful services that can save hours of work for programmers. Unfortunately, as long as there are no tools which help programmers find desired services they are useless. Besides, it is not obvious where to look for them. Service providers share their API documentation where there are descriptions of how to use their services. Thus, one can use traditional search engines to find providers and check their documentation manually. However, searching without proper tools is so time consuming that often the only solution for developers is to create a service themselves. Another difficulty is that many services are still undocumented.

Solving the problem of finding services is the motivation of this work. Bearing in mind the difficulties in service finding on the Internet, this work introduces a solution for RESTful Web Services identification. Presented approach also includes the analysis of resources and their variables in order to create a generic description of particular Web service. The goal is following: to create a uniform and universal RESTful Web service URI structure to allow their identification. Next, to propose engine architecture that gives the ability to effectively identify the services according to their structure. By taking into account that the construction of the URI of such services is composed of many elements and because of the fact that their structure is often ambiguous and difficult to evaluate, it is proposed to use of artificial neural networks as a key element to their identification.

Solutions proposed in the work carry the following benefits: the definition of RESTful Web service URI structure is necessary to enable the recognition and identification of a RESTful service. Secondly, engine design which identifies the service based on neural networks can in turn be used as part of a web crawler architecture that traverses the Internet and finds Web services. Such a solution is proposed in the section of this work. The combination of such a spider with indexer and search engine can be in turn used to create Web service search engine. This would bring great convenience for many programmers, web developers and to RESTful Web services community.

2. RESTFUL WEB SERVICE URI STRUCTURE

Although the URI structure of Web service has definite pattern many service providers use different conventions while designing their services. Moreover, the structure of services still differs significantly between services even within the same service provider. This makes it difficult not only to identify the structural elements of a service but also properly recognize them. In order to create universal URI scheme for REST based services, one should firstly take a look at most popular examples of their URI structure. According to the specification of URI its generic design is composed of: *scheme*, *authority*, *path*, *query* and *fragment*. Its simplified form can be denoted as:

```
scheme://authority/path?query#fragment
```

Transferring this structure into its RESTful equivalent is crucial in order to properly recognize the REST compliant services. First of all they are accessed explicitly over HTTP or HTTPS as for the *scheme* part. The *authority* is composed of a hostname. Some of service providers have special hostname for most of their services, for instance: *search.yahooapis.com*, *query.yahooapis.com* or *where.yahooapis.com*. However, most of them start with a prefix followed by provider's domain name. Usually the prefix is *api.** or *ws.**, for example *api.twitter.com*, *api.del.icio.us* or *ws.audioscrobbler.com*. It is of course possible that no prefix is present like it is done in Google with their *googleapis.com* hostname.

Most important thing in sense of Web service recognition is *path* and *query* part of URI. While the *query* mostly contains data only about service's resources the *path* element is more complex and usually carries four types of information: service's name, version, access policy and resources. On the beginning of the *path* there is section about service's name and version. Some WSs use version first and then service name, other vice versa or do not add version and name at all (at least not in the *path* section as there are also some cases where service's name or version is placed in *query* part). Sometimes the *path* also informs about service's access policy, namely whether it is public or private. Following examples are taken from Google, Yahoo and Twitter and respectively present three different ways of describing Web services: */urlshortener/v1*, */v1/public/yql*, */1/notifications/follow.json*. Since considered elements are not always that meaningful, it is easy to notice the difficulty to determine which part states for name, version and which for access policy.

Next part of the *path* is devoted to resources. According to RESTful Web service design principles [6] URI should have directory structure-like form, also referred to as "pretty URLs". This method avoids using query strings in favour of tree-like hierarchical structure, rooted at a single path. Conforming to BPEL for REST [4] resources of REST compliant Web services may have declared some variables. After implementing such a service values of those variables which in turn correspond to

different resources are visible in the URI. In the case of tree-like path structure the values are preceded by a resource, for example: `/book/{author}/{title}`. Despite of “pretty URL” recommendations, many URIs are still based on *query*. Moreover, for certain services *path* and *query* parts are mixed. This makes the recognition process even more complicated. Generally, one can distinguish four types of different URI resource representation – *path tree-like structure*, *path tree-like structure with query variables*, *query structure* and *mixed structure*. Example below presents them respectively:

```
/books/categories/{category}/authors/{author}
/books?categories={category}&authors={author}
/?books=null&categories={category}&authors={author}
/books/categories/{category}/?authors={author}
```

With presented above different structures it is easy to pull up books within given category and author. The difference between second and fourth example is that in the first case there is only one resource (*books*) with two variables (*categories* and *authors*) whereas in the *mixed structure* there are three resources – *books* and *categories* in the *path* part and separately *authors* in query part. Such a structure is used in *last.fm* API for instance. Despite the fact of using tree-like structure (first example) is recommended all presented parts of URI are still rather intuitive and predictable. However, even though it is relatively easy to extract information about resources, still it is hard to determine which element of path corresponds to a resource and which represents its values.

Regardless of difficulties of explicit recognition of the key elements of the URI it is however possible to clearly define the URI structure for a RESTful Web service. It is as follows¹:

```
http[s]://hostname/[serviceInfo/][resources/][?query]
```

Parts in square brackets are optional where at least one of them must be present. If no *serviceInfo* and *resources* are present the *path* has its shortest possible value which is “/”. The *hostname* is composed of `[WSprefix(.-)]hostName` where *WSprefix* is an optional prefix to the *hostName* element, separated by dot or minus character. The *host* parameter is a hostname.

The *serviceInfo* is composed of `[version/][accessPolicy/][serviceName/]` where its consequent parameters are indicators of service’s version, access policy and name. The order of the parameters is not fixed and may occur in different order.

¹ Please note that in order to simplify expressions presented in this work do not conform to syntax of regular expressions (also referred to as regex). There are only two rules they follow: expression in square brackets is optional and asterisk sign means that the expression in parentheses can occur zero or more times.

The *resources* parameter is composed of $(resourceName/[value/]^*)^*$ where every resource has a name and its corresponding value. It may also happen that *resourceName* has more than one *value* or no values at all.

The *queryResources* parameter is composed of $(resourceName=value[&])^*$ where, as in the case of previous parameter, represents resources names and their corresponding values, separated by ampersand character if there is more than one resource. In case of resources with multiple values they are usually separated by plus, comma, semicolon and space character. It is also easy to notice that many websites may be considered as a RESTful service. The truth is that a web page is a representation of one of WS resources and similarly it is accessed by its URI. A client can request a specific representation of the resource from the representations the server makes available. The task is to also determine whether a particular website is a representation of a WS or it is just an ordinary web page not being connected to any part of an API.

3. RESTFUL WEB SERVICE IDENTIFICATION

Once having defined the URI structure for RESTful Web services it is possible to create intelligent tools which together form compact engine aimed at identifying those services and create their description. The process of identification consists of several stages related to pattern matching, pattern learning and pattern recognition. It uses both Artificial Neural Network (ANN) model with back-propagation algorithm and Adaptive Resonance Theory (ART) type 1 used for binary inputs. Because the ART 1 system is an unsupervised learning model [3] and because it provides data for ANN network, the whole engine works as an unsupervised system too. The algorithm of the engine together with its components is shown as Figure 1.

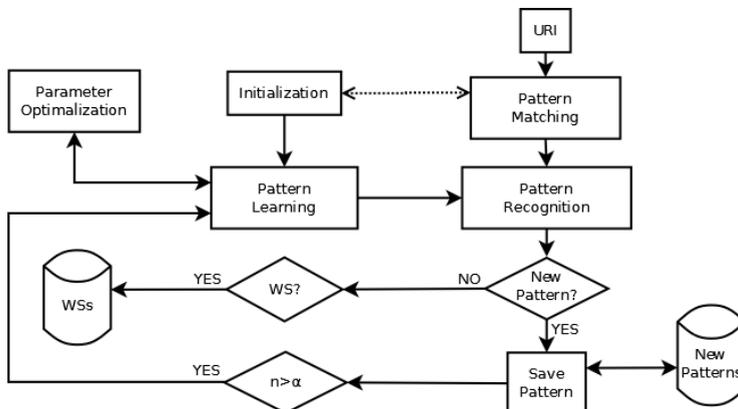


Fig. 1. RESTful Web Services identification engine.

The first step is the *Initialization* process which constructs a data set to train the network. Firstly, module takes specially prepared, non-duplicate set of URIs where half of them are RESTful Web services and another half are any other addresses. Afterwards, it connects to *Pattern Matching* module which converts every URI into training vectors and returns them to *Initialization* module. Training vectors are later passed to *Pattern Learning* component. There, two types of networks are trained. The first one is the ART network which learns all given patterns. It is later used to learn new patterns automatically. The second network is the ANN which is used for classification. It is trained against different activation functions for input, hidden and output layers using *Parameter Optimization* module. After optimal parameters are found the network is trained again using the back-propagation algorithm until Mean Squared Error (MSE) is suitably small. Such a network is then ready to recognize patterns. The initialization process is done only once.

In the next step the *Pattern Matching* module uses parsing algorithm which divides each incoming URI into parts in order to match RESTful URI structure patterns defined in previous chapter. Each part is successively analysed for the presence of particular characteristic. If a feature is present it is assigned a value of one, if not – zero. In result, a specific binary profile is created for each analyzed URI.

First group of characteristics is associated with *hostname* scheme. At this point one can distinguish four pattern parameters: *API indicator*, *access policy indicator*, *version*, and *path WS name*. First parameter indicates whether given service is part of an API. To find it out pattern recognition tool checks if there is proper *WSprefix* or “api” string match in *hostName*. Second parameter indicates whether there is access policy defined simply by matching “public”, “private” or similar string in *path*. Third decides if service was given a version number by using adequate regular expression. The last one checks if there is a name for the service in the *path*.

Second group of pattern characteristics are resources which have four parameters: *path resources*, *query variables*, *mixed resources*, and *query resources*. On this level given URIs are checked in terms of their resource structure as explained in previous chapter. The last characteristic determines whether Web service’s name is included in *query*. If there is no name in *serviceInfo* the algorithm analyses query in order to extract name for a service.

Apart from matching patterns only the module also extracts information (if present) about every input URI, that is about its name, version, resources and resource variables. The whole RESTful Web service URI pattern is presented below in a form of a list, followed by two concise examples of REST based services:

```
[apiIndicator, accessIndicator, version, pathWSname, pathResources,
queryVariables, mixedResources, queryResources, queryWSname]
http://where.yahooapis.com/geocode?q=some+address&appid=[appid]
Name: geocode, qRes: {`q`: `some+address`, `appid`: `[appid]`}
```

```
Pattern: 1 0 0 1 0 0 0 1 0
http://ws.audioscrobbler.com/2.0/?method=album.getshouts&artist=cher
&album=believe&api_key={key}
Name: album, Version: 2.0, qRes: {`method`:`album.getshouts`, `art-
ist`:`cher`, `album`:`believe`, `api_key`:`{key}`}
Pattern: 1 0 1 0 0 0 0 1 1
```

First two URIs represent services with query only resources. In the first example the last element of the path represents its name, not a resource. Therefore all of the resources are contained in query part of URI. The second example illustrates a situation where service's name is included in query. The API indicator is present in all services.

The third step is recognizing matched pattern of a URI by *Pattern Recognition* module. The recognition is implemented in two stages. At first stage the pattern is passed through ART network in order to determine if given pattern have been seen before.

When new pattern is found the ART network automatically learns it and temporarily saves it in a *New Patterns Dictionary*. If the number of patterns in dictionary exceeds certain α value, the ANN network is trained again together with newly discovered patterns from the dictionary. As the pattern recognition reacts to changing environment the identification process stays up to date with changing RESTful Web services pattern trends and improves the recognition accuracy.

If the pattern is recognized as "seen" by ART network and as a RESTful Web service by ANN network, it is added to the *WSs Database*. The database contains all the information extracted by the *Pattern Matching* module as presented in the previous examples. All information is collected and gradually supplemented in order to make complete description of identified services. However, one of the difficulties mentioned in previous chapter was how to check which element of path corresponds to a resource and which represents its values. Collecting information about resources is done in two steps. Firstly, learning about service's resources is done by reading their responses. Secondly, it is done by comparing the same services and checking which parts of *path* remain constant. Those parts which are fixed are marked as resources and the variable ones are their values.

4. RESEARCH RESULTS AND FUTURE WORK

Based on the research of RESTful Web service identification proposed in previous chapter, one implemented working engine by which initial tests were conducted. The analysis was based on a web crawler running on a single computer. The aim of the crawler was finding URIs while traversing the Internet graph by following hyperlinks.

It also analyzed the content of web pages in terms of finding URIs and checking if they conform to RESTful services URI structure. The searching locations concerned hosts which belonged to four well known REST based Web service providers: *www.last.fm/api*, *code.google.com*, *developer.yahoo.com*, *www.delicious.com/help*. The purpose of the experiment was preliminary assessment of the functioning of the engine, possible improvement of its elements, and measuring the accuracy of proposed pattern recognition mechanisms. The engine was running on Intel Core 2 Duo T7500 2x2, x64 2GHz, 2GB RAM, Ubuntu 10.10 (Linux 6.2.35).

Desired MSE for the network training was set to 0.0001, the value of α parameter was set to 3 and optimal parameters obtained by *Parameter Optimization* module were: 8 hidden nodes, learning rate of 0.5, momentum factor of 0.1, linear activation function for hidden layer nodes and sigmoidal activation function for output layer nodes.

The test showed that among total 321 107 analyzed URIs 4170 of them were identified as RESTful Web services which gives 1.3%. In total the engine recognized 20 new patterns. For every service provider there were some new patterns found. This means that in order to improve accuracy the ANN network was retrained 1 time in first case, 3 times in second, 2 in third one, and 0 in last one. Detailed results are presented in table below.

Table 1. Test results of URIs discovered and identified as RESTful Web services.

	last.fm	google.com	yahoo.com	delicious.com	Total
Total URIs	95429	75920	59720	38	321107
Total WS URIs	1225	861	2063	21	4170
% of WS URIs	1.28%	1.13%	3.45%	55.26%	1.30%
New patterns	3	9	8	0	20

To measure accuracy one should compare identified services to the amount of services provided by investigated service providers. First column presented in Figure 2 shows the total number of identified services on given hosts while second column represents total services provided. In every case the number of identified services is higher than the amount of services offered by providers. This is because the engine identified some extra services (fourth column) which belonged to another service provider located on different host (fifth column) or the engine found some other services on the same host which were not “officially” defined as a service within given web site (sixth column). It was also possible that identified services differentiated in version. In case of *www.last.fm/api* the engine identified only 9 of 15 services provided which gives 60%. However, only 9 services of *last.fm* were discoverable because had their URIs exposed on web site. This means that 100% of discoverable services were identified. In case of *code.google.com* the accuracy was 93%. For *developer.yahoo.com* it was 91% and for *www.delicious.com/help* it was 100%.

The engine worked well for both web sites of small Web service providers with few services and low link interconnection structure and for large providers' pages belonging to *last.fm*, *google* or *yahoo*. Performed test not only clearly proves the correctness of introduced REST Web services URI structure but also demonstrates the accuracy of presented pattern recognition mechanisms.

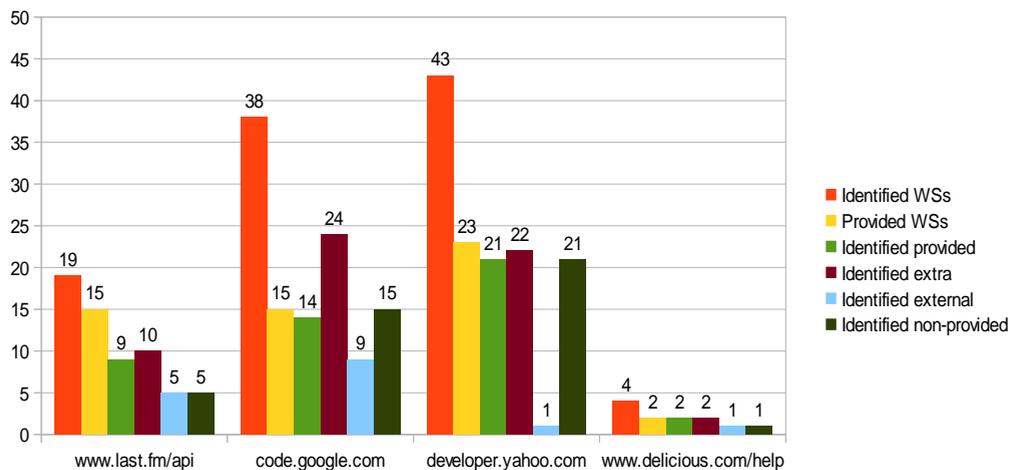


Fig. 2. Test results of services identification among four RESTful Web service providers.

Because pattern learning is relatively time consuming some solutions to speed up this process should be implemented. According to [1] adaptive learning rate algorithm is very effective in accelerating learning performance. In future improvements of the engine this method should be added to the *Pattern Learning* module. In similar manner it is possible to implement adaptive alpha parameter to vary depending on the rate of appearance of new patterns. In result the system would learn more often in the beginning of its life cycle and less frequently when new patterns are rather an exception.

6. SUMMARY

Internet services market is growing very quickly, especially considering the RESTful Web services. There is also no doubt about importance of those services to developers. Unfortunately, they are scattered all over the Internet and difficult to find. Therefore, regardless of the number of available services, they are useless as long as they remain unnoticed by their users. Creating tools that enable finding desired services on the Internet would change this situation.

The goal of this work was to create a uniform and universal RESTful Web service URI structure to allow their identification. Next goal was to propose engine architec-

ture that gives the ability to effectively identify the services according to their structure. Because of the construction of the REST based services URI it was proposed to use of artificial neural networks as a key element to their identification.

To achieve stated goals the study was divided into three chapters. First chapter concerned the research that was necessary to enable the identification of services. The research was grounded on generic URI design and on comprehensive analysis of many examples of different RESTful URIs taken from leading service providers. The result was the uniform and universal structure of REST compliant Web service URI.

The second chapter was devoted to research on engine to properly identify RESTful Web services and create their generic description. Methods used in elaborated engine consisted of several stages related to pattern matching, pattern learning and pattern recognition. It used both Artificial Neural Network (ANN) model with back-propagation algorithm and Adaptive Resonance Theory (ART), together creating unsupervised learning system.

The last chapter contains the results of above studies that were carried out using an engine that applied presented approach. It proved the correctness of introduced REST Web services URI structure and demonstrated the accuracy of presented engine. The chapter was also the prospect of further research on the solution and on improving the quality of the RESTful based services identification methods.

It was shown that solutions proposed in this work are effective in identifying RESTful services. This carries benefits for many programmers and web developers because they can use proposed engine to help them find services they need. Extending these solutions could be also used to create Web services search engine, which would be the most effective tool, in the sense of finding required services, for the production of various types of software.

REFERENCES

- [1] AJITH A., SAJEETH N., *Soft Computing Models for Weather Forecasting*, Journal of Applied Science and Computations, Vol. 11, 2004, 106–117.
- [2] AL MASRI E., MAHMOUD Q. H., *Investigating Web Services on the World Wide Web*, Proceedings of the 17th international conference on WWW, Beijing, ACM Digital Library, 2008, 795–804.
- [3] CARPENTER G. A., GROSSBERG S., *Adaptive Resonance Theory*, In: The Handbook of Brain Theory and Neural Networks, Second Edition, Cambridge, MIT Press, 2003, 87–90.
- [4] PAUTASSO C., *BPEL for REST*, Proceedings of the 6th International Conference on Business Process Management, Milan, Springer, 2008, 278–293.
- [5] RICHARDSON L., RUBY S., *RESTful Web Services: Web Services for the Real World*, Sebastopol, O'Reilly Media, Inc, 2007, 1–54.
- [6] RODRIGUEZ A., *RESTful Web services: The basics*, IBM developerWorks, IBM Corporation, <http://www.ibm.com/developerworks/>, 2008.
- [7] *Web Application Description Language*, <http://www.w3.org/Submission/wadl/>, June 2011.

Anna DOBROWOLSKA*
Wiesław DOBROWOLSKI*

THE USE OF ELECTRONIC QUESTIONNAIRES IN SERVICE QUALITY ASSESSMENT

Services are becoming more and more important in gaining a competitive advantage nowadays. To assess the quality of services, organizations use various methods and techniques based on survey questionnaires and interviews. Traditional, paper-based forms are usually used to record the responses of the interviewees or are distributed to the target group of the survey. The development in information technology offers new possibilities for application of the survey technique in service quality assessment.

The work presents issues connected with the use of electronic questionnaires in organizations for assessing the quality of services. First of all, it shows the advantages and disadvantages of electronic questionnaires, the procedure of electronically-assisted questionnaire surveys, as well as errors and problems associated with preparation of electronic questionnaires and with conducting the surveys, which affect the service quality assessment. The most important diagnosed reasons for problems are incompetent employees and time or budget constraints as well as hardware or software issues.

1. INTRODUCTION

Two types of services are playing currently an increasingly important role in gaining a competitive advantage: services provided to customers and those performed within an organization. To assess the quality of such services, organizations use simple survey techniques, such as survey questionnaires and interviews, but also more complex methods, namely Servqual, CSI, CIT, and WTA, most of which also use the questionnaire survey technique. Questionnaire surveys are conducted more and more frequently with the IT support.

* Wrocław University of Technology, Faculty of Information Technology and Management, Institute of Organization and Management, Smoluchowskiego 25, 50-372 Wrocław, Poland.

The work presents issues connected with the use of electronic questionnaires in organizations for assessing the customer satisfaction. First of all, it shows the advantages and disadvantages of electronic questionnaires, the questionnaire survey procedure, modern web technologies and their use for preparing and conducting interactive questionnaire surveys, as well as problems associated with preparing electronic questionnaires and conducting the service quality assessment.

2. SERVICE QUALITY AND ITS ASSESSMENT

The quality of each service can be defined by adopting the definition given in the ISO 9000:2005 terminology standard as “the degree to which a set of inherent properties of a service meets the requirements”. The primary source of requirements for services is a customer. Service quality assessment requires formulating the inherent features that the customer expects, as well as conducting the studies to determine the degree of fulfillment of the requirements.

Organizations have developed various methods for assessing the quality of services, which usually use two research techniques: interview and questionnaire. Table 1 shows the main methods and research techniques used in service quality assessment.

Table 1. Research techniques in the various methods of service quality assessment

Method of service quality assessment	Main research technique
Analysis of complaints	Questionnaire
CSI – Customer Satisfaction Index	Questionnaire
Servqual	Questionnaire
CIT – Critical Incident Point	Interview
WTA – Work Through Audit	Questionnaire
Mystery Shopping	Direct observation

Source: own study.

3. CLASSICAL QUESTIONNAIRES VS. ELECTRONIC QUESTIONNAIRES

Survey research are defined as: [1, p. 56] “comprises a cross-functional design in relation to which data are collected predominantly by questionnaire or by constructed interview on more than one case (usually quite a lot more than one) and at the single point in time in order to collect a body of quantitative or quantifiable data in connection with two or more variables (usually many more than two) which are then examined to detect patterns of association”.

There are two types of questionnaires, which can be used in surveys, differing in the distribution method:

- paper-based questionnaire – which can be completed in various ways: by mail, by interview, by telephone;
- online web survey – which is distributed by electronic measures (computers, websites).

The purpose and the scope of use of these types of questionnaires in service quality assessment is practically the same, but the method of distribution substantially affects the process of preparing and conducting the surveys.

Surveys conducted via the Internet are becoming increasingly popular and important. Table 2 shows the results of a Google Scholar search for three phrases related to Internet surveys (there is no agreed terminology for surveys conducted online and the three phrases are used rather interchangeably), singly and combined, appearing in articles for each year from 1990 to 2010. Although research based on Google Scholar results has limitations, it can give rough estimates of popularity [4] and it is clear from the Table that there has been a steady and upward rise in the level of interest directed towards surveys using the Internet, whatever term is used to describe them. The corresponding figures in the Table for telephone surveys show slower rise and decreasing interest directed towards such a traditional type of survey. From 2006 the results for online surveys outnumber the ones for telephone surveys each year, reaching two to one in 2009 and are expected to reach three to one in 2011 (6190 to 2140 for Jan-Jul).

Table 2. Results of Google Scholar search on “Internet survey”, “online survey”, “web survey” and “telephone survey”

Year	Internet survey	Online survey	Web survey	Internet, online, and web combined	Telephone survey
1	2	2	4	5	6
1990	4	13	1	18	759
1991	1	8	1	10	794
1992	7	9	3	19	881
1993	4	14	0	18	965
1994	15	18	1	34	1160
1995	12	28	6	46	1220
1996	45	23	16	84	1440
1997	59	55	40	154	1760
1998	103	85	63	251	2050
1999	123	152	100	375	2330
2000	218	261	150	629	2750
2001	291	763	188	1242	3240
2002	366	723	350	1439	3340
2003	485	984	456	1925	3670

2004	589	1440	542	2571	3970
2005	706	2090	722	3518	4900
2006	867	2860	863	4590	4460
2007	1120	4030	1010	6160	4600
2008	1270	5310	1210	7790	4800
2009	1390	6830	1420	9640	4300
2010	1660	7710	1520	10890	3900

Source: own study based on [2].

4. ADVANTAGES AND DISADVANTAGES OF ELECTRONIC SURVEYS

The questionnaire surveys conducted with the use of electronic tools are superior in many respects to traditional (paper-based) questionnaires. First of all, surveys of such a type improve the comfort of respondents, allow getting to large groups of respondents, considerably facilitate the acquisition and processing of the information, and are much cheaper at the same time. From the viewpoint of service quality management it is also important that an electronic survey allows quickly rewarding a respondent for the time spent – in the form of thanks or bonuses (such as games, wallpapers, participation in a contest, promotional coupons), which may strengthen the relation between a customer and the organization.

Despite those advantages, electronic questionnaire surveys have also disadvantages, which are associated primarily with the requirements put on the persons preparing the surveys, as well as those resulting from the limitations on the side of customers (potential respondents). The persons preparing computer-assisted questionnaire surveys should have technical skills and adequate knowledge of IT infrastructure to prepare an online survey. If they do not have such skills, the costs and time of preparing and conducting the surveys may significantly increase. The organization must in such a case hire a specialist or commission a specialized company and purchase appropriate software. The limitations on the side of customers are primarily associated with a lack of appropriate computer hardware among a certain group of customers, to which the surveys are addressed. In such a situation, the assumed large coverage of the surveys is often considerably reduced. Customers can also affect the results of a survey by completing the form several times.

The argument about the anonymity of Internet surveys is also called into question. There is a belief that “nobody is anonymous in the Internet”, which has a practical justification. There are some ways to identify a respondent, for example by identifying the IP address or associating web browser “cookie” with known visitor.

Table 3 shows main advantages of electronic surveys, which do not occur in the surveys conducted with the use of traditional (paper) questionnaires, as well as disadvantages or counterarguments that weaken the positive arguments.

Table 3. Main advantages and disadvantages of an electronic survey

Criterion	Advantage	Disadvantage / counterargument
1	2	3
Comfort of a respondent	Anonymity of surveys.	Anonymity in the Internet is illusory.
	Possibility of an attractive form of presentation of questions – colors, graphics and videos.	Too sophisticated form of presentation may discourage a potential respondent.
	The survey does not discourage a respondent by a large number of visible questions – the questions are placed on consecutive pages.	A respondent may feel discouraged anyway to complete a questionnaire, if it is prepared incompetently or has too many pages.
	An electronic survey allows quickly rewarding a respondent for the time spent – in the form of thanks or bonuses (such as games, wallpapers, participation in a contest, promotional coupons).	Attractive reward may encourage a respondent to complete the questionnaire many times.
Coverage of the survey (reaching the respondents)	Possibility of distributing multilingual versions.	Questionnaires prepared in various languages can be sent in traditional surveys as well.
	Easy and quick way for reaching a large number of respondents – publication on a website, sending the address of the survey by e-mail.	Difficulties with ensuring that a respondent will complete the questionnaire only once.
	Easy and quick way for reaching the target group – a link to the survey (URL address) on a vortal.	Limited coverage – people who do not have a computer or Internet access cannot participate in surveys.
Preparation of surveys	Quick and easy preparing and editing of the survey.	Preparation of the survey is not easy for employees who are not adequately prepared for it.
	Quick and easy conducting of a pilot survey.	
Conducting the surveys	Mandatory questions are easy to impose (it is impossible to go forward without giving a required answer).	Forcing a respondent to answer questions (especially difficult, unclear, time-consuming) may discourage him/her from completing whole survey.
	Notification – immediate notification to the author of the survey of completion of a questionnaire.	
	Possibility of monitoring the responses on a current basis; quickly responding to errors and correcting the survey.	

	Possibility of limiting availability of the questionnaire to time-frame.	
Data processing	Easy data processing – no need to write down the results from paper media, the possibility of immediate statistical processing of data and presenting them in a graphical form (diagrams, tables).	
Economic efficiency	Preparations of the questionnaire, the survey itself, as well as data processing are relatively cheap; low costs of reaching the respondent (delivery and processing of questionnaires).	If external experts and software are used, it may significantly extend the time for preparing the surveys and make them more expensive.
	The process of preparing and conducting questionnaire surveys takes a relatively short time.	

Source: own study.

As it appears from the Table above, the advantages of online surveys outweigh the limitations in their use, although the occurrence of some of the presented arguments in a survey may make it impossible to conduct such a survey.

5. PROBLEMS ASSOCIATED WITH PREPARING ELECTRONIC SURVEYS AND CONDUCTING SERVICE QUALITY ASSESSMENTS

Service quality assessment conducted with the use of an electronically-assisted survey technique requires performing a certain procedure consisting of seven main stages: formulation of the research problem and designing the survey procedure, technical preparation of the electronic questionnaire survey, designing the questionnaire, pilot survey (testing the designed form), customer survey (data acquisition), data analysis (statistical processing) and visualization of the results.

These stages are comprised of fifteen phases: from defining the customers to visualizing the results (presenting them in a graphical form). Figure 1 shows the stages and phases in the model of electronic surveys of service quality.

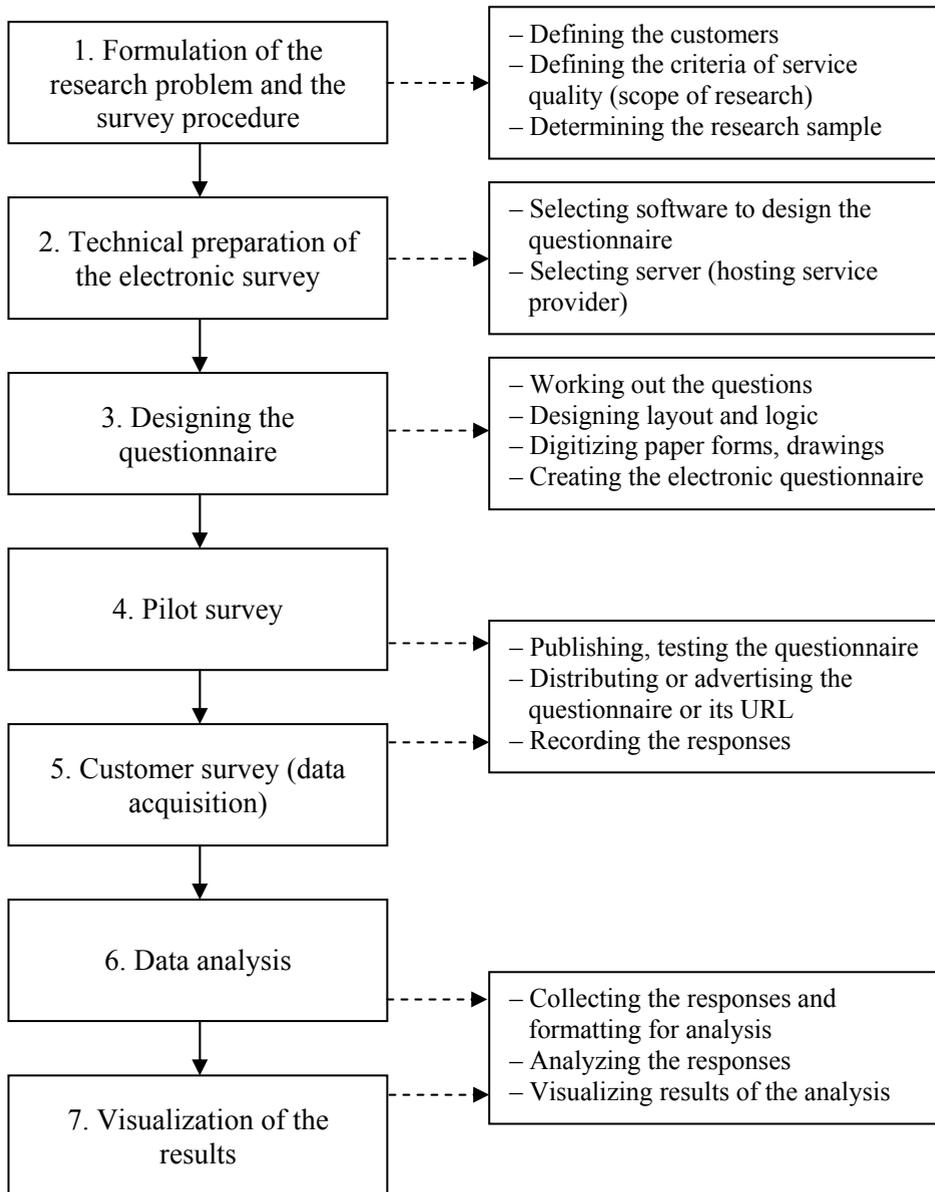


Fig. 1. The model of electronic surveys of service quality.

Source: own study.

The process of preparing electronic surveys may affect the final results of the survey and have negative impact on the service quality assessment.

Table 4 shows typical errors (faults) that may occur in each of the phases of an electronic questionnaire survey, which affect the service quality assessment, and

indicates their reasons and effects. The most important reasons, occurring frequently, are incompetent employees (or an external firm, if the research was outsourced) and time or budget constraints. Other notable reasons – limited in functionality, slow or faulty software, server, network or hosting provider, are of technical nature but can be often attributed to employees as well (care not taken of selection or testing).

Table 4. Typical errors, their causes and effects in each of the phases of an electronic questionnaire survey

Phases of electronic survey	Potential errors	Causes of the errors	Effects of the errors
1	2	3	4
Defining the customers	The target customers are not determined properly.	Lack of marketing expertise.	The questionnaire reaches improper respondent group (other than intended).
Defining the criteria of service quality (scope of research)	Failing to account every important attribute of the service quality characteristics.	Lack of profound understanding of the service or the aspects of service quality.	Incomplete information about the service quality.
Determining the research sample	Improper sample size (not statistically justified).	Insufficient knowledge of statistics or population size.	The results may not be generalized to the whole of the population.
Selecting software to design the questionnaire Selecting server (hosting service provider)	Selecting software, development platform or database with limited functionality or performance. Server (or network) is slow or appends ads to webpages.	Incompetent employees, vendor market not recognized, lack of time or money.	Questionnaire poorly designed or served. Respondent not anonymous, can fill the form many times, is irritated due to ads or waiting.
Working out the questions	Unclear terms, mismatched types (free-text vs. multi-choice), difficult or too many questions.	Incompetent employees, lack of time, lack of understanding of customers.	Respondents are discouraged and give up.
Designing layout and logic	Form too long (no subpages), unreadable (too colorful), no skip-logic, unclear instructions.	Incompetent employees, lack of time.	Respondents are discouraged and give up.

Digitizing paper forms, drawings Creating the electronic questionnaire	The electronic version differs from paper or from the design specifications.	Incompetent employees, limited functionality of software.	Project delayed due to investigations, design / development corrections or replacing software.
Publishing and testing the questionnaire	The questionnaire looks or behaves incorrectly in various browsers.	The electronic form was not tested thoroughly.	Customers may not be able to fill the form.
Distributing or advertising the questionnaire or its URL	Sending mail to wrong recipients. Using black-listed mass mail server.	Wrong addresses in mailing database. Server was or got black-listed.	The questionnaire (or request to fill it) will not reach potential responders.
Recording the responses	The responses are not stored entirely or partially (all or some responses / fields) or were intercepted.	Faulty mechanism of recording results. Insecure transmission or storage of results.	The results are not properly stored. Leakage of sensitive information about organization or customers.
Collecting the responses and formatting for analysis	Collected results are incomplete or not suitable for immediate processing.	Database structure or file format improperly designed.	Analysis delayed due to additional preprocessing required.
Analyzing the responses Visualizing results of the analysis	Results of the survey are questioned, explained doubtfully or are charted unreadably.	Results are improperly processed statistically (lack of expertise in statistics). Careless employees. Limited software.	Hard or impossible interpretation of the results.

Source: own study.

6. SUMMARY

Basing on the analyses presented in the work it can be stated that surveys conducted via the Internet are becoming increasingly popular but have several limitations or pitfalls in their development or implementation procedures. The level of interest directed towards online surveys has recently begun to outnumber the level of interest towards traditional telephone surveys, reaching three to one in 2011. Online surveys are cheaper, easy in distribution, and processing of the obtained results is also

simple. Identified disadvantages of online surveys are outweighed by their advantages, which makes them suitable for measuring the quality of services. However, certain limitations cause that it is not always possible to use online surveys. The customers themselves are the most important issue – they must have a computer and access to IT network, as well as the willingness to use them. Serious potential limitations may be related to the performance of the research procedure itself in the organization. From the first stage of the procedure, i.e. definition of customers, to the final stage, i.e. analysis and visualization of data, various faults may occur making a survey impossible to conduct or causing that the results of the survey will be affected by a considerable error. The most important diagnosed reasons for problems are incompetent employees and time or budget constraints as well as hardware or software issues. It is worth noting that an electronic questionnaire survey is a tool which, apart from customer satisfaction surveys, can be used for a wider range of activities undertaken in an organization in the scope of quality, such as employee satisfaction surveys, online registration of customers, collection of qualitative data concerning the course of business processes.

REFERENCES

- [1] BRYMAN A., BELL E., *Business research methods*, Oxford University Press, New York, 2007.
- [2] FIELDING N., *The SAGE handbook of online research methods*, SAGE Publications Ltd, London, 2008.
- [3] FORZA C., *Survey research in operations management: a process-based perspective*, International Journal of Operation and Production Management, Vol. 22, No 2, 2002. 152–194.
- [4] HARZING A.W., *Google Scholar - a new data source for citation analysis*, http://www.harzing.com/pop_gs.htm, 2008.
- [5] ISO 9000:2005, *Quality management systems – Fundamentals and vocabulary*. International Organization for Standardization.
- [6] SYNODINOS N.E., *The “art” of questionnaire construction: some important considerations for manufacturing studies*, Integrated Manufacturing Systems, Vol. 14, No 3, 2003, 221–237.

PART IV
SECURITY PROTOCOLS, PROCEDURES
AND ALGORITHMS

Jakub FLOTYŃSKI*, Willy PICARD*

TRANSPARENT AUTHORIZATION AND ACCESS CONTROL IN EVENT-BASED OSGI ENVIRONMENTS

Information systems become increasingly more complex and difficult to maintain. The Open Services Gateway Initiative Framework (OSGi) has been proposed as a container-based framework specifically designed for building modular systems. In OSGi, a system consists of applications based on modules.

An OSGi module, referred to as *bundle*, is an application element which has its own lifecycle: it can be remotely installed, started, updated and uninstalled without rebooting the environment. Each bundle may provide access to its functions, enabling the implementation of systems in a service-oriented architecture (SOA). Broadcasting of events among bundles is a communication paradigm supported by OSGi, along with the more traditional communication paradigm based on synchronous service calls among bundles. Event-based communication decouples bundles from applications built on the bundles, and vice-versa. As a consequence, an event-based OSGi environment may be open, allowing new bundles and applications to be added to and removed from the system, under the condition that all the interactions among the bundles and the applications are based on events. The openness of event-based OSGi environments raises the need for means for authorization and access control to bundles, to protect both applications and bundles from unauthorized accesses. In this work, a transparent authorization and access control protocol for event-based OSGi environments is proposed. The proposed protocol is based on an authorization method relying on security policies in which access control may be granted to an event sender using a particular application, for a given activity to be performed by a particular bundle, and with regard to contextual data. The proposed solution has been implemented with XACML as the security policy description language and Sun's XACML Implementation as the security policy engine.

1. INTRODUCTION

Increasing complexity is frequently the reason of difficulties in maintaining and updating information systems. Modularization is one of the possible solutions amend-

* Poznań University of Economics, al. Niepodległości 10 61-875 Poznań, Poland.

ing the software development process. Modular systems are split into many decoupled entities, referred to as modules. Modular systems gains increasing popularity in a variety of domains, such as web information extraction [8], performance analysis [18], or certification systems [9]. Different solutions have been proposed to build modular systems, such as Ada, CORBA, Enterprise Java Beans or Microsoft Windows Foundation. In the Java world, the Open Services Gateway Initiative Framework (OSGi) has been proposed as a framework to build complex modular Java systems. It has been popularized by the Eclipse integrated development environment (IDE), which is based on OSGi. However, the potential applications of OSGi are not limited to the development of IDEs, but it may be also a promising solution for clouds [6] or real-time systems [4].

Each OSGi module, referred to as a *bundle*, has its own lifecycle which is managed by an OSGi container. The bundle lifecycle consists of the installation, startup, update and removal of the bundle. These activities may be performed independently for individual bundles. Bundles can expose their capabilities to other bundles in the OSGi container, enabling the development of complex information systems in a service oriented architecture (SOA).

Functions shared with other bundles may be exposed and invoked in at least three manners. First, exposed methods of a sharing bundle—the provider— may be directly invoked by a recipient bundle—the consumer—if the package containing the classes with the exposed methods is exported by the provider and imported by the consumer. Second, exposed methods may be invoked by a consumer using a specialized invoker object which takes an invoked method name and its parameters as arguments. This way of exposing methods among bundles is more flexible than the first one because of the decoupling provided by the specialized invoker.

The third manner is based on OSGi events broadcasted by a consumer bundle and handled by potentially many providers bundles. Event-based communication is a well-known paradigm with various applications, e.g., graphical user interface development, wireless sensor networks [10], services collaboration [5], or dynamic reconfigurable network environments [2]. Event-based communication is supported by the OSGi Event Admin Service, which is the part of the OSGi Service Platform Compendium Specification [16]. The OSGi Event Admin Service provides support for a publish/subscribe approach to event delivery among bundles. An event consists of a topic for identification and a set of properties. Events may be sent in both a synchronous or asynchronous manner.

The event-based communication is the most convenient one for development of complex SOA systems. Let assume that many logically decoupled bundles provide functions for application development. Event-based communication in an OSGi environment enables many-to-many relations between applications and bundles. New bundles and applications may then be added, modified and removed without any changes of other bundles and applications.

In an open OSGi environment, in which bundles and applications are decoupled and communicate via events, authorization and access control are a major requirement. Without appropriate authorization and access control, a malicious application may compromise bundles by invoking in an uncontrolled manner methods of these bundles. Similarly, a malicious bundle may gather/modify data sent by the users via existing applications. Appropriate authorization and access control methods adapted to open event-based OSGi environments should be developed, and provide transparent authorization and access control means with regards for both bundles and applications.

In this work, a method providing bundles and applications with transparent authorization and access control is proposed. The main contribution of this work is two-fold. First, the concept of the OSGi Security Policy Engine (OSPE) is proposed as a set of OSGi bundles and classes providing means for transparent authorization and access control. Second, a Transparent Authorization and Access Control Protocol (TAAP) for open event-based OSGi environments is presented.

This work is structured as follows: in Section 2 a review of authorization and access control approaches is given with a particular stress on security frameworks for Java applications and security policy description languages. Next, requirements for transparent authorization and access control are presented in Section 3. Both OSPE and TAAP are detailed in Section 4. Next, an illustrative example is presented in Section 5. Finally Section 6 concludes the work and presents future works.

2. AUTHORIZATION AND ACCESS CONTROL

Authentication is the confirmation of a principal identity with a specified or understood level of confidence, e.g., showing an ID card may authenticate John Doe as being John Doe [13]. Authorization is the process of determining whether the particular entity has the right to perform some action on some resource, e.g., showing an driving license may authorize John Doe to drive his car [13]. Access control is the protection of the resources against unauthorized access e.g., John Doe needs the keys of his car to enter it. Authorized system entities relying on security policies are a popular manner to enforce authorization [13].

In this section, security frameworks for Java applications and security policy description languages are discussed.

2.1 SECURITY FRAMEWORKS FOR JAVA APPLICATIONS

Security frameworks enable complex management and enforcement of various security issues, such as authentication, authorization, access control or encryption. The

most popular security frameworks for Java applications are presented below, with a focus on authentication and access control.

Java Authentication and Authorization Service (JAAS) [14] is a security framework providing an API enabling authentication and authorizations of users, within the Java Runtime Environment since version 1.4. Authorization and access control are performed in accordance with the API and rules contained in the configuration file. The main advantages of JAAS are the ease of use and being part of Java. Its main drawback is the lack of roles and its proprietary language for security policy description.

Spring Security (Acegi Security) [1] is a robust security framework written in Java and aimed at Spring, but it is also able to work independently. It offers rich functionality, such as filtering messages delivered to a web application, protection against session hijacking, tag libraries which ease JSP writing. Authorization in Spring Security is supported by Access Control Lists and hierarchical roles. The main advantage of Spring Security is its rich functionality. The main drawback is its proprietary language for security policy description.

OSGi services are responsible for various management aspects of OSGi environments. Three OSGi services are responsible for authorization and access control. The Permission Admin Service and the superseding it Conditional Permission Admin Service [15] provide means for code-based access control (i.e., checking the right of a piece of code to access some resources). The code-based access control is beyond the scope of this work, so it will not be discussed in details. The User Admin Service (UAS) [16] provides means for user-based access control (i.e., checking the right of a given user to access some resources). The User Admin Service provides a role-based approach. The role represents the initiator of the request. Two types of roles are available. The first type is a user with credentials (e.g. password), properties (e.g. address, phone number), and roles. The second type is a group being the aggregation of roles. Such an approach improves the flexibility of management of users rights to perform authorization and access control in an effective way. The main advantage of the OSGi services is the integration with the OSGi environment. Their main disadvantage is its proprietary language for security policy description.

2.2 SECURITY POLICY DESCRIPTION LANGUAGES

Security policy description languages enable the definition of constraints concerning users, resources, and actions. In this subsection two standard languages for authorization and access control are discussed.

The **Security Assertion Markup Language (SAML)** is an XML dialect standardized by OASIS [11]. SAML provides means for authorization and authentication with a strong focus on the single-sign-on technique. SAML requests have the form of documents containing a few elements. The first element—assertion—is a packet of data describing the relation between the user and the resource. Other elements—protocol

and binding—specify a transport layer for sending the message (usually SOAP encapsulated within HTTP request). SAML is a proper solution for authentication but does not enable the definition of complex security policies.

The **eXtensible Access Control Markup Language (XACML)** is a second XML dialect standardized by OASIS [12]. XACML provides means for the definition of security policies, requests and responses. The XACML specification additionally defines the architecture of security policy engine (SPE) which is responsible for the enforcement and management of security policies.

Rules are a basic element of XACML. A rule is the triplet consisting of a subject (a user), an action, and the resource for which the access may be granted. An example of a rule would allow John Doe to open the car. Rules may then be grouped in a policy, together with an associated target specifying a set of subjects requiring to perform actions. For instance, a policy may define a set of rules concerning the actions related to driving of the car (open the car, start the engine, etc.) for a group of users (John Doe and his wife). Finally, policies may be grouped in policy sets, e.g., a policy set may define the rules concerning driving of the car and the potential use of a credit card.

An XACML request consists of a subject (user), a resource, and an action name. Both policy and request may contain also additional attributes which may determine the roles associated with users. An XACML response may have four values: permit, deny, indeterminate (if many policies are suitable for the request) or not applicable (if no policy suits the request).

The XACML implementation includes five components [12]. Dependencies between them are presented in Fig. 1. Interactions between components are synchronous and represented as arrows.

- The **Policy Enforcement Point (PEP)** is responsible for enforcing security policy. It receives requests from the requester and communicates with the Context Handler to obtain the decision about access of the subject to the resource.
- The **Context Handler (CH)** is responsible for transforming the messages received from the PEP into XACML request and sent it to the Policy Decision Point (PDP). The CH is also responsible for transforming the XACML response obtained from the PDP into a response format understandable by the PEP. The request sent to the PDP is built according to the subject, resource and action retrieved from the Policy Information Point.
- The **Policy Information Point (PIP)** is responsible for storing and retrieving subject, resource, action and additional attributes from a database when required by the CH.
- The **Policy Decision Point (PDP)** is responsible for verifying an XACML requests received from the CH against the set of security policies obtained from the Policy Administration Point. The PDP sends back an XACML response to the CH.
- The **Policy Administration Point (PAP)** is responsible for storing and retrieving policy sets from database when required by the PDP.

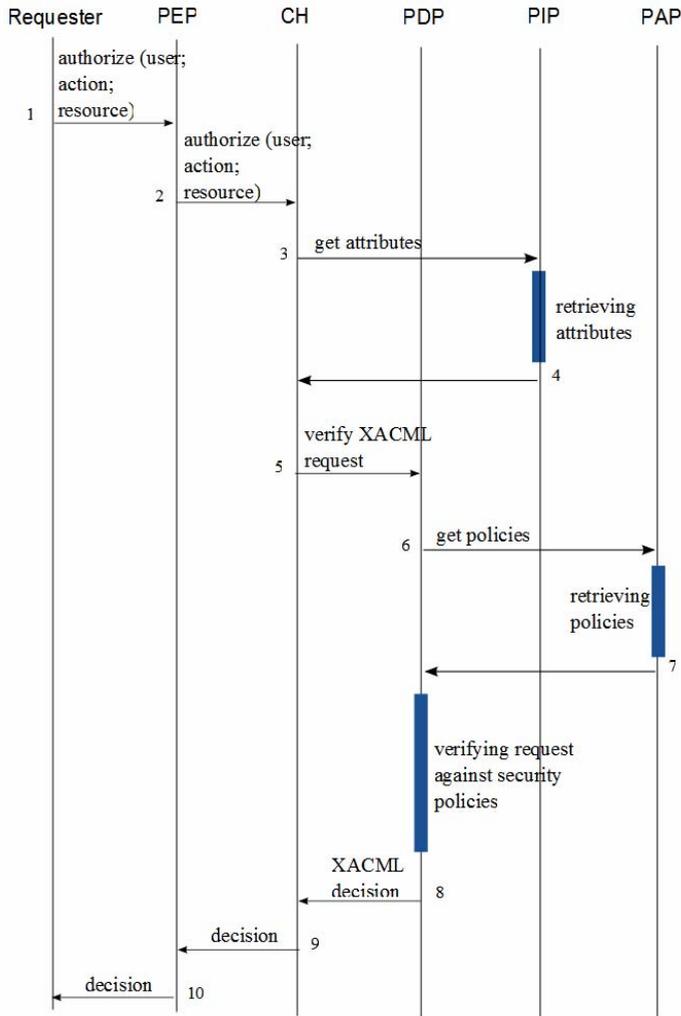


Fig. 1 Interactions among the components of an XACML-compliant security policy engine

The subject (user) begins the authorization and access control process when he tries to perform the action on the resource. A request is sent from the user to the PEP which forwards it to the other components of XACML implementation to receive access decision. The interactions among the components of an XACML-compliant security policy engine (SPE) are illustrated in Fig. 1.

The main advantages of XACML are the possibility to define complex security policies and the fact that XACML is a standard. A main disadvantage of XACML is the relative inefficiency of the XML request verification.

3. REQUIREMENTS FOR A TRANSPARENT AUTHORIZATION AND ACCESS CONTROL METHOD

In this section the assumptions and requirements for a transparent authorization and access control method in event-based OSGi environments are presented.

In the rest of this work, the term *bundle* will refer to bundles which are consuming events, while bundles that are sending events are referred to as *applications*.

The method should satisfy the following requirements:

- 1) Each authorization request specifies a user (a sender of the request), an action (which has to be performed) and a context. The context is a triplet consisting of the application which sends the request, the resource to which the action is related, and the bundle which performs an action.
- 2) The description containing users, actions and contexts has the form of a set of security policies (SP) supporting eventually additional aspects of authorization and access control, e.g., time periods during which access may be granted.
- 3) The security policies are described by standardized languages to be vendor-independent.
- 4) Access rights to resources should be able to be grouped into roles. Several roles may be assigned to a user. Such an approach enables complex descriptions of access rights, while keeping the managing of these rights reasonable.
- 5) The security policy should be managed by a separated entity—the Security Policy Engine (SPE)—that manages the security policies, receives requests from bundles, and grants or prohibits access to resources.
- 6) The method should enable the management of security policies, authorization and access control in the OSGi environment. It should be an extension of the event-based communication.
- 7) The proposed method should be resilient to attempts of unauthorized access to confidential resources by untrusted bundles and protect receiver bundles against dangerous requests from malicious applications.
- 8) The method should be transparent for bundles and applications. Neither applications nor bundles should be modified to take advantage of the authorization and access control method.

4. THE TRANSPARENT AUTHORIZATION AND ACCESS CONTROL METHOD

The proposed method for transparent authorization and access control consists of two elements: the OSGi Security Policy Engine (OSPE) and the Transparent Authorization and Access Control Protocol (TAAP). The OSPE is responsible for managing

the security policies and making decisions about users' authorization and access control. The TAAP enables interaction between applications which would like an action to be performed, a bundle which is able to perform this action, and the OSPE. In this section, both the OSPE and the TAAP are presented.

4.1. THE OSGI SECURITY POLICY ENGINE

The OSGi Security Policy Engine (OSPE) is responsible for the management and the enforcement of security policies. The architecture of the OSPE is an adaptation of the solution proposed in the XACML specification [12] to an OSGi container. The solution is vendor-independent and may be used in conjunction with different languages or APIs. The OSPE physically consists of two groups of components: three OSGi bundles and events implemented as application classes. The first group includes the OPDP, OPIP and OPAP bundles responsible for authorization. The second group contains the OPEP class responsible for access control.

- 1) The **OSGi Policy Enforcement Point (OPEP)** is implemented as the base event class which is inherited by all other events in the system. An event is sent by an application which wants to perform an action. The OPEP is accessed by bundles to obtain information from the request. Before granting access to the information, the OPEP sends a query to the OPDP. Many OPEP are usually simultaneously active in the system.
- 2) The **OSGi Policy Decision Point (OPDP)** is implemented as an OSGi bundle. The OPDP extends the functionality of the PDP and the CH. It is accessible from the OPEP (events) through simple method invocation. It builds requests using attributes retrieved from the event and the OSGi Policy Information Point. Finally the request containing the user's roles, resource and action is created and verified against security policies.
- 3) The **OSGi Policy Information Point (OPIP)** is implemented as an OSGi bundle. It retrieves roles associated with users from a database while requested by the OPDP.
- 4) The **OSGi Policy Administration Point (OPAP)** is implemented as an OSGi bundle. It stores and retrieves sets of security policies in a file system or database while requested by the OPDP.

The interactions among the components of OSPE occur in the way similar to the interactions among the components of a XACML-compliant security policy described in section 2.2 (Fig. 2).

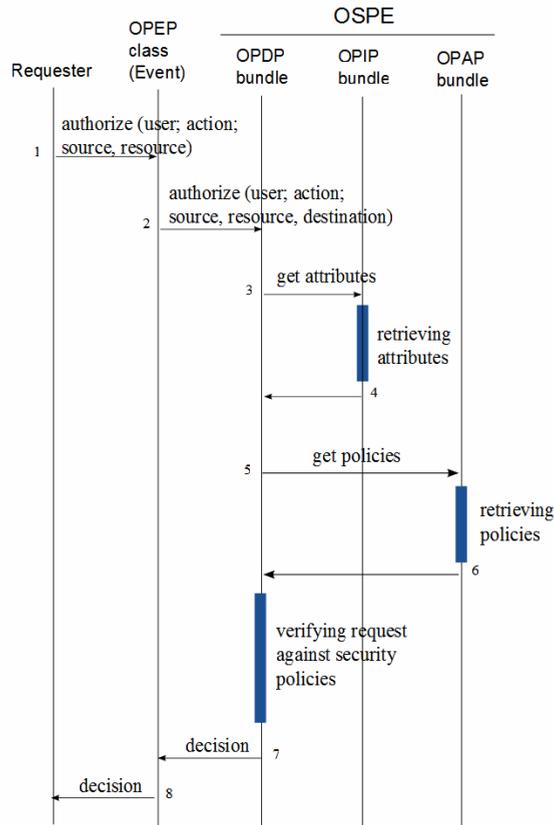


Fig. 2 Interactions among components of the OSGi Security Policy Engine (OSPE)

4.2. THE TRANSPARENT AUTHORIZATION AND ACCESS CONTROL PROTOCOL

The Transparent Authorization and Access Control Protocol (TAAP) defines a sequence of interactions among bundles, applications, and the OSPE to enforce authentication and access control.

The TAAP aims at providing protection in two aspects.

- 1) Confidential data contained within the event is protected against a malicious bundle which may be a receiver of the request. Destination bundle is a part of the context which decides about access to the resource.
- 2) A bundle is protected against access to the event sent by a malicious application. Source application is also a part of the context.

The protocol is illustrated in Fig. 3 for the case when access is granted. The dotted arrowed line symbolizes the asynchronous posting/submission communication based on the Event Admin service. The continuous line symbolizes the synchronous method invocation.

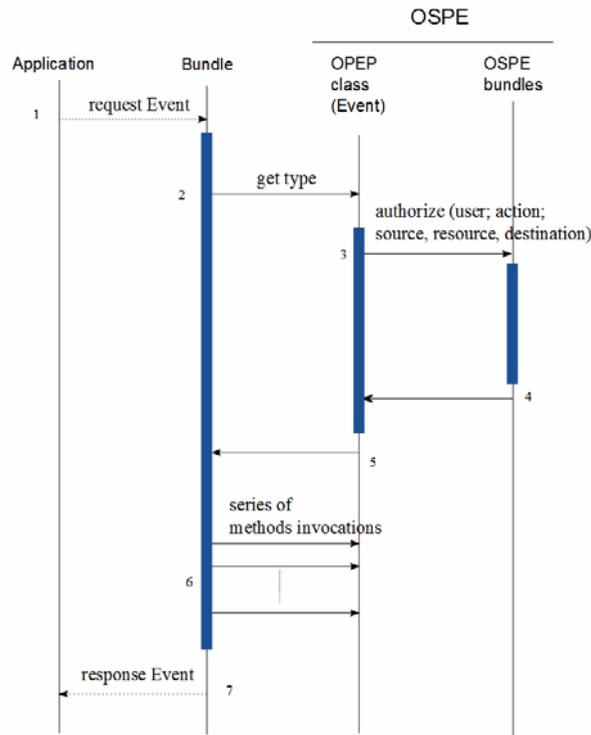


Fig. 3 Scenario of the TAAP interaction

Let assume that an application requires a bundle to perform an action. The application generates and broadcasts an event. Each event consists of an action to be performed, the user that would like the action to be performed, and additional information about a context, as a triplet <source application, resource bounded with the action, destination bundle>. The event transports usually also additional data needed for the execution of the request. The broadcasted event is visible in the OSGi environment for many bundles which may handle it utilizing their event handlers objects.

Each event is also associated with a topic, allowing bundles to ignore events they are not interested in. Therefore the first step taken by a bundle to handle an event is the invocation of the `getType()` method of the event. The result returned by the method would allow the bundle to decide to handle or not the event.

However, the implementation of the `getType()` method is part of the OPEP. Therefore, when a bundle requires the type of the event it is facing, the event sends a query to the OSPE. Next the OSPE makes a decision in the manner described in Section 4.1 and replies to the event. If the access is not granted, the bundle does not have access to the data from the event. If the access is granted, the bundle may access all the methods of the event to obtain the data needed to execute the action.

5. ILLUSTRATIVE EXAMPLE

In this section, an example of authorization and access control procedure based on the OSPE and the TAAP is presented. The example concerns a blog application that can be updated by an author and readers. The author is able to add and remove only posts, while readers may add comments and remove his former comments. The blog system consists of two bundles. The first one—`BlogApplication`—is responsible for communication with the user. This bundle may be accessible for the user, e.g., through a web interface, but the interactions between the user and the system is beyond the scope of this example.

Each method causes generation of a proper event.

The second part of the system—`BlogBundle`—is responsible for blog management. It handles events generated by `BlogApplication` and performs the appropriate methods: `addPost()`, `removePost()`, `addComment()`, `removeComment()` which correspond to the methods of `BlogApplication`. The steps of different interactions between applications and bundles within the system are presented in the following subsections.

5.1. GRANTING ACCESS BETWEEN A TRUSTED BUNDLE AND AN APPLICATION

In this scenario, an author would like to add a new post. The interactions between the author, the blog system and the OSPE is presented below.

- 1) The author invokes the method of `BlogApplication` GUI to add a new post.
- 2) `BlogApplication` generates and broadcast a new event – `AddPostEvent`. The event type has semantics of the required action. It contains also a user, resource which the action is related to (`blog`), context (a pair: `BlogApplication` - source, `blog` - resource) and content of the new post.
- 3) `AddPostEvent` is handled by `BlogBundle` which tries to check the event type. The process of authorization is started.
- 4) Before reply, `AddPostEvent` acts as the OPEP, invokes method of the OPDP to generate the request message and transfer it to the OPDP. The request to the OPDP includes the user, context and action. The context is a triple: `BlogApplication` - source, `blog` - resource, `BlogBundle` - destination.
- 5) The OPDP queries the OPIP about the roles associated with the user.
- 6) The OPIP retrieves the roles associated with the user from database and replies to the OPDP.
- 7) The OPDP combines the user's (author's) roles, resource (`blog`) and action (`add post, BlogBundle`) into request and invokes the method of the OPDP which is responsible for checking user rights giving the message as the parameter.

- 8) The OPDP verifies the request against the security policy and makes a decision about user's access to the resource. The decision is positive (permit) because the author may add the new posts as mentioned before. The decision is sent to the OPDP which forwards it to the OPEP (`AddPostEvent`).
- 9) The session is started and `BlogBundle` retrieves the post content from `AddPostEvent`. `BlogBundle` may then create the new post by executing the method `addPost()` associated with the action "add post".

The scenarios of adding a new comment by a reader, removing a post by the author or a comment by a reader are similar to the scenarios above.

5.2. DENYING ACCESS BETWEEN A TRUSTED BUNDLE AND AN APPLICATION

The scenario presented in the previous subsection finishes with a positive decision. Slightly different from it is the interaction with the negative decision e.g. an attempt of removing a comment by the author or a post by a reader. Suppose a reader tries to remove a post added by the author of the blog. Steps 1-7 are similar as in the previous example. `BlogApplication` generates and broadcasts `RemovePostEvent` which is handled and processed by `BlogBundle`. After the negative authentication, the exception is propagated to the user. It informs the required action is denied.

5.3. DENYING ACCESS TO AN UNTRUSTED BUNDLE

Let assume that a malicious bundle `SnifferBundle` is added to the OSGi container. The `SnifferBundle` wants to listen to all messages sent to the system. The author requires to add a new post, leading to the creation and broadcast of `AddPostEvent` with the content of the post. The event is handled by two bundles: `BlogBundle` and `SnifferBundle`. `BlogBundle` is authorized as previously and accesses data from the event. `SnifferBundle` is not authorized. Setting the value of destination bundle to `SnifferBundle` leads to an access denial by the OSPE (only `BlogBundle` is allowed).

5.4. DENYING ACCESS TO AN UNTRUSTED APPLICATION

Let assume that a malicious bundle `PhishingApplication` is added to the OSGi container. The `PhishingApplication` tries to add a malicious post: a `AddPostEvent` is generated and broadcasted. The event is handled by the `BlogBundle` leading to the execution of the TAAP. Access is denied as the context contains `PhishingApplication` as the source (only `BlogApplication` is allowed).

6. CONCLUSIONS AND FUTURE WORK

In the work, a novel to authorization and access control in open event-based OSGi environments is presented. The proposed solution consists of the OSGi Security Policy Engine and the Transparent Authorization and Access Control Protocol. The presented method is completely transparent to both bundles and applications. An important aspect of the proposed solution is the protection it provides for both bundles and applications with regard to malicious bundles and applications.

The proposed solution has been fully implemented for the Equinox OSGi container. XACML has been chosen as the security policy description language, because it is a vendor independent language for which several implementations are available e.g. JBossXACML [7] and Sun's XACML Implementation [17]. The Sun's XACML Implementation is chosen for the OSPE implementation due to its satisfactory efficiency [3] and the acceptable quality of the available documentation.

Among future works, authentication should be integrated with the proposed method. In the current implementation of the proposed method, applications are sending the user's token as plain string, enabling the possibility of identify spoofing. Similarly, bundles and applications should also be authenticated by the OPEP during the construction of the context. Asymmetric encryption techniques are probably an serious source of inspiration for authentication solutions.

Finally, the scalability of the proposed solution still need to be evaluated. Authorization and access control with the OSPE and the TAAP is probably an expensive task: whenever an event is handled by a bundle, an XACML request has to be built and checked against the security policies. Advanced caching techniques that may probably improve the performance of the proposed protocol still remains to be built.

REFERENCES

- [1] ALEX B., *Acegi Security. Reference documentation*, <http://www.acegisecurity.org/reference.html>, retrieved 07.07.2011
- [2] BENGHAZI K., NOGUERA M., RODRIGUEZ-DOMINIGUEZ C., *Redefinable events for dynamic reconfiguration of communications in ubiquitous computing*, In: Proceedings of the First International Workshop on Data Dissemination for Large Scale Complex Critical Infrastructures, Esposito C., Gokhale A., Cotroneo D., Schmidt D., Valencia, 2010, 17–22
- [3] CRISPO B., TURKMEN F., *Performance Evaluation of XACML PDP Implementation*, In: Proceedings of the 2008 ACM workshop on Secure web services, Damiani E., Proctor S., Alexandria, 2008, 37–44
- [4] DIANES J., DIAZ M., RICHARDSON T., WELLINGS A., *Providing temporal isolation in the OSGi framework*, In: Proceedings of the 7th International Workshop on Java Technologies for Real-Time and Embedded Systems, Higuera-Toledano M., Schoeberl M., Madrid, 2009, 1–10

- [5] ESCHNER L., HINZE A., MICHEL Y, *Event-based communication for location-based service collaboration*, In: Proceedings of the Twentieth Australasian Conference on Australasian Database – Volume 92, Bouguettaya A., Lin X., Wellington, 2009, 125–134
- [6] HAUCK F., ELSHOLZ J., KAPITZA R., NIKOLOV V., SCHMIDT H., *OSGi4C: enabling OSGi for the cloud*, In: Proceedings of the Fourth International ICST Conference on COMMunication System softWAre and middleware, Bosh J., Clarke S., Dublin, 2009, 1–12
- [7] JBOSS, *JBossXACML*, <http://community.jboss.org/wiki/PicketBoxXACMLJBossXACML>,
- [8] retrieved 07.07.2011
- [9] JELINEK I., JIRKOVSKY V., *Proposing of modular system for web information extraction*, In: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, Rachev B, Smrikarov A., Ruse, 2009, 1–4
- [10] KELLY T., *Using software architecture techniques to support the modular certification of safety-critical systems*, Proceedings of the eleventh Australian workshop on Safety critical systems and software - Volume 69, Clant T., Melbourne, 2007, 53–65
- [11] LE H.-C., GUYENNET H., FELEA V., ZERHOUNI N. *A Low Latency MAC Scheme for Event-Driven Wireless Sensor Networks*. In: MSN'07, Mobile Ad-Hoc and Sensor Network, Beijing, China, LNCS 4864, 2007, 291–301
- [12] OASIS, *Security Assertion Markup Language (SAML) V1.1*, 2003, <http://www.oasis-open.org/standards#sam1v2.0>, retrieved 07.07.2011
- [13] OASIS, *eXtensible Access Control Markup Language 2 (XACML) Version 2.0*, 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf,
- [14] retrieved 07.07.2011
- [15] OASIS, *Glossary for the OASIS Security 2 Assertion Markup Language (SAML) V1.1*, <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf>, retrieved 07.07.2011
- [16] ORACLE, *Java Authentication and Authorization Service (JAAS). Reference Guide for the Java 2 SDK, Standard Edition, v 1.4*, <http://download.oracle.com/javase/1.4.2/docs/guide/security/jaas/JAASRefGuide.html>, retrieved 07.07.2011
- [17] OSGi ALLIANCE, *OSGi Service Platform Core Specification*, Release 4, Version 4.3 April 2011, <http://www.osgi.org/download/r4v41/r4.core.pdf>, retrieved 07.07.2011
- [18] OSGi ALLIANCE, *OSGi Service Platform Service Compendium*, Release 4, Version 4.3, 2011, <http://www.osgi.org/download/r4-v4.2-cmpn-draft-20090310.pdf>, retrieved 07.07.2011
- [19] SUN, *Sun's XACML Implementation*, <http://sunxacml.sourceforge.net>, retrieved 07.07.2011
- [20] THIELE L., WANDELE E., *Abstracting functionality for modular performance analysis of hard real-time systems*, In: Proceedings of the 2005 Asia and South Pacific Design Automation Conference, Tang T., Shanghai, 2005, 697–702

Tymon MARCHWICKI, Grzegorz KOŁACZEK*

SECURITY LEVEL EVALUATION AND ANOMALY DETECTION IN DATA EXCHANGE FOR SERVICE ORIENTED SYSTEM

This work presents Security Analyzer anomaly detection tool and its application to security level monitoring of service oriented systems. Security Analyzer monitors the status of the system's security and identifies risk factors which can be used for improvement of security management of the system. This anomaly detection tool offers two classes of services. The first one is a monitoring service which examines selected characteristics of the data exchange system (e.g., patterns of communication between the services, the intensity of the communication, the time of services implementation, etc.). The second class of the provided services analyses the results obtained from the monitoring service and detects and identifies security incidents. Security Analyzer provides its functionality using dedicated software agents. There are three classes of agents: agents specialized in detection of the global anomalies in the system's behaviour, agents specialized in an anomaly detection of a separate service behaviour and client behaviour agents.

1. SECURITY LEVEL EVALUATION

Security Analyzer is a type of security monitor for the service-oriented systems ([8]) which evaluates the level of security on the basis of detection of the anomalous traffic patterns in network traffic generated by the services available in the system. In this work we consider a service-oriented data exchange system. The following Fig. 1 presents the relationships between service oriented system and Security Analyzer module. Abbreviation SLA ([8]) stands for Service Level Agreement, which is the contract between a service user and a service provider that specifies conditions of service usage.

* Wroclaw University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wroclaw, Poland.

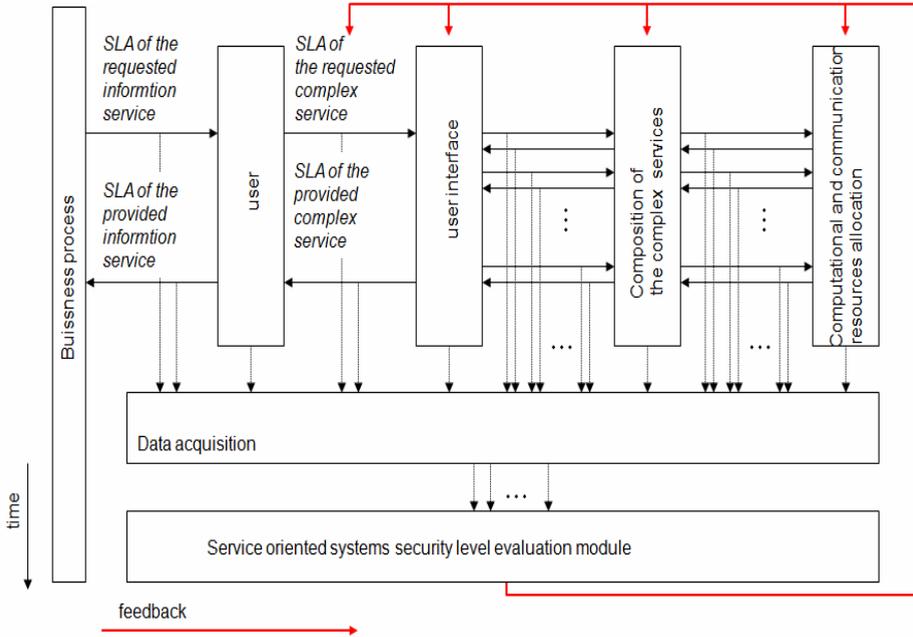


Fig. 1. Security level evaluation in service oriented systems

The generic system’s architecture that is monitored with Security Analyzer is depicted in Fig. 2. It consists of services that communicate with each other. These services are observed by local agents – each agent is responsible for monitoring communication of one service [2,5,6]. Agents collect services communication information and then send it asynchronously to the main database. The main database is used by Security Analyzer tool to detect anomalies.

In the figure, circles represent atomic services (there are 10 atomic services here denoted as: $S1, S2, \dots, S10$). Each atomic service is monitored by one agent. Agents are denoted as: $A1, A2, \dots, A10$. Solid lines represent communication between services (for simplicity, we neglect the communication direction here as well as communication time, we also neglect number of messages exchanged, i.e. communication strength). Dashed lines represent asynchronous database updates made by local agents. Each agent continuously collects its service communication information and updates the main database with some fixed frequency (e.g. every 5 minutes).

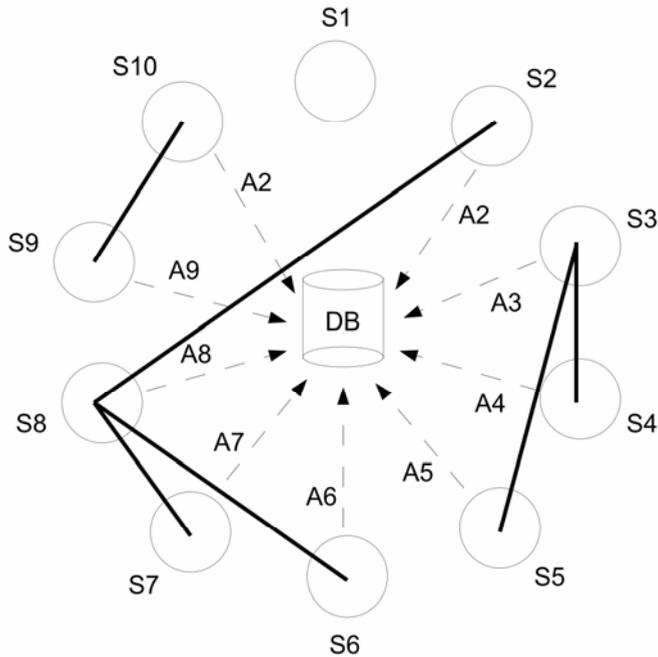


Fig. 2. Monitored system's architecture

1.1. ANOMALY DETECTION

Anomaly detection is an alternative to misuse detection technique that tries to identify security related events [1,2]. Anomaly detection uses a model of normal system behaviour and tries to detect deviations and abnormalities. E.g., raise an alarm when a statistically rare event(s) occurs. As a precise set of attacks against service oriented data exchange system's security cannot be defined by now (most of the security problems will appear during tests and normal operation of the system) anomaly detection is the most convenient way of protecting its security. Anomaly detection can potentially detect unknown attacks and security problems without necessity to know all details constituting so called 'attack signature'. In this way, anomaly detection may support the service oriented systems security from the early beginning [3,4].

A set of anomaly detection modules will automate monitoring of the service oriented data exchange system security. The anomaly detection algorithms to be implemented focus among others on profiles and anomaly detection in:

- resources consumption
- user behaviour
- global system behaviour characteristic.

2. SECURITY ANALYZER ARCHITECTURE

The general architecture of Security Analyzer consists of four basic functional blocks:

- analysis of the security requirements in the profile of the service requests
- security analysis of a complex service execution plan
- security analysis of the implementation of the service after the allocation of computing and communications resources
- comparison of planned and achieved level of security.

The task carried out by a block analysing security requirements in the profile of service requests performs comparison of the security level requirements defined by the user and saved in the service request. The effect of the evaluation is a numeric value representing the level of the discrepancies between the security level desired by user and security level defined in the request for a service. The security analysis of a complex service execution plan block shall evaluate the level of security available for a particular execution plan of a complex service. The block of security level analysis of services implementation evaluates the security of the service-oriented system after the allocation of computing and communication resources, using in this analysis all available information about computing and communication resources consumption level within the system being monitored. The last block is responsible for the assessment of the accordance level of the security level expected by the user and the actual one. Within this generic model we can distinguish two classes of services for each of its functional block: monitoring services (that collects information of system services) and analysing services (that operate on monitoring services data and analyse its results).

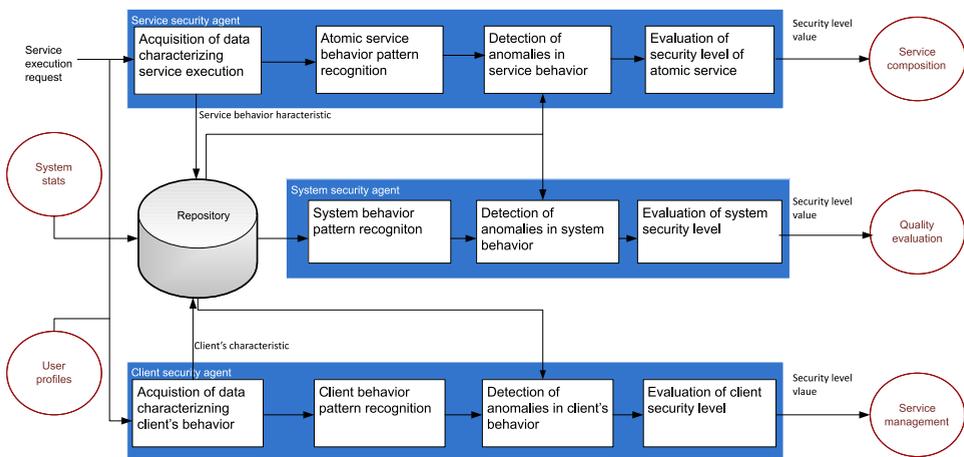


Fig. 3. Generic architecture of the security monitoring module in service oriented systems

The generic idea of Security Analyzer has been implemented using multi-agent approach. The proposed multi-agent architecture depicted in Fig.3. introduces the following types of agents:

- service security level monitoring agents
- system security level agents
- client behaviour security level agents.

In the lowest layer of the multi-agent system we get a collection of specialized agents which are directly responsible for the monitoring and preliminary processing of data derived from the service-oriented system. This layer is the layer of the service security level agents. The second layer – the agents responsible for the evaluation of security level of the system are used to aggregate the data from the agents of the lower layer. In result, the level of the whole service-oriented system security can be evaluated. The next layer is the layer of the agents which estimate the security level of the system clients. The task of the client security agent is to combine the data provided by lower layers agents and to automatically discover user behaviour patterns so to be able to detect their abnormal behaviour. The first type of agents are the monitoring part of Security Analyzer. The last two agent types analyse results of the former.

3. SECURITY ANALYZER IMPLEMENTATION AND APPLICATION

The current implementation of Security Analyzer introduced in previous section includes it's first functional block: service requests profile block and comprises the following components (Fig 4).

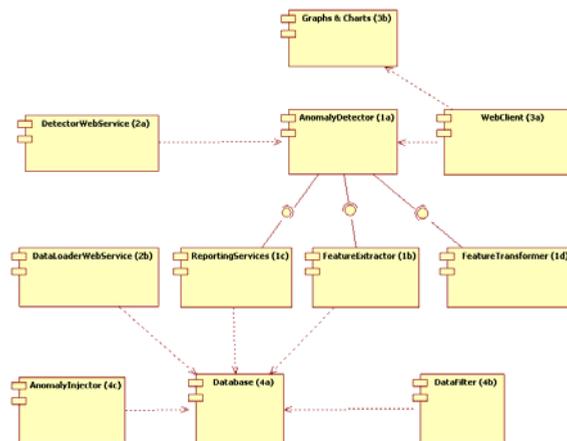


Fig. 4. Detailed Security Analyzer architecture

- 1) Core components – algorithms and calculations
 - a) *Anomaly Detector* – component responsible for detecting security violations
 - b) *Feature Extractor* – extracts features for the needs of *Anomaly Detector*
 - c) *Reporting Services* – enables reporting *Anomaly Detector* calculations
 - d) *Feature Transformer* – enables representing extracted features in different metric space using multidimensional scaling techniques
- 2) Web Services components
 - a) *Anomaly Detector Web Service* – returns anomaly measure calculated by *Anomaly Detector* component, can be used by external services
 - b) *Data Loader Web Service* – loads data into the database, can be used by service monitoring agents
- 3) Graphical User Interface components
 - a) *Web Client* – uses other components (i.a. *Anomaly Detector*) to present the information to the user
 - b) *Graphs & Charts* – enables displaying visual information about the system in the form of graphs and charts
- 4) Data Layer components
 - a) *Database* – stores all communication data that was loaded into the system by monitoring agents
 - b) *Data Filter* – enables reducing the amount of information needed to be processed by randomly selecting database records
 - c) *Anomaly Injector* – enables to modify the communication data information loaded into the system

3.1. SPECIFICATION OF THE MAIN SECURITY ANALYZER COMPONENTS

AnomalyDetector must operate on feature values. These values are supplied by the feature extractor. Each feature extractor is the implementation of the *IFeatureExtractor* interface. There can be many possible features. An example is *InWeightFeature*, the total number of messages received. Extractor of this feature connects to the database and uses *SQL* query to get the information that it needs. Feature values are the aggregated characteristics of the system's state. Because events that occur in the system are saved into the database, *SQL* queries can be easily used for aggregating this data. The application of Security Analyzer to service oriented system monitoring requires observation and analysis of the following features (each calculated for a separate node¹):

- *InWeight* (input number of messages received from all other nodes)
- *OutWeight* (output number of messages sent to all other nodes)

¹ Nodes are system services. Compare with Fig. 2.

- *InDegree* (number of in-neighbours)
- *OutDegree* (number of out-neighbours)
- *AvgInWeight* (average number of received messages per one neighbour)
- *AvgOutWeight* (average number of sent messages per one neighbour)

The set of possible features can be different for different systems as well as it can grow as the system evolves and as some new attributes are being added to the database [1].

Our *Anomaly Detector* is an abstract class. One of its implementations is *EigenAnomaly Detector* that detects anomalies by calculating eigen values of the correlation matrix between services' feature values. It contains two methods: one for determining the current state of the system and the second for calculating the previous state of the system (as a mean from n last states). Anomaly measure is obtained by comparing these two values.

Reporting Services component uses an external database to report the results of *Anomaly Detector's* analysis. It uses specialized classes (also called *services* here), which enable saving values calculated by algorithm into the database.

Feature Transformer is used to change the input Euclidian space (with observations' distances measured using standard Euclidian metric) into the estimated geodesic space (with geodesic distances between observations' points). What is more, *Feature Transformer* enables converting standard values (integer numbers) into ranked values (sorted from minimum to maximum, with ranks assigned for subsequent sorted values).

3.2. SECURITY ANALYZER APPLICATION TO DATA EXCHANGE SYSTEM MONITORING

Security Analyzer is a generic solution for security level monitoring of service oriented distributed systems. Security Analyzer has been designed in accordance with the Security as a Service (SaS) and Software as a Service (SaaS) paradigms. Security Analyzer provides security level evaluation and system monitoring services which help to identify risk factors and can be used for improvement of security management of any system. Especially, this tool can be applied as a monitoring service which examines behaviour of the distributed data exchange system (e.g., patterns of communication between the services, the intensity of the communication, the time of services implementation, etc.).

To verify the soundness of the proposed approach to security level evaluation problem, Security Analyzer has been applied to monitor security level of distributed data exchange system which general architecture has been presented in Fig. 5. The main components and functionalities of this system are as follows:

- institution (*INST*) with a public access as well as a clerk interface to be used for document routing and as an on-line directory

- mechanisms to exchange structured electronic documents
- a set of access points (*AP*) per distinct country
- the compulsory use of a central node (*CN*) for inter-country distribution of messages

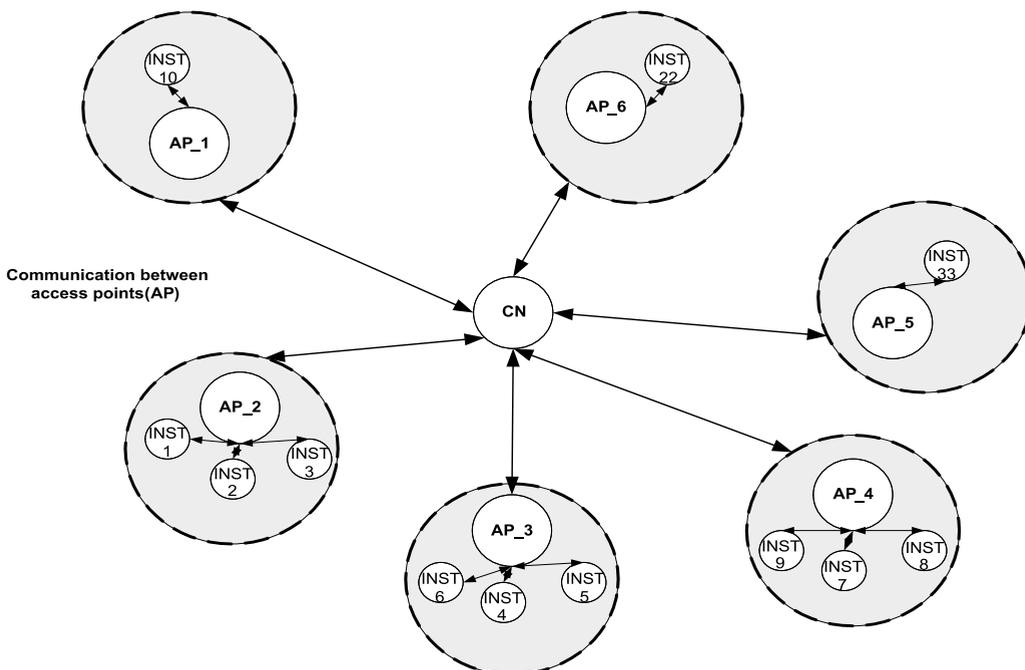


Fig. 5. Architecture of distributed data exchange system

As Security Analyzer is an independent security service it can be applied for monitoring and analysing the behaviour of any selected element of the distributed data exchange system depicted in Fig. 5. So, Security Analyzer can be integrated with the system at the local level (*INST* nodes), midlevel (*AP* nodes) and at central node level (*CN*). At the current stage of Security Analyzer development, it has been applied to support security at the central node level where information about all exchanged data has been aggregated. The idea of Security Analyzer application to monitoring and securing central node has been described in Fig. 6.

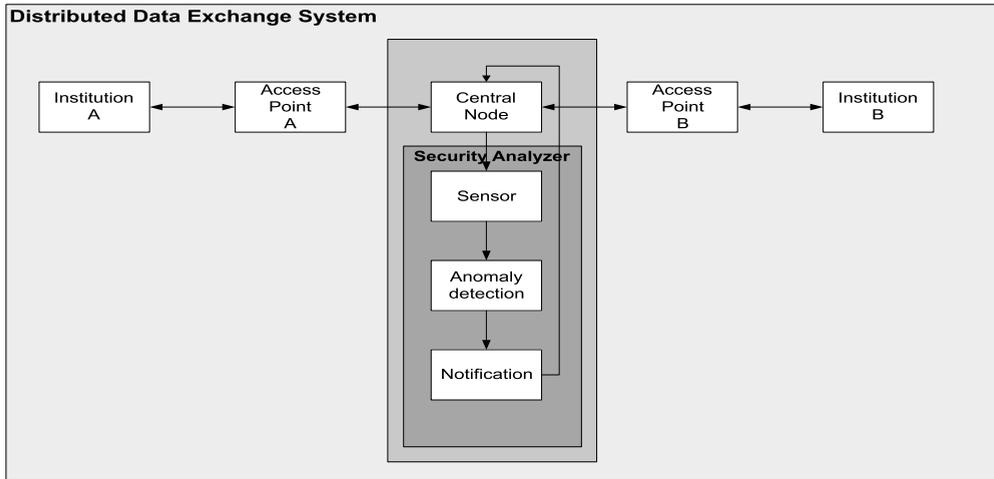


Fig. 6. Security Analyzer applied to monitoring central node security level

The implemented method is applicable for monitoring the system of communicating nodes at the global level, which means that only changes in a behaviour of multiple nodes (considered together) will be detected. This is connected with the method nature. Determining the current state is based on finding the principal eigenvector of the correlation matrix c , where c_{ij} means the correlation between nodes i and j for a fixed feature and fixed window the correlation is calculated for. When each feature is taken separately, we calculate M vectors for M features. By the principal vector for each feature we mean the vector which corresponds to the largest eigen value λ_{\max} , which is the most informative vector and can be considered as a system's state. We detect anomaly when all principal vectors representing the current state change comparing to the previous state vectors for all the features we consider. This happens when multiple nodes change their behaviour. To determine the strength of the change we measure the distances between each of pair of the state vectors (current state vector and previous state vector).

States are calculated in discrete moments of time and the interval length between those moments corresponds to the detector reaction time.

4. CONCLUSIONS

The work presents the architecture and the application of Security Analyzer. Security Analyzer is a Security as a Service based approach which can be used to provide security level evaluation of any service oriented system. The main features and benefits of proposed solution are as follows: a dedicated interceptor activation to monitor

the activity of the service, constant monitoring and identification of communication patterns and generated network traffic between services, recording of selected characteristics of the services activity to the local repository, detection of the abnormal services behaviour, evaluation and recording of the services' security level in the local repository, detection of the abnormal system behaviour, evaluation and recording of the system's security level in the local repository, the abnormal behaviour detection of the clients of the service oriented system, the identification of user, network and access point profiles of behaviour.

ACKNOWLEDGEMENT

This work has been partially supported by the Polish Ministry of Science and Higher Education within the European Regional Development Fund, Grant No. POIG.01.03.01-00-008/08.

REFERENCES

- [1] CHANDOLA V., BENERJEE A. KUMAR V. *Anomaly Detection, A Survey*, 2007.
- [2] KOŁACZEK G., *Multiagent Security Evaluation Framework for Service Oriented Architecture Systems*. In International Conference on Knowledge Engineering Systems (KES'2009), Lecture Notes in Artificial Intelligence (LNAI), Santiago, Chile, September 2009, pp. 30–37
- [3] BURGESS M., *An Approach to Understanding Policy Based on Autonomy and Voluntary Cooperation*, DSOM (2005) pp. 97–108.
- [4] BURGESS M., *Two Dimensional Time-Series for Anomaly Detection and Regulation in Adaptive Systems*, DSOM (2002), pp.169–180.
- [5] GORODETSKI V., KARSAEV O., KHABALOV A. I., KOTENKO, L. POPYACK, SKORMIN V., *Agent based model of Computer Network Security System: A Case Study*, In: Proceedings of International Workshop Mathematical Methods, Models and Architectures for Computer Network Security, Lecture Notes in Computer Science, vol. 2052, Springer Verlag, Berlin Heidelberg New York, pp. 39–50, 2001.
- [6] HWANG K., LIU H., CHEN Y., *Cooperative Anomaly and Intrusion Detection for Alert Correlation in Networked Computing Systems*, Technical Report, USC Internet and Grid Computing Lab (TR 2004-16), 2004.
- [7] LAKHINA A., CROVELLA M., DIOT C., *Characterization of Network-Wide Anomalies in Traffic Flows*, Technical Report BUCS-2004-020, Boston University, <http://citeseer.ist.psu.edu/715839.html>, 2004.
- [8] AMMELLER D., FRANCH X., *Service level agreement monitor (SALMon)*, In 2008 7th International Conference on Composition-Based Software Systems, pp. 224–7, 2008.

Grzegorz GÓRSKI*

NOVEL MULTISTAGE AUTHORIZATION PROTOCOL

The work presents typical architecture of user authentication protocol which usually consists of three independent protocol components i.e. authorization client, pass-through authenticator and authentication server. The authorization server is the central and crucial component that ought to be particularly protected. It stores users' profiles which contain for example user identifiers, verification methods, lists of available resources, periods of time when the user can login. Another component of mentioned above architecture is an pass-through authenticator that is service delivered by border network access point. The work describes the novel authentication protocol based on the same architecture. Due to applied multistage authorization sequence during each couple of protocol components must identify each other. The new solution can ensure transmitted data protection even in case of enemy pass-through authenticator component interception. The order of identity verifications between protocol components assumes that first authorization client and pass-through authenticator have to mutually prove their identities. This feature of new authentication protocol significantly reduce even potential security vulnerability for DDoS attacks against authorization sever component.

1. INTRODUCTION

One of the most important challenges for network protocols designers is how to achieve controlled access to intranet resources for external users as well as internal ones. Due to necessity of constant information exchanges corporate network architectures must remain open for external traffic. There are many ways of granting access to protected applications, servers i.e. local solutions – based for example on directory services – and global ones which can rely on IEEE 802.1x standard [1] implementations or NAC (Network Access Control) products [4,5]. Regardless of applied access control method obtained level of security is closely associated with

* Koszalin University of Technology, Faculty of Electronics and Computer Science, Koszalin, Poland.

quality of user authentication process. It usually consist of two basic phases i.e. user identity verification stage and following authorization one when user's access privileges are checked up.

The typical architecture of user authentication protocol usually consists of three independent protocol components i.e. authorization client, pass-through authenticator and authentication server [6]. The authorization server is the central and critical component that ought to be particularly protected. It stores users' profiles which store for example user identifiers, verification methods, lists of available resources, periods of time when the user can login. Another component of described architecture is an pass-through authenticator that is service delivered by border network access point. This functionality could be supported by switches, routers, wireless network access points, VPN gateways and firewalls. The last component is an authorization client i.e. dedicated software that usually is installed on PC. The main task of authorization client is to establish communication with the nearest border network access unit with pass-through authenticator service. After successfully fulfilled authorization procedure the client should be connected to appropriate networks resources via protected channel. The obtained level of security strongly depends on the way how protocol components exchange confidential information, generate and refresh encryption session keys.

The following part of the work presents security analysis of typical authentication architecture. The derived conclusions describe security vulnerabilities and their primitive sources. One of them is operation mode of typical pass-through authenticator component that in fact is the protocol stack gateway.

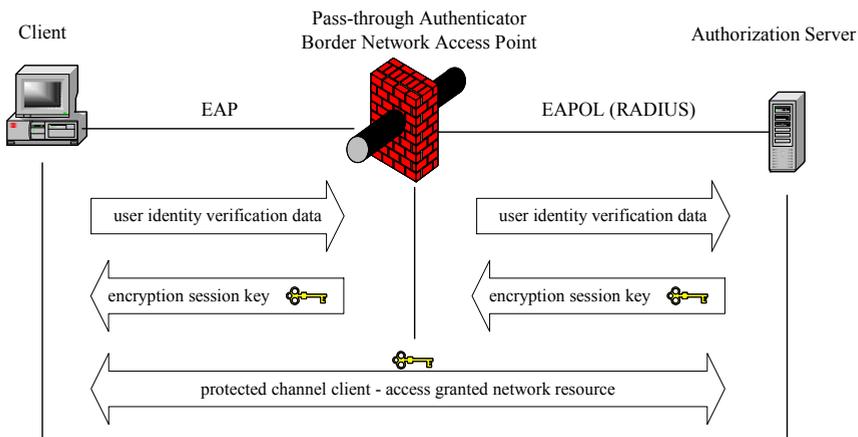


Fig. 1. Typical architecture of authentication protocol

The border network component gains access to all information transmitted between the authorization client and the authorization server inclusive user login and

password. However in applied authentication architectures the protocol stack gateway must process all data and changes message syntax for example in 802.1x standard solutions pass-through authenticator component translates from EAP (Extensible Authentication Protocol) messages into EAPOL (*EAP Over LAN*) ones [2,3].

If an intruder intercepted the border network access point with pass-through authenticator component the attacker gains full access to all data transited between authentication server and clients connected to the border unit. The other potential treat of typical user authentication architecture is the quality of identity verification level. Typical protocols offer single-side first degree verification level [8]. Applied authentication architectures assume that only external components i.e. authorization client and server identify each other. It cause that the authorization server have to process all user login request and exposes the central security component for DDoS attacks which can hardly be blocked with SPF (*Statefull Packet Filtering*) on firewalls.

2. MULTISTAGE AUTHORIZATION PROTOCOL

Considering all limitations and potential treats the work presents the novel authentication protocol based on the same logical architecture which consists of three described above components. Contrary to typical solutions the new algorithm assumes multistage authorization sequence between each couple of protocol instances. They must identify each other because the new solution must ensure transmitted data protection even in case of enemy pass-through authenticator component interception. This additional requirement makes direct impact on complexity of whole user authentication procedure. Instead of single identity verification between the client and the authorization server the described protocol requires triple longer verification track. The order of identity verifications between protocol components assumes that first the authorization client and the pass-through authenticator have to mutually prove their identities. In second step the pass-through authenticator and the authorization server must identify each other. And finally at the beginning of the last stage - based on client's profile received from just verified pass-through authenticator - the authorization server establishes communication with the client and begins the last component authentication. Thanks to much more complex procedure it is not possible that anonymous users' requests could be processed by the crucial security component i.e. the authorization server. The communication between the client and authorization sever is established in case of succeeded mutual identity verification between the client and the pass-through authenticator as well as between the pass-through authenticator and the authorization sever. The first failure in

any verifying couple of protocol components breaks down the whole authentication scheme.

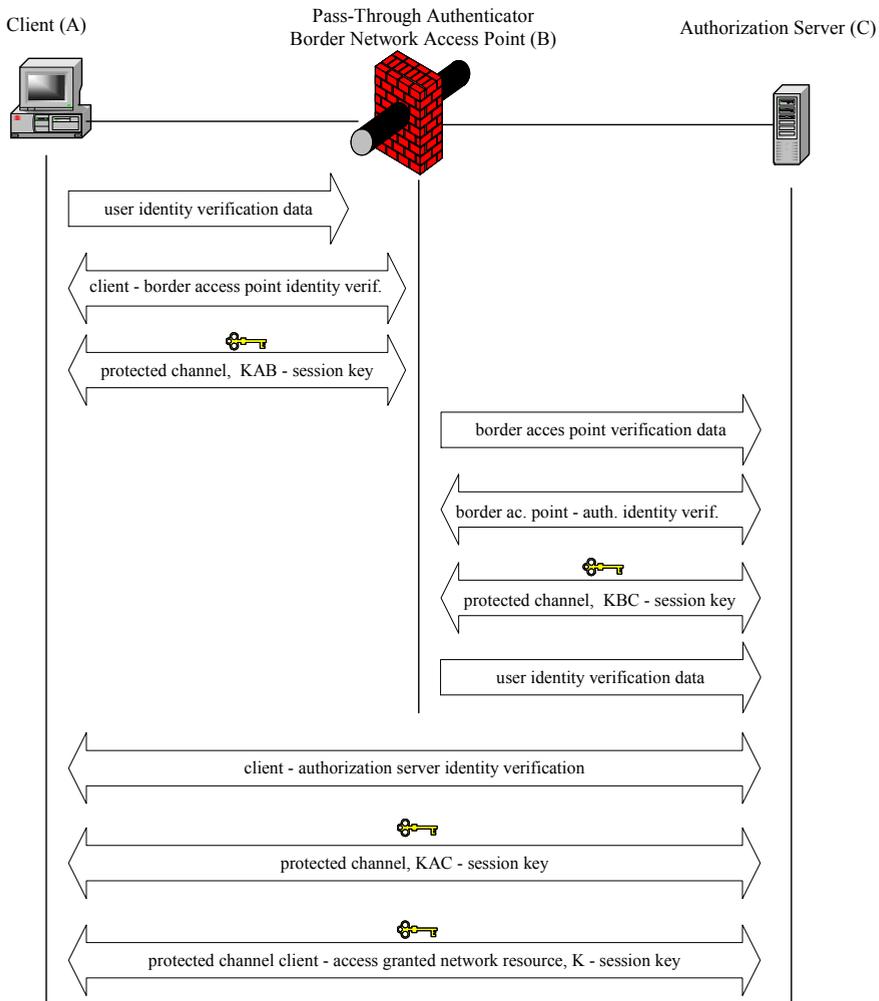


Fig. 2. Multistage authentication protocol

If the whole authentication sequence succeeds the client and the network are mutually identified. Additionally all component couples, which previously took part in procedure, exclusively share appropriate session keys. They are used to establish protected channels for secret data exchange. In the next step which is referred to as authorization user's access privileges are checked up. The authorization server generates the session key K and protected channel between the client and required network resource is es-

tablished. The other session keys defined during identity verification steps are used during periodic session key K refreshment procedure to ensure long lasting level of security.

In the diagram Fig. 2 there is a sequence of three mutual identity verification between protocol components. This fundamental activities require that all protocol components hold valid public key certificates. Such a structure is made out by Certificate Authority that is regarded as TTP (*Third Trusted Part*) by all networks components and users. Additionally presented in the work authentication procedures use TVPs (*Time Variant Parameters*) such as time stamps, session identifiers, random numbers to achieve required level of security. These parameters assure that messages that are sent between protocols sides are up-to-date and unrepeatable. Identity verification protocols must provide assumed level of security. Therefore adequate choice of protocol components identifier and TVPs is the most important issue in protocol design phase [8]. Random numbers and component private keys should be appropriately generated and stored by hardware signing modules.

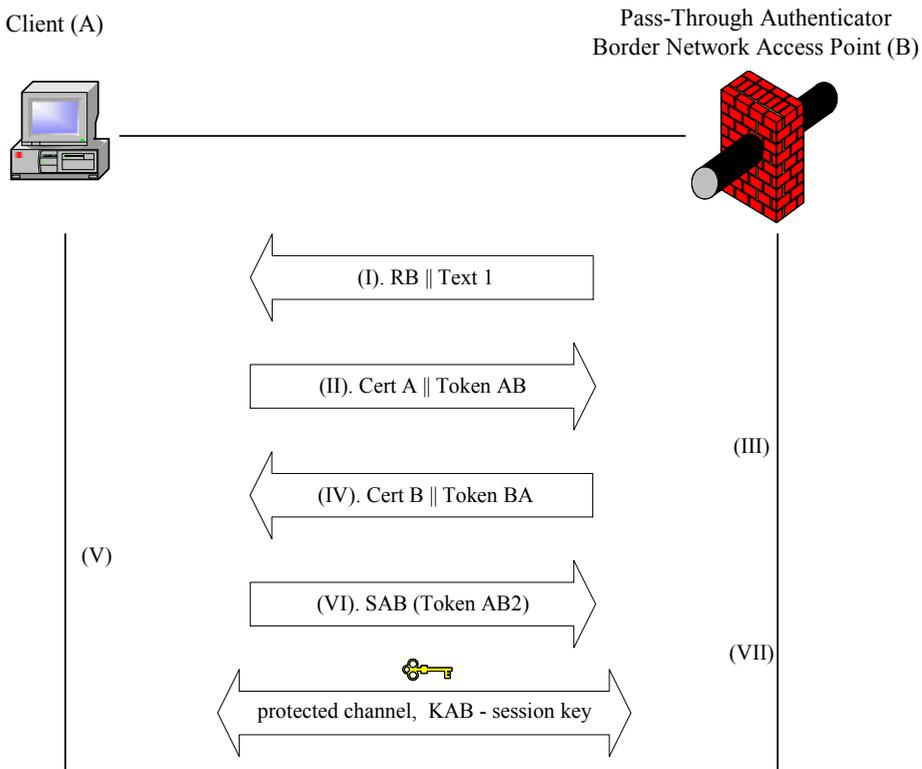


Fig. 3. 4-way handshake mutual authentication algorithm with protected session key deliver

Contrary to standard 802.1x which does not define its own identity verification protocol, there is a dedicated authentication algorithm for presented multistage authorization protocol. The algorithm uses 4-way handshake scheme that ensures mutual second degree identity verification [8] and protected session key delivery.

The algorithm steps are presented below and are numbered consecutively.

- I. The component **B** sends to the component **A** one-time random value (nonce, TVP) **RB** and string **Text1** optionally that could be the unique **B** identifier.
Text1 - the unique component **B** identifier
- II. The component **A** sends to the component **B** message **TokenAB** and its public key certificate **CertA** optionally.

$$\mathbf{TokenAB} = \mathbf{RA} \parallel \mathbf{RB} \parallel \mathbf{B} \parallel \mathbf{Text3} \parallel \mathbf{sSA}(\mathbf{RA} \parallel \mathbf{RB} \parallel \mathbf{B} \parallel \mathbf{Text2}) \quad (1)$$

RA – one-time random value (nonce, TVP) generated by the component **A**

B – the unique component **B** identifier

A – the unique component **A** identifier

Text3 – it could also be the unique component **A** identifier

$$\mathbf{Text2} = \mathbf{A} \parallel \mathbf{B} \quad (2)$$

sSA(RA || RB || B || Text2) – the electronic signature of the junction of parameters: **RA**, **RB**, **B** and finally the string **Text2**

- III. The component **B** checks if it owns the component's **A** public key. If not **B** validates received certificate **CertA** and withdraws the component's **A** public key from the structure. Next **B** tests if received in step (II) one-time random value **RB** is equal to the one generated and sent in step (I). Finally **B** validates electronic signature received as the part of **TokenAB**.
- IV. The component **B** sends to the component **A** message **TokenBA** and its public key certificate **CertB** optionally.

$$\mathbf{TokenBA} = \mathbf{RB} \parallel \mathbf{RA} \parallel \mathbf{A} \parallel \mathbf{Text5} \parallel \mathbf{eSa}(\mathbf{RB} \parallel \mathbf{RA} \parallel \mathbf{A} \parallel \mathbf{Text4}) \quad (3)$$

Text5 – it could be full certification path for **CertB**

eSa(RB || RA || A || Text4) – the junction of parameters **RB**, **RA**, **A** and string **Text4** whole encrypted with component's **A** public key. Only the component **A** which holds the appropriate private key will be able to decipher the message.

$$\mathbf{Text4} = \mathbf{KAB} \parallel \mathbf{RB2} \parallel \mathbf{Text6} \parallel \mathbf{sSB}(\mathbf{Text7}) \quad (4)$$

KAB – secret session key exclusively shared by component **A** and **B**

RB2 – next one-time random value (nonce, TVP) generated by the component **B** which is not equal to nonce **RB**

Text6 – it could be the unique session **A-B** identifier. It prevents the protocol from cross-session attacks, when the intruder injects to the communication stream real frames originated from previous i.e. not current sessions.

sSB(Text7) – the electronic signature of parameter **Text7**

$$\mathbf{Text7} = \mathbf{RA} \parallel \mathbf{RB2} \parallel \mathbf{KAB} \parallel \mathbf{Text6} \quad (5)$$

V. The component **A** initiates the similar procedure to the one from step (III). **A** checks if it owns the component's **B** public key. If not **A** validates received certificate **CertB** and withdraws the component's **B** public key from the structure. Next **A** tests if received in step (IV) one-time random value **RA** is equal to the one generated and sent in step (II). Then **A** decrypts the part of **TokenBA** i.e. **eSa(RB || RA || A || Text4)** using its own private key. Once more **A** compares one-time random value **RA** from plain and decrypted part of **TokenBA**. If they are equal the component **A** starts to process string **Text4**. First the electronic signature of parameters **Text7** is validated. In case of successful verification the component **A** assumes that the **KAB** is the current session key and from the moment all data sent to component **B** will be encrypted using symmetric cipher algorithm with **KAB** as a key.

VI. The component **A** sends to the component **B** message **TokenAB2** encrypted with symmetric cipher algorithm with **KAB** as the session key.

SAB(TokenAB2) – encrypted **TokenAB2**

$$\mathbf{TokenAB2} = \mathbf{RB2} \parallel \mathbf{KAB} \parallel \mathbf{Text6} \quad (6)$$

VII. The component **B** decrypts **TokenAB2** received from **A** with the current session key **KAB**. Then **B** checks if one-time random value **RB2**, session key **KAB** and string **Text6** are equal to the ones sent by **B** in step (IV). The equivalence of appropriate parameters successfully finishes the identity verification procedure.

3. SECURITY ANALYSIS OF MULTISTAGE AUTHORIZATION PROTOCOL

Opposite to typical authentication protocols the described algorithm ensures high level of data protection thanks to more complicated initial verification procedure. All component partake in it mutually identify each other with the highest possible degree what was formally proved [7]. The more complex authorization protocol is the more time it lasts, however first implementations confirmed that whole user login procedure usually does not take more than 10 seconds.

After the last identity verification between the client and the authorization server all transmitted data is encrypted regardless which network resource is concerned. It requires completely no participation of intermediary units even the verified pass-through authenticator. The intruder who took unauthorized control over boarder network access point with the pass-through authentication service has no chance to gain access to transmitted data. The only possible attack scenario is to break off the protected communication channels. Such abnormal protocol behavior is immediately discover because either the one side wait to long for the next message or two many received messages are corrupted. The protocol assumes that message integrity is controlled with unique session identifier that is always placed at the end of each message. If the message payload after decryption does not contain at its end the appropriate TVP the whole packet is dropped. The described type of attack can bring down only the single boarder network access point and users attached with it. However the most crucial component of network security architecture i.e. authorization server remains operating and protected.

The next attack scenario examines potential security vulnerability for DDoS attacks against authorization server component. The author assumed that the intruder controls the group of computers with authorization client software installed. The goal is to break down the chosen pass-through authentication unit and than utilize all authorization server resources for processing fake users' access requests. If the central component of authentication architecture was practically switched off it would be possibility to gain unauthorized access to any given network resource. The applied multistage sequence assumes that before user's login request is processed by the authorization server, components identity had to be mutually proved between the client and the pass-through authenticator as well as the pass-through authenticator and the authorization server. It is not possible that anonymous users' requests can utilize resources of the central security component. However the intruder could use identity of real user with valid public key certificate. The attacker must have gained access to private key of the compromised user. The potential risk of the mentioned above attack could be significantly reduced by defining the maximum concurrent connections limit. If it is exceeded all next request are dropped without processing. This procedure is very effective as far as numerical complexity is con-

cerned. In addition the client identity could appear on black list of users whom the access control system does not deal with. This ban could be permanent or for defined period of time that makes DDoS attack inefficient.

However the probability of successful DDoS attack against the presented authorization protocol is not equal to zero. Such a case could happen if the intruder would have to gain access to so many different private keys of real users that it could create the number of concurrent session (number of clients multiplied by maximum concurrent connections per user limit) which could utilize all of the authorized server resources. Considering the fact that if even single private key is revealed the corresponding public key certificate is immediately revoked (appears on CRL lists published by Certificate Authority) the probability of DDoS attack with large number of intercepted user identities is extremely low. Additionally the applied multistage authentication sequence results that even attack co-ordination with intercepted border network access point would not lengthen the odds on successful DDoS attack.

4. CONCLUSION

The novel authorization protocol operates on the same architecture as solutions relied 802.1x standard or NAC systems. However it assumes more complex three-stage authentication sequence during all communicating protocol components identify each other. The longer procedure results that appropriate components exclusively share their session keys. There is no more necessity that pass-through authenticator instance must process secret data received from the client and send them to the authorization server. After successful client authorization described protocol handles secret session key delivery separately for each network resource to which access was granted. The data are sent directly from the authorization server to the client and it requires completely no participation of intermediary units even the verified pass-through authenticator. This protocol feature results that presented solution ensures transmitted data protection even in case of enemy pass-through authenticator component interception. The intruder - who is not entitled to control the border network access point – receive only encrypted data streams and there is no efficient type of attack except for breaking off the whole communication. Even than none of transmitted information is revealed.

The applied multistage sequence assumes that before user's login request is processed by the authorization server, components identity had to be mutually proved between the client and the pass-through authenticator as well as the pass-through authenticator and the authorization server. It is not possible that anonymous users' requests can utilize resources of the central security component. This feature of new

authentication protocol significantly reduce even potential security vulnerability for DDoS attacks against authorization server component.

REFERENCES

- [1] IEEE Standards for LAN and MAN: *Port-Based Network Access Control. IEEE Std P802.1X-2001*, 14th June 2001.
- [2] RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*, June 2000
- [3] RFC 3580 *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*, September 2003.
- [4] Net Implementation *White Paper - Cisco Network Admission Control*, In: Help Customers Improve Security. Cisco Systems Inc., <http://www.cisco.com/go/nac/>, 2008.
- [5] *Standardizing Network Access Control: TNC and Microsoft NAP to Interoperate*, Trusted Computing Group, Microsoft Corporation, May 2007.
- [6] Górski G., *Bezpieczne sieci VLAN z autoryzacją dostępu oparte na certyfikatach atrybutów*, In: Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej Nr 2 Seria: Technologie Informacyjne vol.3, 2004, 51–58.
- [7] Górski G., *Dynamic Virtual LANs with Strong User Identity Verification Based on Public Key Infrastructure*, In: Information Systems Architecture and Technology – Information Systems and Computer Networks, Biblioteka Informatyki Szkół Wyższych, Oficyna Wydawnicza Politechniki Wrocławskiej, 2008, 175–185.
- [8] Górski G., *Projektowanie i analiza formalna algorytmu silnego uwierzytelniania użytkowników dla dynamicznych sieci VLAN*, In: Przegląd Telekomunikacyjny – Rocznik LXXXII i Wiadomości Telekomunikacyjne – Rocznik LXXVIII – No. 8–9/2009, 2009, 1748–1757.

Ewa IDZIKOWSKA*

ERRORS DETECTION IN S-BOXES OF HASH FUNCTION HAF-256

HaF-256 (Hash Function) is a dedicated cryptographic hash function considered for verification of the integrity of data. It is suitable for both software and hardware implementation. HaF has an iterative structure. This implies that even a single transient error at any stage of the computation of hash value results in a large number of errors in the final hash value. Hence, detection of errors becomes a key design issue. This is important for all cryptographic chips, produced in large quantities (cheap solutions are needed). Concurrent checking of cryptographic chips has also a great potential for detecting faults injected into a cryptographic chip to break the key [2].

A parity based Concurrent Error Detection (CED) approach to protect the S-boxes core of function HaF is presented in this work. We provide simulation results related to the error detection capability of the proposed approach and compare these results with the results for the simplest form of error detection scheme with 100% hardware redundancy, i.e. with Duplication With Comparison (DWC) scheme. Simulation results for single and multiple as well as transient and permanent faults are presented.

1. INTRODUCTION

A hash function H is a transformation that takes an input m and returns a fixed-size string, which is called the hash value h (that is, $h = H(m)$). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography, the hash functions are usually chosen to have some additional properties, e.g. $H(m)$ must be relatively easy to compute for any given m , one-way and collision-free [7]. Very important role of a cryptographic hash function is in the provision of message integrity checks, digital signatures, password storage and verification etc.

*Poznań University of Technology, pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland.

Crypto-systems are computationally complex, and in order to satisfy the high throughput requirements of many applications, they are often implemented by means of VLSI (Very Large Scale Integration) devices. The high complexity of such implementations raises concerns regarding their reliability. There is a need to develop methodologies and techniques for designing robust cryptographic systems, and to protect them against both accidental faults and intentional intrusions and attacks, in particular those based on the malicious injection of faults into the device for the purpose of extracting the secret information [2, 3]. If an attacker deliberately generates a glitch attack, causing a flip-flop state to change or corrupt data values when they are transferred from one digest operation to another, even a single fault can result in multiple errors in the hash value computed. The severity of the problem necessitates detection of errors a key design issue. A digest round consists of several operations. Errors can creep in at any of these operations and can affect one or several bits at any of the operations in a digest round. As HaF is considered for use in security services, concurrent error detection is very important.

CED has certain associated penalties such as hardware cost and the performance degradation due to interaction between the circuit and the detection logic, which need to be considered while designing the error detection circuit. The design goal of the CED is to achieve 100% error detection with minimal penalty. CED techniques involve redundancy in the form of hardware, time or information. A CED circuit based on hardware redundancy can for example duplicate the complete circuit. It means that hardware overhead is more than 100%. In time redundancy, the same hardware is used to perform both the normal computation and re-computation using the same input data. The advantage of this technique is that it uses minimum hardware. The drawbacks of this technique are that it entails $\geq 100\%$ time overhead and it can only detect transient faults. In information redundancy technique, data are appended with additional bits and a coding scheme is used to detect errors. Coding techniques marginally increase the hardware as well as performance overhead. Combinations of the above techniques are also employed to minimize the overhead for CED. For example the hardware cost of time redundancy techniques is generally smaller than that of hardware redundancy, but the systems performance is directly affected.

In this work we focus on hardware redundancy CED techniques. We present a parity based Concurrent Error Detection approach to protect the S-boxes core of function HaF-256. We compare simulation results of the error detection capability of this method with the results of the simplest form of error detection scheme with 100% hardware redundancy, i.e. with DWC (Duplication With Comparison) scheme.

This work is organized as follows. Sec. 2 presents family HaF of hash functions. The nonlinear elements of HaF, S-boxes, are described in Sec. 3. Possible faults and faults models in S-boxes are shown in Sec. 4. Parity bits CED scheme for functions S is described in Sec.5. Simulation results are presented in Sec. 6. They are compared with fault detection possibilities of DWC scheme.

2. FAMILY HAF OF HASH FUNCTIONS

The family HaF is formed of three hash functions: HaF-256, HaF-512 and HaF-1024, producing hash values of the length equal to 256, 512 and 1024 bits, respectively. The general model for HaF is presented on Fig. 1 [1].

After formatting the original message m we have the message M and this message is divided on blocks M_0, M_1, \dots, M_{k-1} . Each block M_i is processed with the salt s by the iterative compression function φ . After processing all blocks we receive the hash value $h(m) = H_k$ as the result.

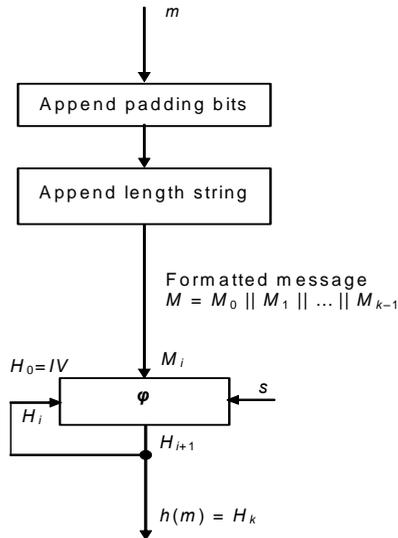


Fig. 1. Model for hash function HaF [1]

The original message m has to be formatted before hash value computation begins. The length of formatted message should be a multiple of $16n$ bits. It means, that the length of the input block equals $16n$ bits, where n is a parameter depending of the hash value we want to obtain. The parameter n equals 16, 32 and 64 bits for HaF-256, HaF-512 and HaF-1024 respectively. The parameter n indicates in fact the length of the working variable used in the step function.

The block M_i is processed in two rounds. Each round consists of 16 steps. The step $\#j$ is indicated by the integer $j \in \{0, 1, \dots, 15\}$. Operations creating a function F_j are performed in each step. Substitution S is one of these operations. It consists of four S-boxes S_0, S_1, S_2 and S_3 , of dimension 16×16 each. We consider HaF-256 function in this work.

3. S-BOXES

S-box is a substitution function and is used to obscure the relationship between the plaintext and the ciphertext. It is an important element of cryptographic algorithm and it should possess some properties, which make linear and differential cryptanalysis as difficult as possible. HaF S-box is a function that takes 16 input bits and produces 16 output bits – it is an 16×16 S-box.

HaF S-box has been generated using the multiplicative inverse procedure similar to AES (Advanced Encryption Standard) with a randomly chosen primitive polynomial defining the Galois field. Nonlinearity of this S-box is 32510 and its nonlinear degree is 15. Sixteen Boolean functions that constitute this S-box have nonlinearities equal to 32510 or 32512 and are all of degree 15 [1].

A 16×16 S-box can be stored as a table containing 65536 values indexed by an input of the S-box function, i.e., x_1, x_2, \dots, x_{16} . The table stores S-box outputs (16 bits: $f_1(x_1, x_2, \dots, x_{16}), f_2(x_1, x_2, \dots, x_{16}), \dots, f_{16}(x_1, x_2, \dots, x_{16})$).

4. FAULTS MODELS

Fault attack tries to modify the functioning of the computing device (typically a smart card) in order to retrieve the secret key. More precisely, the attacker induces a fault during cryptographic computations. The faulty results are used for key recovery. The feasibility of a fault attack or at least its efficiency depends on the exact capabilities of the attacker and the type of faults he can induce.

In our considerations we use a realistic fault model wherein either transient or permanent faults are induced randomly into the device. We consider single and multiple faults. Faults are modelled as an 16-bit error vector $E = \{e_{15}, \dots, e_i, \dots, e_1, e_0\}$, where $e_i \in \{0, 1\}$ and $e_i = 1$ indicates that bit i is faulty. The number of ones in this vector is equal to the number of inserted faults. Fault simulations were performed for two kinds of fault models. In one model a fault flips the bit, and in the other model (stuck-at-0/1) the bit takes a constant value 0 or 1.

Let $X = \{x_{15}, \dots, x_i, x_0\}$ be an error-free vector of bits, and $Y = \{y_{15}, \dots, y_i, y_0\}$ be an output vector. Vector $Xe = \{xe_{15}, \dots, xe_i, xe_0\}$ is an erroneous vector [4, 5]:

- $xe_i = x_i \oplus e_i$ — if the fault flips the bit,
- $xe_i = x_i + e_i$ — for stuck-at-1 fault,
- $xe_i = x_i \times (\text{not } e_i)$ — for stuck-at-0 fault,

where: \oplus - xor, + - or, \times - and.

Error injection points are the following:

- S-box inputs,
- S-box outputs,

Errors are observable on the S-box outputs [6].

5. PARITY-BASED ERROR DETECTION SCHEME

We present parity based Concurrent Error Detection approach to protect the S-boxes core of function HaF. Results of this method will be compared in next section with results of DWC scheme.

The S-box is usually implemented as a 65536x16 bits of memory, consisting of a data storage section and an address decoding circuit. We propose replacing the 65536x16 bits memory that stores the S-box values with 65536x18 bits memory to increase the dependability and detect input, output and internal memory errors of the S-box. One of these two additional bits is a parity bit generated for incoming data bits. The other one is a parity bit generated for outgoing data (Fig. 2).

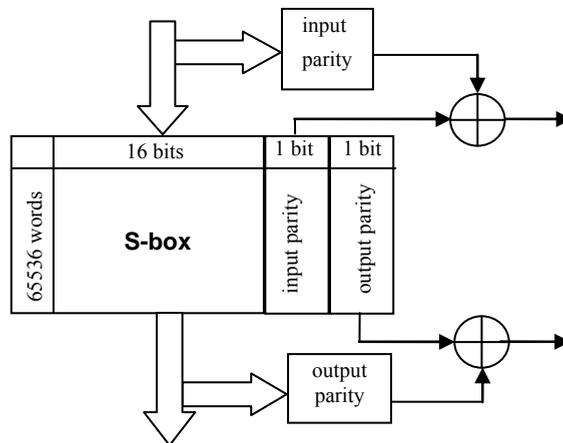


Fig. 2. Parity based CED

Thus solution demands only 131072 additional bits of memory (redundancy is equal to 12,5%) for one S-box, simple combinational circuit for calculation of parity and two comparators (Fig. 2. and Fig.3.). Capability of single and multiple, transient and permanent fault detection using this scheme of parity prediction is presented in Sec. 6.

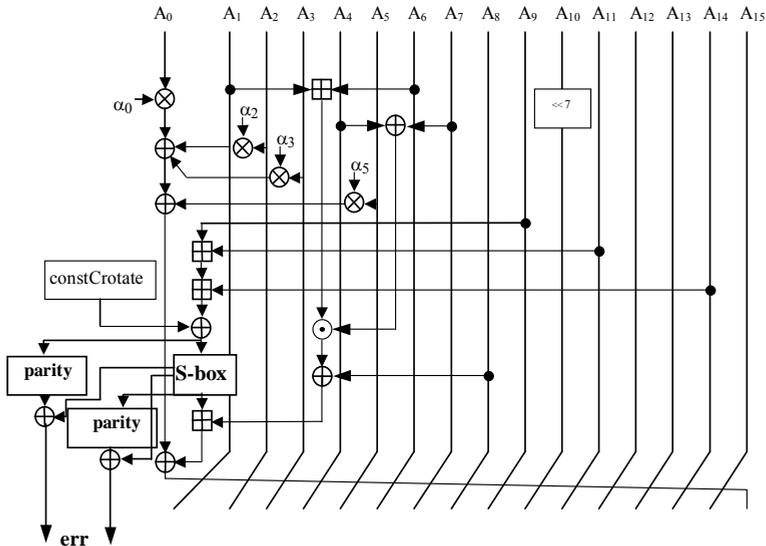


Fig. 3. Step function F_j with CED elements.

6. SIMULATION RESULTS

We provide simulation results related to the fault coverage of the proposed approach in this section. We present simulation results on the vulnerability of this scheme for fault models from Section 4. The faults were injected into inputs, outputs and internal memory of the S-box. We consider random faults, in the sense that the faulty value is assumed to be random and uniformly distributed. In order to measure the detection capability of the proposed architecture (Fig. 3.) we used VHDL hardware description language and the VHDL simulator, Active-HDL by Aldec. The VHDL model of the S-box has been modified with the faults. The output signals have been compared to correct signals. In this way, the obtained fault coverage gives a measure of the error detection capability [6].

In this experiment we focused on transient and permanent, single and multiple stuck-at faults and bit flips faults. The obtained faults coverage is shown in Fig. 4. for permanent faults and in Fig. 5. for transient faults. Single permanent stuck-at-0/1 faults are detected by proposed parity bit CED in 57.7% for stuck-at-0 and 54.9% for stuck-at-1, transient faults in 55.5% for stuck-at-0 and 53.5% for stuck-at-1. Single permanent bit flip faults are detected in 100%, transient in 94.5%. Figures 4. and 5.

shown also dependence of error detection probability on the number of injected faults for parity bit CED. Multiple bit flip faults are detected always.

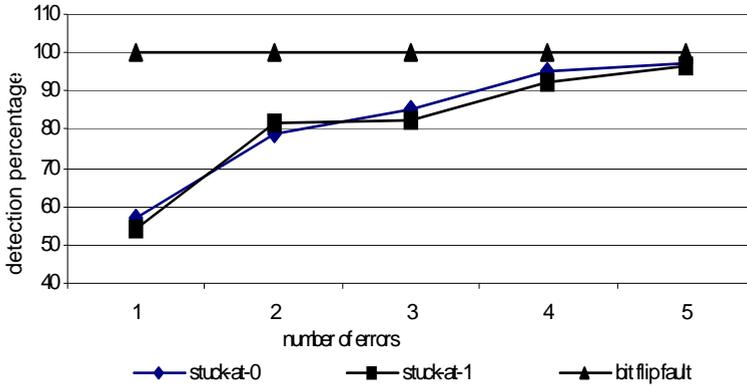


Fig 4. Probability of permanent error detection using parity bits CED

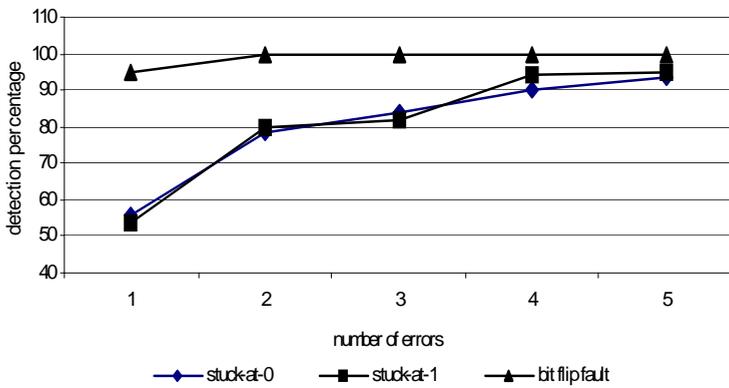


Fig 5. Probability of transient error detection using parity bits CED

A comparison between proposed parity bit solution and the architecture with 100% hardware redundancy (with Duplication With Comparison scheme) is shown in Table 1. In this table we compare probability of permanent error detection for stuck-at-0/1 and bit flip fault models, for different number of injected faults (from 1 to 5).

Table 1. Probability of permanent error detection using Parity Bits and DWC schemes

Fault type	Stuck-at-0					Stuck-at-1					Bit flip fault				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Number of errors															
Detection percentage Parity Bits	57.7	79.5	85.2	95.3	97.4	54.9	85.1	85.4	94.4	96.5	10.0	10.0	10.0	10.0	10.0
Detection percentage DWC	75.1	86.6	94.8	98.4	99.5	76.4	88.7	95.3	98.2	99.9	10.0	10.0	10.0	10.0	10.0

Table 2. Probability of transient error detection with Parity Bits and DWC schemes

Fault type	Stuck-at-0					Stuck-at-1					Bit flip fault				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Number of errors															
Detection percentage Parity Bits	57.1	79.1	85.1	92.8	95.3	50.5	79.6	81.8	93.3	95.1	94.5	10.0	10.0	10.0	10.0
Detection percentage DWC	66.5	84.8	90.1	96.3	99.6	88.8	92.4	95.7	97.2	99.3	100.0	10.0	10.0	10.0	10.0

Detection percentage is better for DWC scheme, first for all for single stuck-at-0/1 error. A difference between DWC and parity bit scheme reaches more than 20% however this difference is much smaller for multiple errors and decreases to 3-5%. There is no difference between detection percentage for these two error detection schemas for bit flip errors. All permanent errors are detected.

Transient stuck-at-0/1 faults are also better detected by DWC scheme (Table 2.) For multiple bit flip there is no difference between detection percentage for this two schemas. Single bit flip error is detected in 94.5% by parity bits and in 100% by DWC. We can say that detection percentage for both methods is the same for bit flip errors despite the difference in hardware overhead. Redundancy for parity bits scheme is about 12.5%

and for DWC scheme is more than 100%. Comparison of detection percentage of stuck-at-0/1 and bit flip errors, regardless of number of injected faults, for parity bits and DWC schemas is shown in Fig. 6. for permanent errors and in Fig. 7. for transient errors.

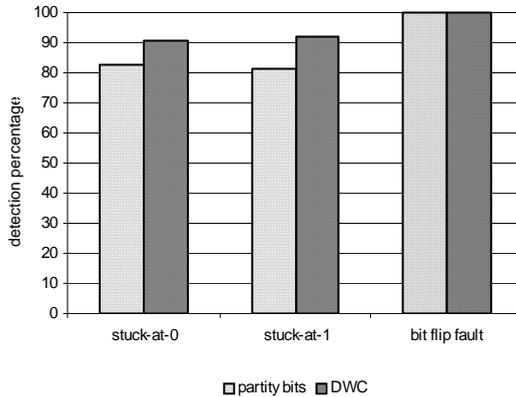


Fig 6. Probability of permanent error detection using parity bits and DWC

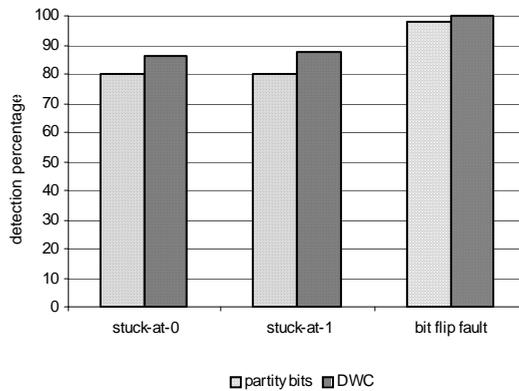


Fig 7. Probability of transient error detection using parity bits and DWC

7. CONCLUDING REMARKS

Fault attacks are becoming a serious threat to hardware implementations of cryptographic systems. Proper countermeasures must be adopted to foil them. In this work, hardware redundancy CED method for detection of faults injected in S-boxes of hash function HaF is presented. It can detect both transient and permanent faults that affect

the output of the S-box. It can protect also against single and multiple errors. This method can provide high coverage for multiple-bit errors, which are the most common fault attacks. The coverage depends heavily also on the fault model. Our simulation experiments conducted on a large number of test cases show that proposed parity bit solution has 100% fault coverage in the case of bit flip errors. The fault coverage drops below 100% only for single transient errors. The coverage equals 94.5% in this case. This solution can be useful for concurrent checking cryptographic chips, especially designed for platforms with very limited resources.

ACKNOWLEDGMENTS

This research was supported by the Polish Ministry of Education and Science as a 2010-2013 research project.

REFERENCES

- [1] Bilski T., Bucholc K., Grochowska-Czuryło A., Stokłosa J., *Hash Function Concept*, Report No, 596, Poznań University of Technology, Poznań 2011.
- [2] Boneh D., DeMillo R., Lipton R.: *On the Importance of Eliminating Errors in Cryptographic Computations*, Journal of Cryptology, vol. 14, 101–119, 2001.
- [3] Boneh D., DeMillo R., Lipton R.: *On the importance of checking cryptographic protocols for faults*, Proceedings of Eurocrypt, Springer-Verlag LNCS 1233, 37–51, 1997.
- [4] Idzikowska E., Bucholc K.: *Concurrent Error Detection in S-boxes*, International Journal of Computer Science & Applications, Vol. 4, No. 1, 2007, 27–32.
- [5] Idzikowska E., Bucholc K.: *Error detection schemes for CED in block ciphers*. Proceedings of the 5th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing EUC 2008, Shanghai 2008, 22–27.
- [6] Idzikowska E., *CED for S-boxes of symmetric block ciphers*, Measurement, Automation and Monitoring, vol.56, nr 10, 2010, 1179–1183.
- [7] Stokłosa J., Bilski T., Pankowski T., *Bezpieczeństwo danych w systemach informatycznych*, PWN, Warszawa-Poznań 2001.

PART V
NETWORKS

*IP networks,
IEEE 802.11 based WLANs,
handover, mobility protocols,
performance analysis, comparison*

Józef WOŹNIAK*, Przemysław MACHAŃ*, Krzysztof GIERŁOWSKI*,
Tomasz GIERSZEWSKI*, Michał HOEFT*, Michał LEWCZUK**

PERFORMANCE ANALYSIS OF MOBILITY PROTOCOLS AND HANDOVER ALGORITHMS FOR IP-BASED NETWORKS

A rapid growth of IP-based networks and services has created the vast collection of resources and functionality available to users by means of a universal method of access – an IP protocol. At the same time, advances in design of mobile electronic devices have allowed them to reach utility level comparable to stationary, desktop computers, while still retaining their mobility advantage. Following this trend multiple extensions to the base IP protocol stack have been developed, devoted to user mobility support. In this work we present a short overview of the most popular methods of handling mobility in IPv4 and IPv6 networks, along with their overall performance analysis and comparison. Critical performance issues of IP mobility mechanisms are presented, as well as optimizations proposed in standardized solutions.

1. INTRODUCTION

During the last decade, two dominant technology trends can be observed. The first one is the Internet – an easily accessible internetwork offering numerous services based on a single network protocol – the Internet Protocol (IP). This trend, commonly called “All-IP” approach, results in a universal way in which services are provided to remote users.

* Gdańsk University of Technology, Faculty of Electronics, Telecommunication and Informatics, Gdańsk, Poland.

** Mega S.A., ul. Hutnicza 42, 81-038 Gdynia, Poland.

The second one is mobility and is mostly stimulated by advances in design and development of portable electronic devices allowing them to reach utility level comparable to stationary, desktop computers, while retaining their mobility advantage. Unfortunately, mobility in IP networks brings a number of problems which need to be solved if IP-based services are to be offered to end-users with satisfactory quality. Mobile users need to communicate without interruption while moving across different access networks, which results not only in the necessity to change points of physical network access (handover) but also in probable changes of users IP addresses.

According to the OSI model, the mobility can be addressed by two classes of solutions: mobility assisted by low layer mechanisms (typically ISO-OSI Layer 2 solutions) and mobility in the network layer (ISO-OSI Layer 3). Network layer mobility, based on the IP protocol, is more universal than the lower layer solutions because it is not technology-dependent and can be deployed in heterogenic environments. On the other hand, Layer 2 mobility outperforms the IP-based solutions due to the access to much more information useful in mobility support. Both mentioned mobility solutions are independent from each other and may be applied simultaneously. Layer 2 mobility is useful in scenarios covering relatively small geographical area, as it is strictly limited to one network technology and thus may be classified as a micro-mobility. From the networking point of view, this kind of mobility is suitable for one broadcast domain (although is also used in emulated broadcast environments like WiMAX which is connection-oriented). IP-based mobility, on the other hand, can be used to widen the potential area of host mobility and thus can be classified as a macro-mobility solution. Still, the IP-based solutions also address the issues of micro-mobility. Extended discussion on micro- and macro-mobility is presented in Section 4.

This work presents IP-based mobility in IEEE 802.11 environment along with several performance optimization proposals. In Section 2 we provide definitions of basic terms and general handover taxonomy. The following two sections are dedicated to more detailed description of the most popular standardized solutions for Layer 2 (Section 3) and Layer 3 (Section 4) handovers, completed with performance analysis, examination of common problems and most prominent optimization proposals. Section 5 describes cross-layer approach to handover, integrating Layer 2 and Layer 3 mobility mechanisms. The work ends with Section 6 containing final conclusions.

2. HANDOVER

As a mobile user moves, it becomes necessary for him to change his physical point of network access. Such change can result in variety of additional requirements for the mobile node to fulfill, starting from a simple change of network identification information in transmitted data frames. This can be done through performing a complete or

partial association procedure with a new access point, to a complete access technology change and/or change of network level addressing information (also resulting in different routing paths). The complexity of necessary procedures can differ very significantly depending on particular scenario.

In the context of network mobility terms roaming, handover and handoff often appear, as they refer to aspects of the mobility support [5]. Roaming is the network operator-based term and a set of procedures, involving formal agreements between operators, which allow a mobile (terminal) to obtain an access to the network using an foreign infrastructure. The term handover (or handoff) refers, in turn, to the process of mobile node moving from one point of attachment to the Internet to another one. There are several types of handover defined depending on which layers of communication stack are affected [6], [24], [32].

New technologies and applications have a strong impact on the handover requirements. The first point is the need for enhanced address concept. In both IPv4 and IPv6 technology IP address is used as host identifier and additionally provides information about the location of terminal. Possible solutions are to hide address location or to include user location into addressing concept [6]. The second case is the support for unsymmetrical (upward and downward) vertical handovers in heterogeneous networks. Upward vertical handover is time-critical because duration of small cell layer is time constrained. Heterogeneous network implies that mobile host is able to operate in any technology that is used in the network. Operation means forwarding services, support for Authentication, Authorization, and Accounting (AAA) services and quality of service (QoS) support.

The elementary mechanisms for mobility support tend to be relatively simple in their base architecture. However, the necessity of taking into account the above requirements while still providing adequate performance creates the need for optimizations of these solutions which are sometimes complex.

In the following sections we present the justification for the need of improvements and describe the major mechanisms, optimization methods and procedures supporting users mobility.

3. LAYER 2 HANDOVERS – TRANSITIONS BETWEEN APS IN IEEE 802.11 STANDARD - BASED WLANS

The IEEE 802.11 standard was initially published in 1997 [11] by Institute of Electrical and Electronics Engineers (IEEE). The standard has been prepared as a wireless extension of existing IEEE 802.3/Ethernet standards. Many amendments were merged into the standard since, resulting in its current, vastly extended version, named IEEE

802.11-2007 [12]. The most recent additions seem to focus on providing support for creating complex network systems (both homogenous and heterogeneous). Handover mechanisms, as an important element of such systems' functionality, are also being rapidly developed – Layer 2 handover support procedures were described in IEEE 802.11f amendment [13]. The document introduced a way of exchanging information about Mobile Stations between Access Points. However, the extension did not gather the expected popularity and had been withdrawn.

When the mobile station moves between Infrastructure BSSs it will reassociate with AP in the new BSS, i.e. perform Layer 2 handover. To facilitate seamless handover the neighbor APs are configured to operate on different channels to overlap the coverage. The example handover procedure is presented in Figure 1.

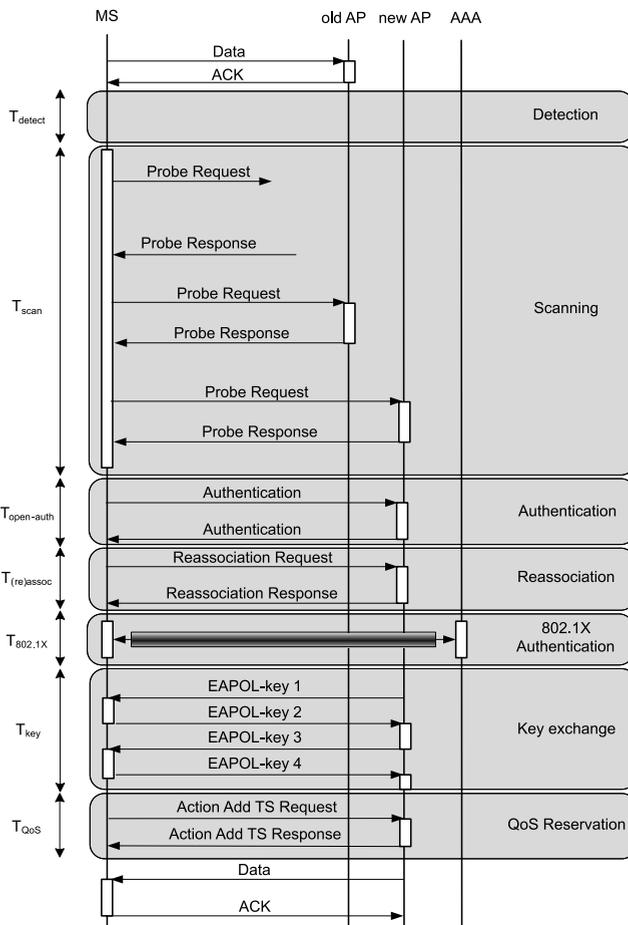


Fig. 1. Timing of IEEE 802.11 handover with open authentication

The detection phase can be defined as the time before the station decides to move to another AP but some operation to determine that decision is already made. For example, if the station decides to reassociate when 3 Beacons from the current AP are lost, then the timeout of 3 Beacons would be the detection time. In practice that phase is hard to specify and measure. During the scanning phase the station checks all physical channels by switching radio frequency for APs in vicinity. In the next steps the station executes 802.11 authentication and association with Access Point. The mobile station may also execute 802.1X authentication on the top of the 802.11 association. In such a case additional 4-way handshake is for key exchange and derivation. The QoS reservation refers to the resources reservation procedures introduced by IEEE 802.11e amendment. Finally, the Layer 2 handover delay ($T_{802.11}$) can be presented with the equation (1).

$$T_{802.11} = T_{\text{detect}} + T_{\text{scan}} + T_{\text{open-auth}} + T_{(\text{re})\text{assoc}} + T_{802.1X} + T_{\text{key}} + T_{\text{QoS}} \quad (1)$$

The empirical results for handover parameters are presented in Table 1.

Table 1. Empirical results for handover procedures, after [36]

Parameter	Value
T_{detect}	0 – 1600 ms
T_{scan}	58 – 400 ms
$T_{\text{open-auth}}$	1 – 10 ms
$T_{(\text{re})\text{assoc}}$	1 – 10 ms
$T_{802.1X}$	200 – 700 ms
T_{key}	5 – 50 ms
T_{QoS}	1 – 10 ms
$T_{802.11}$	267 – 2790 ms

The most time consuming operations are detection, scanning and 802.1X authentication. Each of them has an alternative solution which minimizes the time consumed. The detection time T_{detect} can be reduced by introducing AP-initiated handover. In such a case the AP monitors current connection quality and compares it with the one possible to obtain on other channels. The mobile host is then switched by the AP according to the measured qualities. The scanning time T_{scan} can be reduced by the usage of algorithms proposed in literature, e.g. The 802.1X delay $T_{802.1X}$ is addressed in 802.11r standard amendment. sync scan [30].

In case when the Access Points are under consolidated management there is a number of mechanisms designed to facilitate the handover process and help mobile client to make a seamless transition. Apart from a number of proprietary solutions introduced by hardware developers (most often based on dedicated wireless network controllers, among which Virtual Cell technology is one of the most recognized), there exists a widely known IEEE 802.11r standard amendment [14].

All of the mentioned improvements address the issue of handover time by the incorporation of Layer 2 mechanisms dedicated for the particular technology. In the next section we will describe the major mechanisms, optimization methods and handover procedures.

4. IP MOBILITY PROTOCOLS

Apart from low-layer handover procedures, regarding fast and seamless change of physical point of network attachment, a number of issues concerning IP protocol operation must also be addressed. The exact required mechanisms depend on a specific mobility type [18] – for example terminal mobility is the ability for a user terminal to access the network when the terminal moves. Another type of mobility – user mobility – is the ability for a user to access the network under the same identity when the user changes location and often includes the ability to access the network from the different terminals under the same identity. Service mobility is the ability for a user to access the service regardless of user location. All of these mobility types bring different challenges and require specialized solutions.

Mobile IP has introduced the concept to decouple the host and network identifiers that had been fundamental in traditional IP addressing. With the introduction of two addresses Mobile IP has solved the problem of delivering IP datagrams to the mobile host that moves between networks. The basic concept was later improved in multiple fields like routing optimization, handover efficiency and deployment cost.

Classical MIPv4 solution contains a significant disadvantage in form of triangle routing – the data from Corresponding Node is delivered to Home Agent in MH's home network, which in turn delivers it to MH's current IP address. The Route optimization extension [28] was proposed to overcome that problem. In case of MIPv6, an extension of registration procedures allows MH to inform its corresponding nodes of its current IP address – such knowledge then permits them to transmit data directly to MH's current address, without retransmission by Home Agent.

The handover efficiency oriented algorithms concentrates on optimization of data paths, which reduces transmission latency, packet loss during handover and consumption of network resources. Because of frequently encountered in IP systems division of network into hierarchical domains, the mobility can be divided into two broad types: inter-domain mobility and intra-domain mobility. Such an approach opens the possibility of performance optimizations. A domain is defined as a large wireless network under a single authority.

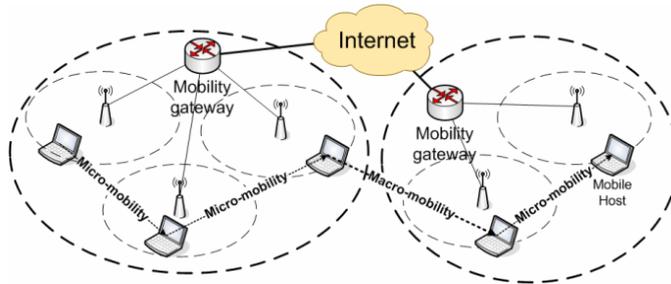


Fig. 2. Micro and macro-mobility

Inter-domain mobility (also called macro-mobility) is related to a movement from one domain to another. Such mobility most often results in complex handover procedures including full low-layer handover, full authentication, new IP address acquisition and verification, mobile node registration and radical data path change.

On the other hand, intra-domain mobility (also called micro-mobility) refers to user's movement within a particular domain. In this case, many of the necessary handover procedures can be simplified – for example: fast re-association in place of full association/authentication procedure and no IP address change.

Mobile IP remains the most popular solution for macro-mobility support. It is not considered as the efficient solutions but provides the required mobility support for the infrequent movements between domains. The number of solutions strives to provide low latency handovers for micro-mobility (e.g. Cellular IP, HAWAII and TIMIP). Similarly Hierarchical Mobile IP [8], [34] introduces a regional mobility agent called Gateway Foreign Agent that facilitate local mobility.

Another approach to decrease the handover delay is usage of link layer triggers. The Mobile IP extensions, named fast handovers [19] or low latency handovers [23], aim at forecasting handover and preparing the transition to the new Access Router (AR) before the connection to the old AR is lost.

The deployment of Mobile IP requires both network architecture changes as well as changes in the mobile host protocol stack. Even though the solution is on the IETF standardization track for a long time, the protocol implementations are still not widely available. For those reasons the alternative solutions are proposed. The first concept assumes that mobility can be handled completely inside the network. Thanks to that no changes are required to the current mobile hosts. Proxy Mobile IPv6 extends these concepts.

The second group of protocols utilizes NAT concept that is already widely used in the IP network. The authors argue that NAT-based solution is easier to deploy with the compromise to the functionality. For example, Reverse Address Translation (RAT) [33] is the macro-mobility approach based on NAT that supports only UDP traffic. On the other hand, Mobile NAT [2] provides both micro- and macro-mobility support and can be deployed as the Mobile IP replacement.

New trend in mobility support utilizes effective cross-layer mechanisms combining low-layer (ISO-OSI layer 2) and network-layer handovers. The example algorithm – Simultaneous Handovers IEEE 802.11r for Mobile IPv6 – is described in details.

4.1. MOBILE IP

The IETF Mobile IP [15], [16] is the oldest and the most widely known approach for mobility support in IP networks. There are two versions of the algorithm, namely for IPv4 and IPv6 protocols. Both solutions are on standard track run by IETF organization. Mobile IP offers mobility support in the network layer and isolates higher layers from mobility. The key idea introduced in Mobile IP is the usage of a couple of addresses to manage user movement. Mobile host owns its own IP address which can be referred as traditional IP address. Mobile IP introduces a term of home address for such an address. Each time the mobile host connects to the network, the temporary IP address for the current network is obtained. The host remains reachable by the way of both home and temporary addresses. For Mobile IP the temporary IP address is named Care-of Address (CoA). A correspondent host (CH) sends datagrams destined to the mobile host using its IP home address and the datagram is tunneled via Mobile IP infrastructure to the mobile node current location.

In the Mobile IPv6 the MH is able to create its own CoA using its link-local address and automatic address configuration (i.e. merge subnet prefix with own hardware address). Mobile IPv4 adds also Foreign Agents (FA) that are located in any network that can be visited by the mobile host and facilitates CoA generation.

Moreover, Mobile IP extends IP infrastructure by the concept of Home Agent (HA). HA is located in the home network, which is defined as the network that mobile IP address belongs to. It must be provided current information about current CoA of MH. It is up to MH to notify appropriate MIP entity of its location. In case of MIPv4 MH informs HA of its current CoA with assistance of Foreign Agent. In MIPv6 the process is simplified, and MH informs HA directly.

To enable MH to detect location changes and addresses of appropriate Mobile IP Agents, they periodically send Agent Advertisement message (in case of MIPv6 an extension of ICMP Router Discovery message [4]) with fields for mobility support. Mobile Host can solicit Agent Advertisement by sending Agent Solicitation message.

The HA intercepts traffic addressed to MH home address and, if MH is not in home network and thus cannot receive such traffic directly, sends the traffic to a network in which the MH is currently located by means of tunneling. In case of MIPv4 the tunnel is created to Foreign Agent in MH's current network, which terminates the tunnel and delivers the traffic to MH. In case of MIPv6 the procedure is simplified, as the tunnel is created directly to MH.

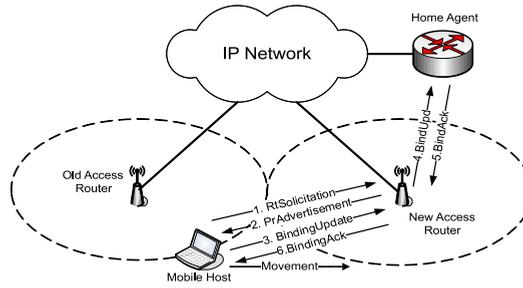


Fig. 3. Mobile IPv6 handover

The MIPv6 handover example is presented in Fig. 3. When a mobile host leaves its home IP network it detects foreign networks based on Router Advertisement messages that can be solicited. To begin data transmission mobile host updates bindings with Home Agent and corresponding hosts if any.

Summing up, from the perspective of the correspondent host, the mobile host is identified by its home address. When the packets are sent to the mobile host by another host (called in this instance a Corresponding Node – CN or equivalently a Corresponding Host – CH), HA (which stores the registration mapping between home address of the host and its current Care-of address) intercept packet based on home address of mobile host. The datagram is then tunneled from HA to a Mobile Host (directly or through FA).

Table 2 presents the relative TCP goodput for the different handover models compared with the “no handoff” scenario. The goodput is here understood as the throughput in the application layer, having considered all the possible lower layers overhead like headers and the possible retransmissions. Two mobility scenarios were taken into

Table 2. Empirical results for Mobile IP family of protocols [10]

Frameworks	Avg. Goodput (Kbytes/second)		Time to transfer 6.5M file (seconds)	
	Linear	Pingpong	Linear	Pingpong
MIPv6	100.847	83.820	66.001	79.408
MIPv6 + Fast'H	101.213	90.240	65.762	73.759
HMIPv6	101.520	91.587	65.563	72.674
HMIPv6 + Fast'H	101.593	93.867	65.516	70.909
Bicast/n-cast	101.580	92.713	65.524	71.051
SMIP (nonop)	102.007	91.113	65.127	68.538
SMIP	103.106	102.120	64.554	65.178
No Handoff	103.293		64.438	

account. In the linear case, the Mobile Host executes a single reconnection to the next router. Pingpong scenario is when the hosts performs multiple handovers between old and new router in a short time. The impact of the Mobile IPv6 handover on the goodput is around 3%. However, handover protocol optimizations can make this impact negligible.

The solution described above, straightforward and robust in its basic approach, contains a number of critical elements, which can negatively impact MH's communication performance.

The first such element is movement detection. Absent some cross layer information from link layer, the MH has no other means of detecting that it has changed location, than by sending Agent Solicitation and analyzing Agent Advertisement messages. Various algorithms can be used to decide, when MH should change its current agent if such change is detected. Some (for example Lazy Cell Switching [27] – LCS) prefer unfrequented change of agent, to minimize number of handoffs which by definition disrupt MH's connectivity. The downside of this solution is somewhat longer connectivity disruption when handoff can be no longer avoided. Others take opposite approach – for example, Eager Cell Switching (ECS) [7] performs handoff as soon as new agent is detected. Such algorithms generally result in shorter handoff times, however they take various assumptions which, when incorrect, result in excessive number of unnecessary handoffs. In case of ECS, it is assumed that MH does not often change its direction of movement. Therefore, if new agent is detected it is highly probable that MH will remain in its domain for a considerable amount of time. If the above assumption is correct, ECS will provide about 3.5 s shorter handoff time than LCS.

The second MIP element affecting performance and MH service continuity is the method of traffic handling. In the base MIP specification, the traffic to MH is first sent to HA which in turn tunnels it to MH current location resulting in triangle routing. Such approach creates a number of possible problems, the most evident one being triangle routing - different paths of MH incoming and outgoing traffic for the same CN. Because the incoming traffic is initially delivered to HA which in turn forwards it to MH current location, the incoming path is significantly longer. Apart from unnecessary consumption of network resources, it has been shown [1], [29] that each additional device can reduce connection throughput by difficult to assess but significant degree. The exact number depends on network configuration and range from 1% per hop to over 45% in adverse network conditions. In addition to throughput degradation, this routing asymmetry results in asymmetrical transmission delay – for each 20% of throughput degradation due to triangle routing there is typically about 50% increase in transmission time [1].

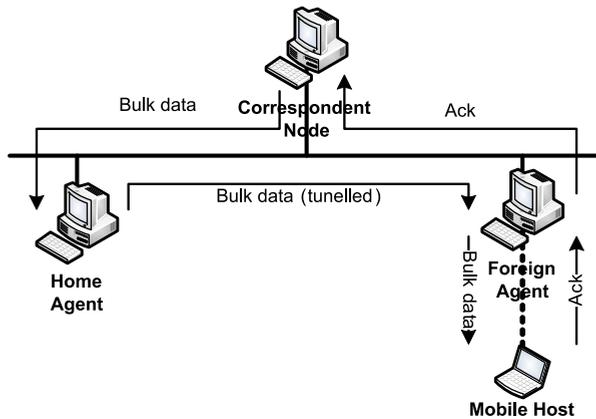


Fig. 4. Triangle routing example

Apart from the obvious unsymmetrical way of traffic delivery, we should remember that the incoming traffic is delivered to MH by means of tunneling. The most popular (due to relatively small overhead) is IP in IP tunneling [26], which requires 20 data octets for tunnel maintenance. While insignificant in case of bulk data transfer, this overhead can reach over 10% of effective throughput in case of small-datagram services, such as Voice over IP.

Moreover, the 20 octet overhead results in change of Maximum Transmission Unit (MTU) size. The results are twofold: simple performance decrease due to fragmentation or significant increase of handoff time when Path MTU Discovery [20] mechanisms are used.

While analyzing MIP impact on network performance, the analysis should not be limited to the network layer. It should also take into account the effects on higher layer protocols. While analysis of numerous application layer protocols is a tremendous task, a simple example of immensely popular, transport layer TCP protocol can serve as a good example of possible impact.

First, it should be noted that, while MH outgoing traffic is delivered directly, many TCP requires acknowledgements of received data and the acknowledgements will be sent through HA. In case of TCP, it can result in about 7% throughput degradation [29]. Many other protocols also have similar requirements concerning two-way communication, so the effect is going to be widespread.

The practical impact of the abovementioned MIP performance problems has been tested in [29]. The results of testbed experiments described there have been presented in Table 3. The experiments have been conducted in a network setup similar to Fig. 4 by performing FTP transmissions between CN and MH. The table includes FTP (TCP) throughput values obtained for 5 different scenarios, designed to illustrate effects of described MIPv4 performance problems.

Table 3. MIPv4 FTP transmission performance results. Source: [29]

Scenario	Tunneling overhead	Fragmentation	Dogleg route (CN-HA-FA-MH)	Mean transfer time [s]	Standard deviation [s]	Throughput [Mbit/s]
To MH in home network	No	No	No	41.21	1.81	5.86
From MH in home network	No	No	No	43.30	1.04	5.58
To MH in foreign network	Yes	Yes	Yes	78.22	1.43	3.09
From MH in foreign network	No	No	Yes (ACK)	46.17	1.14	5.23
From MH in foreign network	Yes	No	Yes	75.62	1.33	3.19

The impact of handoff time on TCP performance is even more evident. It is obvious that during the handoff itself MH should expect a loss of connectivity. However, we should also take into account mechanisms of exponential backoff and slow-start present in TCP.

TCP requires a transmitted datagram to be acknowledged in a specific time interval or the datagram is going to be retransmitted. The interval is exponentially extended with each subsequent retransmission of the same datagram due to lack of acknowledgement. As can be seen in Figure 5, handoff delay combined with this mechanism can

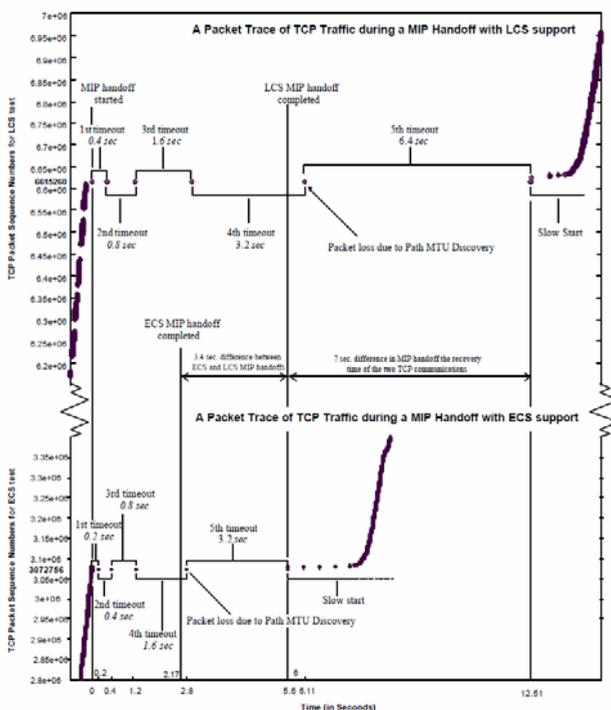


Fig. 5. MIP handoff time impact on TCP performance. Source: [7]

result in much longer connectivity loss than previously anticipated. Moreover, TCP specification requires slow start is any timeouts were encountered during transmission, which is certainly the case here, and which results in further performance degradation. If Path MTU Discovery mechanism is used and MH handoffs from its home network to foreign network, should also expect the loss of the first datagram transmitted after handoff completion, as MTU of the transmission path is lower due to tunneling. It further increased the described exponential backoff effect.

Due to the obvious impact of the described MIP performance problems on quality of end-user experience, many optimizations to the base MIP specification have been proposed. The handoff-related connectivity loss problem have been addressed by solutions aimed to minimize Layer 2 handoff time (such as IEEE 802.11r described in Section 4.2) and a considerable number of optimizations (described in Section 4.4) modifying or extending MIP mechanism itself (ISO-OSI Layer 3). Triangle routing inefficiencies were also considered and some of the proposed solutions are described in Section 4.3.

4.2. FAST BSS TRANSITION (IEEE 802.11R)

IEEE 802.11r-2008 [14] is an amendment to the IEEE 802.11-2007 standard that introduces Fast Basic Service Set Transition. The handover has already been supported under the base 802.11-1999 standard; four messages were required to connect to the new AP in the typical case. However, as the standard is extended, the number of frames went up dramatically. IEEE 802.11r-2008 amendment proposes algorithms to bring the number of frames required for handover down to the level of 802.11-1999. This is expected to be achieved by limiting the number of frames for 802.1X authentication and 802.11e admission control.

IEEE 802.11r-2008 amendment introduces Fast BSS Transitions (FT) which allows Mobile Station to fully authenticate only with the first AP in the FT Domain and use shorter association procedure with the next APs in the network. The amendment defines the FT Domain as the group of APs that support FT Protocol and are connected over a Distribution System (DS). The MS session i.e. security and QoS information is cached in the network. When the station associates with the first AP in the FT Domain it is now pre-authenticated with other APs in the domain.

The first AP the station authenticates to, will cache its Pairwise Master Key (PMK – the starting point of key hierarchy) and use it to derive session keys for other APs. The first AP is named R0 Key Holder (R0KH) as it holds level 0 PMK (PMK-R0). When MS reassociates to the next AP in the domain, R0KH generates PMK-R1 and forwards that to the next AP, which is called R1KH. The R1KH interacts with the R0KH, rather than directly with AAA server. Next, when the MS requests R1KH to prepare reassociation with consecutive AP, R1KH communicates with the R0KH. R0KH generates PMK-R1 and forwards that to the consecutive AP.

The amendment defines two methods of FT: over-the-air and over-the-DS. In the first case MS communicates over a direct 802.11 link to the new AP. In the over-the-DS method the MS communicates with the new AP via the old AP. In the Over-the-air FT protocol the Mobile Station is already associated with the old AP from the domain. At some point MS decides to reassociate with nAP sending 802.11 Authentication frame with Information Elements required by FT Protocol. The new AP responds with 802.11 Authentication frame that contains the same types of Information Elements as request. In the next step MS sends 802.11 Association Request message with FT information elements. Access Point responds with 802.11 Association Response message that also convey FT information elements.

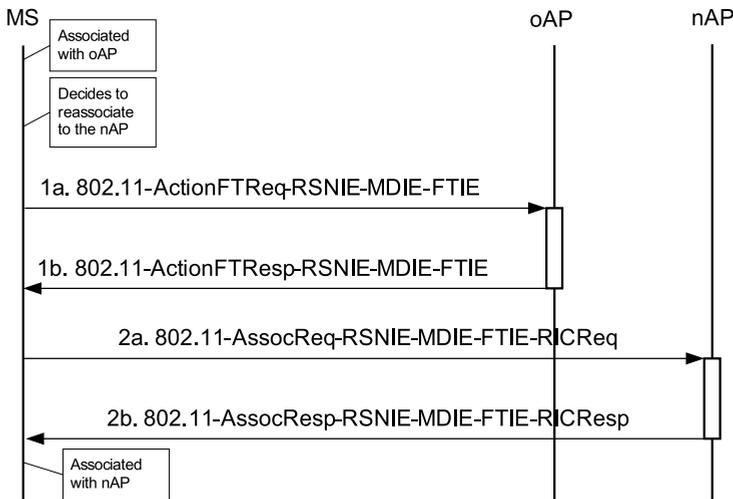


Fig. 6. Over-the-DS FT Protocol

Fig. 6 presents Over-the-DS FT protocol version. The MS uses Action frame to communicate with the current (old) AP, providing the address of the new AP. Old AP communicates over the DS with new AP forwarding STA request. The new AP responds over DS and oAP sends Action FT Response to MS. At this step MS is authenticated with nAP. Then, MS switches the channel and begins association procedure with nAP. The type and content of information elements is the same using both methods: over-the-air and over-the-DS.

The IEEE 802.11r-2008 handover performance is discussed in [3], [22]. The handover with FT algorithm is much faster comparing to the legacy mode, simply because 802.1X Authentication phase can be substantially shortened. The example simulation results are presented in Table 4. These results show the real benefits arising from the use of TF method, in particular with the Over-the-DS FT.

Table 4. Handover delay for FT protocols [22]

Phase / Algorithm	Regular 802.11	Over-the-Air FT	Over-the-DS FT
Scanning	315,14 ± 3,24	314,90 ± 3,62	0
Authentication	0,37 ± 0,14	0,82 ± 0,24	0
Association	0,79 ± 0,28	1,25 ± 0,58	3,30 ± 2,77
1X Authentication	542,19 ± 0,85	0	0
4Way Authentication	0,97 ± 0,47	0	0
QoS	1,23 ± 0,98	0	0
Channel switching	10,00 ± 0,00	10,00 ± 0,00	10,00 ± 0,00
Layer 2 delay (ms)	866,93 ± 5,38	324,65 ± 5,12	13,22 ± 2,87

4.3. ROUTING PATH OPTIMIZATIONS

The triangle routing problem was addressed by route optimization extension [28]. It adds binding update to inform the corresponding host on the current CoA. When the handover occurs the old FA communicates with the HA using binding warning. Thanks to the HA can update binding in the corresponding host which in turn allows CH to send data to MH CoA directly, bypassing inefficient HA/FA tunneling and eliminating triangle routing.

As can be expected, the elimination of triangle routing removes some of the important issues of MIP discussed before – it is possible to avoid both the mandatory transmission path asymmetry and the need to tunnel incoming traffic between HA and MH. Absence of triangle routing results in lack of additional (unilateral) transmission delay, lowers network resource consumption and improves throughput. Absence of tunneling removes its direct overhead and prevents fragmentation-related performance losses.

Handoff time is not significantly affected by these optimizations, unless Path MTU Discovery is used to avoid fragmentation. In such case, the initial handoff directly from home network can be performed with loss of one less datagram which can be important for protocols such as TCP as shown in the earlier example.

The drawbacks of the solution are the new requirements on the CH – ability to encapsulate IP packets and store CoA. Moreover, route optimization increases the signaling overhead. The extension proposed in [20] adds link and signaling cost functions to capture the trade-off between the signaling cost and processing load. The alternative approach presents DHARMA [25] that uses overlay network to select location-optimized mobility agent from the distributed set of home agents to minimize routing overheads.

4.4. HANDOVER PERFORMANCE OPTIMIZATIONS

As can be seen from the provided analysis, the handoff time directly impacts the length of the period during which MH is lacking network connectivity. Moreover, if we take into account higher layer protocols (such as, for example TCP), there is also an indirect, and often difficult to predict, impact of handover time on the length of network connectivity disruption. That being the case, it is evident that reducing handover time is a logical method of improving MIP performance. The most popular approaches involve distinction between micro/macro-mobility and cross layer solutions.

The generic Mobile IP protocol does not differentiate between local and global handovers. In consequence the amount of signaling is large, especially if HA is distant from the mobile host. Moreover, MIP applies the same scheme for horizontal and vertical handovers which makes performance optimizations difficult. For those reasons micro- and macro-mobility concepts were introduced.

In case of micro-mobility, MH moving inside an area called mobility domain does not need to perform all the procedures described for macro-mobility scenario when change of physical network access point becomes necessary. It is sufficient to employ much simpler signaling with local mobility support entities, which is enough to provide correct data routing to MH's public address within a given mobility domain. This approach allows reducing handover time significantly. For example, Cellular IP [35] adds the micro-mobility support to the MIPv4. It also addresses a problem of MIPv4 scalability by adding local caches to support slow moving and sleeping nodes. Handoff Aware Wireless Access Internet Infrastructure (HAWAII) [31] is another micro-mobility support extension to MIPv4 that optimizes both handoff latency and data paths, by including cross-layer triggers and improved IP QoS maintenance.

Another concept is used in fast handovers [19], [23]. Such solutions are typically applied for intra-technology handovers and are tightly coupled with link layer protocols. Those protocols utilize the physical triggers from lower layers, like "Link Going Down" or "Link Down" to speed up the handover. The simultaneous handover discussed later is the next step to improve the handover performance. Based on the fast handovers concept, the link layer and IP layers handovers are executed in parallel.

4.5. DEPLOYMENT COST OPTIMIZATIONS

Due to relatively high complexity of mobility protocols, the deployment process may also be optimized. The deployment is here understood as all the changes in protocol stacks which have to be done in several locations in order to provide the required functionality to mobile host. The changes may be presented here in the form of a list:

- home agent functionality deployment,
- foreign agent functionality deployment,
- mobile host functionality deployment.

The home agent and foreign agent functionalities are provided usually as joint modified network stack, but are discriminated here to emphasize the need for all the potential Mobile Host locations to be addressed. The changes are required for both the macro- and micro-mobility and if the optimizations are present, like Gateway Foreign Agent for example, the according deployment forms another step in the list above.

In many popular Mobile IP solutions, the end-user requires a lots of modifications by himself. The modifications involve mainly the signaling of handover procedure handling. The various mobility protocols implementations for Mobile Host are prepared for either user space or kernel space. Kernel space alteration is required in both cases though, as the ARP (for IPv4) or ND (for IPv6) behavior modification is required.

In order to simplify the deployment procedure, an interesting variant of mobility protocol called Proxy Mobile IP has been proposed in [16]. Its most attractive feature is the lack of Mobile Host stack modification requirement. The protocol is presented in detail in Section 4.6.

4.6. PROXY MOBILE IP

As opposed to Mobile IPv4 and Mobile IPv6 which are host-based mobility standards, Proxy Mobile IPv6 (PMIPv6) [16] presents a network-based approach, which does not require any kind of client-side mobility agent. Hence, it stands in an opposition to all other MIP protocols and thus Proxy Mobile IP is also called a technology which has shifted the mobility paradigm – from CMIP (Client Mobile IP) to network-based solutions. Such a paradigm shift brings numerous advantages, such as simplified management, the ability to support legacy clients and better efficiency of radio-link utilization.

PMIPv6 extends and reuses a proven MIPv6 idea, however it does not require any modification of a standard mobile node's IPv6 stack. A network-side proxy mobility agent used in place of MIP client-side agent, and performs signaling and management on behalf of the mobile host. As a result, PMIPv6 provides efficient solution without tunneling and signaling overhead on radio access link. Nevertheless, Proxy Mobile IPv6 cannot be deployed as a standalone global mobility system, due to the lack of standardized macro-mobility procedures and mechanisms.

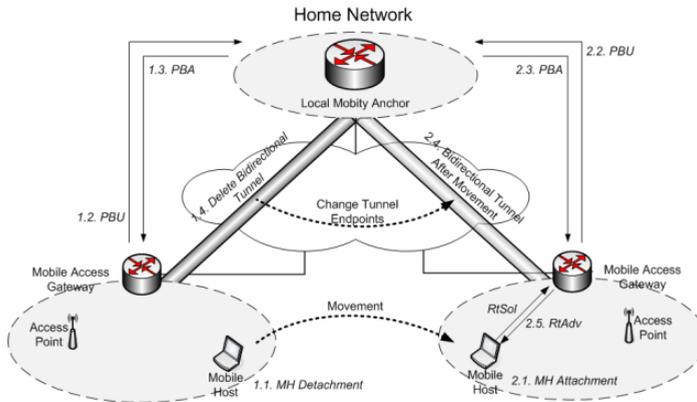


Fig. 7. Proxy Mobile IPv6 Domain

Proxy Mobile IPv6 (as defined in [16]) uses two specialized network elements: Media Access Gateways and Local Mobility Anchors (Fig. 7). Media Access Gateway (MAG) is responsible for several tasks. Firstly, it tracks the MH's movement between the old and new AP. MAG. Secondly, it is responsible for creating a bidirectional tunnel to Local Mobility Anchor and thus managing the connectivity between the MH and the LMA. Local Mobility Anchor (LMA) plays similar role to Home Agent from the typical Mobile IPv6. It is responsible for maintaining routes to all Mobile Hosts in the domain and forwarding traffic to and from them.

The PMIP protocol is a relatively new concept, but an optimization proposals have been already published in [17]. It is called Fast Proxy Mobile IPv6 and addresses the issue of handover performance. It proposes mechanisms reducing packet loss and handover latency. Two operation modes are proposed in this standard – a predictive and a reactive one. Relying on reports about a forthcoming handover received from the MH, in the predictive mode PMAG (Previous MAG) sets up a tunnel with NMAG (Next MAG). The tunnel is used for traffic forwarding during the handover. In this case, downlink data (the one addressed to the MH) may be buffered in the NMAG.

The reactive mode is similar and may be used if the MH does not send information about a forthcoming handover. In this approach the tunnel is set up by the NMAG just after Layer 2 handover.

An example PMIPv6 testbed has been prepared in Gdańsk University of Technology. It consists of IEEE 802.11a/g access points and a set of virtual machines under the control of either Hyper-V or XEN hypervisors. As the publicly available PMIP implementations were only partially functional [9], a base PMIPv6 functionality was developed from scratch. In such environment handover performance in a typical simple mobility configuration was evaluated. The tests involved two APs, one MH, one CN and a set of two MAGs and one LMA for the PMIPv6 functionality. The sample results of PMIPv6 performance evaluation are presented in Figure 8.

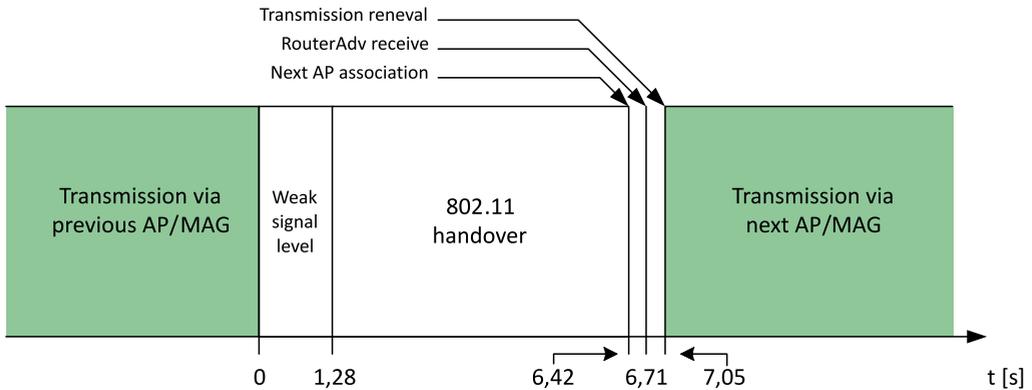


Fig. 8. Simultaneous handover IEEE 802.11r for Mobile IPv6

As can be seen, the handover in the described environment takes relatively long time, approximately 7 seconds. Several reasons contribute to this state. First, the time is measured in the network layer, not in the data link layer. This is driven by the nature of the developed PMIPv6 implementation – for now, it would be hard to measure the handover time in OSI Layer 2. Secondly, due to the fact of other network stack behavior assessment, a TCP traffic has been chosen for tests. As the TCP retransmission mechanism uses geometric delay increase, the time of handover is limited to nearly discrete values. The third reason comes from the IEEE 802.11 standard behavior in the presence of weak signal – the mobile host begins scanning phase (see Section 3) after the drop of three consecutive Beacon frames from Previous AP. The MH wireless card does not offer the same signal level as the AP. Thus, during the handover the beacons are still hearable by the MH, while the traffic frames are yet not received by the AP. This causes dissimilarity in the radio channels and the MH postpones the handover procedures, as it still hears Beacons.

Several alternative IP mobility protocols were introduced that address deployment issues related to the Mobile IP. Reverse Address Translation (RAT) [33] is the macro-mobility approach competitive to the Mobile IP, based on NAT procedure. It advertises easier deployment over MIP by the cost of limited functionality (e.g. no TCP session support). Mobile NAT [2] provides both micro- and macro-mobility support and can be deployed as the Mobile IP replacement. Contrary to the Mobile IP the solution is based on NAT instead of tunneling. Proxy Mobile IP also falls into that category, as it addresses the problem of MIP implementations availability for the mobile hosts. Basic characteristics of described protocols are described in the Table 5.

Table 5. Comparison of IP mobility protocols

Protocol	Mobility Type	Handover Type	Link Detection	Registration	Address Translation
Mobile IPv6 (basic)	Macro	Hard	Router advertisement	At Home Agent	Encapsulation
Hierarchical Mobile IPv6	Universal	Hard	Router advertisement	At Mobility Anchor Point	Encapsulation
Proxy Mobile IPv6	Universal (network-based)	Hard	Events or DNAME	At Mobile Access Gateway	Encapsulation
Fast Handovers	Universal	Hard	Proxy Router Adv.	At Home Agent	Encapsulation
Cellular IP	Micro	Semi-soft or hard	Network specific	Route Updates	No
HAWAII	Micro	Forwarding and non-forw. schemes	Network specific	Path Updates	No
RAT	Macro	Hard (no TCP sup.)	Network specific	At registration server	NAT
MobileNAT	Universal	Hard	Using DHCP ext.	At Home NAT	NA(P)T
Extended SIP	Macro	Hard (no TCP sup)	Network specific	At SIP router	via SIP server

Summing up the above considerations, we can conclude that in spite of wide research in the field of mobility protocols that we now observe in far more to promote universal, reliable algorithm that would meet the demands of users and applications [37]. In light of the results of research on pattern Proxy Mobile IPv6 environment, conducted in our laboratory [9] can be stated that after a significant shortening switching time can become a leading standard among other solutions which offer less flexibility. A major advantage of this solution is the ability to benefit from the mobility of millions of previously manufactured terminals. Bigger effort in this case lying on the carriers, does not appear to be an obstacle, because they are usually obliged to manage the traffic leaving the end-users unaware of controlling the dataflow. In the longer term efforts to optimize the handover latency will help to increase the importance of simultaneous handover in cooperating layers, namely the data link and IP, for example as a proposed standard 802.11r.

5. SIMULTANEOUS HANDOVER IEEE 802.11R FOR MOBILE IPV6

As the need to perform both low-layer IEEE 802.11 and IP-layer handover occurs frequently, an algorithm has been proposed which allows Mobile IPv6 procedures to be executed in parallel to IEEE 802.11 procedures. The algorithm allows faster handovers as MIPv6 detection phase is minimized. MIPv6 handover starts as soon as 802.11 layer detects the new link. Moreover, some 802.11 and MIPv6 procedures are executed simultaneously. The simultaneous handover can decrease the handover delay up to 38% [21]. To achieve that selected Mobile IPv6 frame formats are conveyed as IEEE 802.11 information elements. 802.11 Access Point and MIPv6 Access Router are coupled in a single device.

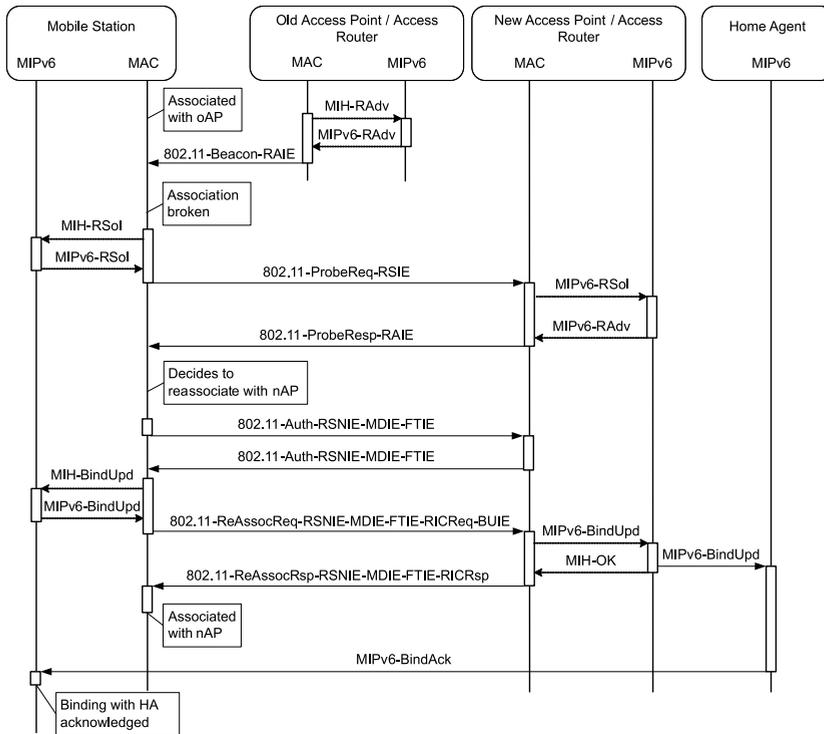


Fig. 9. Simultaneous handover IEEE 802.11r for Mobile IPv6

MIPv6 Router Advertisement unsolicited messages are distributed within RAIE together with 802.11 Beacon messages. While executing active scanning MS can request MIPv6 RA by sending MIPv6 Router Solicitation together with 802.11 Probe Request. The Access Point / Access Router that supports simultaneous handover has to respond with 802.11 Probe Response that includes MIPv6 Router Advertisement. The RA is always known to the Mobile Station when either active or passive scanning method is used. When building Reassociation Request message the Mobile Station will send MIH Binding Update event that is routed to the MIPv6 protocol instance. In response, MIPv6 will send MIPv6 Binding Update message directed to the Home Agent. The MIPv6 Binding Update message is intercepted by 802.11 protocol layer and compressed into BUIE. The BUIE is sent together with Reassociation Request message. When received at the New Access Point BUIE is extracted to MIPv6 Binding Update and passed to the collocated Access Router. AR confirms message reception with MIH OK event. When event is received at MAC layer, it responds with Reassociation Response message. New Access Router sends Binding Update to the Home Agent. The Home Agent responds with MIPv6 Binding Acknowledge. This concludes the handover process.

Table 6. Handover delay for FT protocols [22]

Phase / Algorithm	Over-the-Air FT / MIPv6	Over-the-DS FT / Simultaneous MIPv6
Scanning	314,90 ± 3,62	315,41 ± 2,70
Authentication	0,82 ± 0,24	0,84 ± 0,21
Association	1,25 ± 0,58	1,47 ± 0,62
Channel switching	10,00 ± 0,00	10,00 ± 0,00
Layer 2 delay	324,65 ± 5,12	324,43 ± 5,66
Layer 3 overhead	500,40 ± 56,25	0,84 ± 1,02
Total delay (ms)	825,05 ± 87,99	325,27 ± 6,80

Table 6 presents example simulation results in the scenario of regular Mobile IPv6 and Simultaneous MIPv6 [21]. The Layer 2 handover is executed using 802.11 FT protocol, so the delay is similar. The total handover time is greatly reduced because the Layer 3 operations are executed concurrently with 802.11 association.

6. CONCLUSIONS

The work introduces the handover taxonomy and describes example implementations of the handover procedures, performed in Layers 2 and 3 of the ISO-OSI stack. The authors describe mechanisms and present performance analysis of the mobility support protocols. In particular fast transition solutions for 802.11 WLAN networking environments, standardized by the “r” Working Group, are explained in details. Section 4 also gives an overview of the IP-based Mobility Protocols and their improvements that can provide better handover efficiency (in terms of both handoff latency and data path optimization) over the Mobile IP. Finally, the simultaneous handover methods, performed concurrently in Layers 2 and 3, are also presented, as examples of cross-layer approach to the optimization of the handover latency.

REFERENCES

- [1] BHARGAVA P., *Mobile IP*, CS 599: Wireless Communications and Mobile Computing, 1999.
- [2] BUDDHIKOT M. M., HARI A., SINGH K., MILLER S., *Mobilenat: A new technique for mobility across heterogeneous address spaces*, MONET, vol. 10, no. 3, 2005, 289–302.
- [3] CHUNG-MING H., JIAN-WEI L., *A Context Transfer Mechanism for IEEE 802.11r in the Centralized Wireless LAN Architecture*, IEEE 22nd International Conference on Advanced Information Networking and Applications, 2008.
- [4] DEERING S., *ICMP Router Discovery Messages*, RFC 1256, 1991.
- [5] DUNMORE M., PAGTZIS T., *Mobile IPv6 Handovers: Performance Analysis and Evaluation*, 6NET Project, IST-2001-32603, May 2004.

- [6] FESTAG A., KARL H., SCHAFER G., *Current developments and trends in handover design for ALL-IP wireless networks*, Technical University Berlin, TKN Technical Report TKN-00-007, Version 1.3, 2000.
- [7] FIKOURAS N.A. et al., *Performance Evaluation of TCP over Mobile IP*, 2001.
- [8] FOGELSTROEM E., JONSSON A., PERKINS C., *Mobile IPv4 Regional Registration*, Internet Engineering Task Force, RFC 4857, 2007.
- [9] HOEFT M., GIERSZEWSKI T., GIERŁOWSKI K., WOŹNIAK J., CHRABAŚCZ R., PACYNA P., *Obsługa połączeń terminali ruchomych niewspierających mobilności*, Telecommunications Review, 2011 (in Polish, paper accepted)
- [10] HSIEH R., SENEVIRATNE A., *A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP*, The Ninth Annual International Conference on Mobile Computing and Networking (MobiCom), 2003.
- [11] IEEE Std, *IEEE 802.11-1997: Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1997.
- [12] IEEE Std, *IEEE 802.11-2007: Part11: Wireless LAN Medium Access Control and Physical Layer Specifications*, 2007.
- [13] IEEE Std, *IEEE 802.11f: IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation*, ANSI/IEEE Std 802.11f-2003, 2003.
- [14] IEEE Std, *IEEE 802.11r: Amendment 2: Fast Basic Service Set (BSS) Transition*, 2008.
- [15] IETF, RFC4831, *Goals for Network-Based Localized Mobility Management (NETLMM)*, <http://tools.ietf.org/html/rfc4831>, 2007.
- [16] IETF, RFC5213, *Proxy Mobile IPv6*, <http://tools.ietf.org/html/rfc5213>, 2008.
- [17] IETF, RFC5949, *Fast Handovers for Proxy Mobile IPv6*, <http://tools.ietf.org/html/rfc5949>
- [18] JYH-CHENG C., *IP-Based Next-Generation Wireless Networks : Systems, Architectures, and Protocols*, John Wiley & Sons, 2004.
- [19] KOODLI R., *Mobile IPv6 Fast Handovers*, Internet Engineering Task Force, RFC 5268, 2008.
- [20] LEE Y. J., AKYILDIZ I. F., *A New Scheme for Reducing Link and Signaling Costs in Mobile IP*, IEEE Trans. Computers, vol. 52, no. 6, 2003, 706-711.
- [21] MACHAŃ P., WOŹNIAK J., *Simultaneous handover scheme for IEEE 802.11 WLANs with IEEE 802.21 triggers*, Telecommunication Systems, Volume 43, Numbers 1-2, 2009, 83–93.
- [22] MACHAŃ, P., WOŹNIAK, J., *Performance evaluation of IEEE 802.11 fast BSS transition algorithms*, Third Joint IFIP Wireless and Mobile Networking Conference (WMNC), 2010.
- [23] MALKI EL K. (ED.), *Low Latency Handoffs in Mobile IPv4*, RFC 4881, 2007.
- [24] MANNER J. (ED.), KOJO M. (ED.), *Mobility Related Terminology*, Internet Draft, 2003.
- [25] MAO Y., KNUTSSON B., LU H. H., SMITH J. M., *DHARMA: Distributed Home Agent for Robust Mobile Access*, Proc. IEEE INFOCOM, 2005, 1196–1206.
- [26] MOGUL J., DEERING E., *Path MTU Discovery*, RFC1191, 1990.
- [27] PERKINS C. E., *Mobile IP – Design Principles and Practices*, Addison-Wesley, 1998.
- [28] PERKINS C. E. JOHNSON D. B., *Route Optimization in Mobile IP*, Internet Draft (Work in Progress), Internet Engineering Task Force, draft-ietf-mobileipoptim-11.txt, 2001.
- [29] PERKINS C., *IP Encapsulation within IP*, RFC 2003, 1996.
- [30] RAMANI I., SAVAGE S., *SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks*, Proceedings of IEEE Infocom, 2005.
- [31] RAMJEE R., VARADHAN K., SALGARELLI L. THUEL S. R., WANG S.Y., LA PORTA T., *HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks*, IEEE/ACM Transactions on Networking, vol. 10, no. 3, 2002.
- [32] REINBOLD P., BONAVENTURE O., *A comparison of IP mobility protocols*, University of Namur,

Technical Report Infonet-2001-07, Version 1, 2001.

- [33] SINGH R., TAY Y. C., TEO W.T., YEOW S. W., *RAT: A Quick (And Dirty?) Push for Mobility Support*, Second IEEE Workshop on Mobile Computer Systems and Applications, 1999.
- [34] SOLIMAN H., CASTELLUCCIA C., ELMALKI K., BELLIER L., *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*, Internet Engineering Task Force, RFC 5380, 2008.
- [35] VALKO A., *Cellular IP - A New Approach to Internet Host Mobility*, ACM Computer Communication Review, 1999.
- [36] VATN J.O., *An experimental study of IEEE 802.11b handover performance and its effect on voice traffic*, Royal Institute of Technology, Stockholm, Sweden, 2003.
- [37] WOZNIAK J., MACHAŃ P., GIERLOWSKI K., HOEFT M., LEWCZUK M., *“Comparative analysis of IP-based mobility protocols and fast handover algorithms in IEEE 802.11 based WLANs”*, Communications in Computer and Information Science, Vol. 160, Springer Verlag, 2011, ISBN 978-3-642-21770-8

Sylwester KACZMAREK*, Magdalena MŁYNARCZUK*,
Marcin NARLOCH*, Maciej SAC*

EVALUATION OF ASON/GMPLS CONNECTION CONTROL SERVERS PERFORMANCE

Performance aspects have always been very important in telecommunications, especially in control elements. That problem also regards ASON/GMPLS network, which is one of the propositions for Next Generation Network (NGN) transport stratum. In this work we investigate performance of ASON/GMPLS Connection Control Servers (CCSs) in laboratory testbed, which architecture and operation are also described. In order to evaluate the performance of the ASON/GMPLS connection control elements, a set of scenarios including setting-up and releasing connections were executed in different variants of testbed architecture. During the tests execution call setup and termination operations durations were measured. Test results in the three-layer testbed structure confirmed that connection control layer performance has the main impact on service request processing duration and the influence of the other testbed elements operation is negligible.

1. INTRODUCTION

Latest trends in the modern telecommunications characterized by the growing IP multimedia traffic significance forced the necessity of introducing new network architecture. For this reason ITU-T proposed Next Generation Network (NGN) concept [1]. One of the technologies which are considered for NGN transport stratum realization is Automatic Switched Optical Network (ASON) architecture standardized in ITU Recommendation G.8080/Y.1304 [2]. Control plane of this network is mostly based on GMPLS (Generalized Multi-Protocol Label Switching) [3] protocols like RSVP-TE [4,5].

* Department of Teleinformation Networks, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gabriela Narutowicza 11/12 Street, 80-233 Gdansk, Poland.

Connection control mechanisms in ASON/GMPLS are provided by dedicated Connection Control Servers (CCS). Due to the fact that performance of control elements is crucial in efficiency of every telecommunication network, performance of CCS servers is significant for application of ASON/GMPLS technology in the NGN transport stratum. In the work we present results of performance tests of ASON/GMPLS Connection Control Servers (CCS) carried out in laboratory testbed which was developed and implemented in our department.

The work is structured as follows. The realization of ASON/GMPLS testbed is described in section 2. Section 3 is devoted to performance tests of connection control layer regarding different variants of realization. Carried work is summarized in section 4, in which conclusions and outlook to future are presented.

2. ASON/GMPLS TESTBED

The concept of ASON/GMPLS architecture testbed is presented in Fig. 1. The implementation includes functionality of service control layer, connection control layer as well as optical resource layer represented by Resource Terminals (RTs) emulating Optical Cross-Connects (OXC)s operation. Corresponding layers communicate over dedicated interfaces using communication protocols. Presented architecture fulfills ITU-T recommendations and communication protocols conform to GMPLS standardization.

Service Control Server (SCS) performs the call control functions. It is responsible for handling of user call and call termination request transformed into connection requests in the connection control layer. Additionally SCS provides WWW interface and stores all necessary information in local Oracle 10g XE database [6]. The database in SCS include, among others, CALL_STATE and CALL_STAT tables. The CALL_STATE table stores the state of processed requests and the time of sending and receiving response from the lower layer. The CALL_STAT table (Fig. 2) gathers statistical data regarding performance of handling requests in the system, which is computed based on the content of the CALL_STATE table. Among values stored in the CALL_STAT table are time values describing the interval for which statistic are calculated (T_START and T_STOP), total number of requests (NUM_REQ) and number of lost request (NUM_REQ_LOST), calculated request loss probability (LP), minimal, maximal, average and standard deviation of call set-up time (respectively MIN_T23_1, MAX_T23_1, AVG_T23_1, STD_DEV_T23_1) as well as similar statistical parameters for call release time (respectively MIN_T56_4, MAX_T56_4, AVG_T56_4, STD_DEV_T56_4). Detailed information about operation codes regarding particular call states as well as description of the remaining tables in the SCS database can be found in [12].

Connection Control Server (CCS) performs connection control functions. Each CCS server is responsible for dynamic management of optical resources in transport layer by processing requests for establishing (setting) and releasing (deleting) paths. The data necessary for proper CCS operation are stored in a local database. During the development of the Connection Control Server software different database solutions were implemented and tested. Due to the fact that this aspect is very important in terms of CCS performance, it is described in details in section 3.

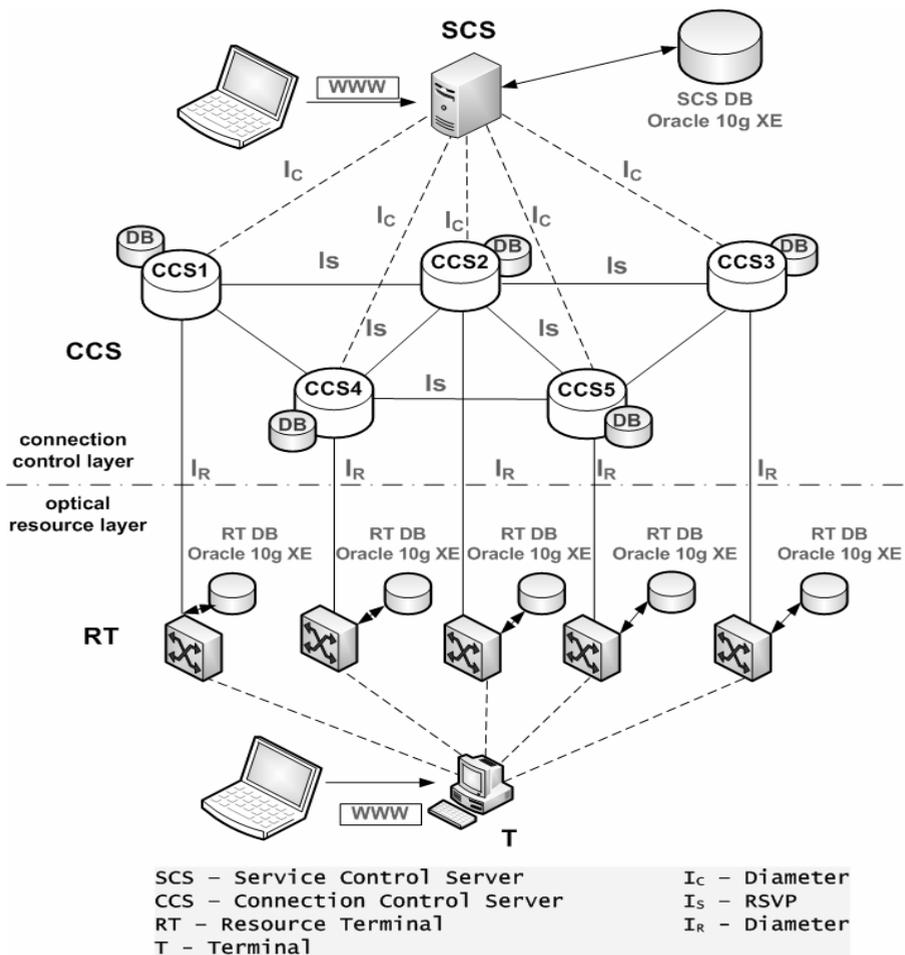


Fig. 1. The concept of ASON/GMPLS architecture testbed

Connection Control Servers utilize RSVP [7] protocol extended to transport objects regarding resource reservation in optical layer [8,9]. The design of Connection Control Server functionality was based on the following assumptions:

- mapping of elements from control layer to transport layer is one-to-one,

- single reservation session results in reservation of one or more transport unit, depending on the bandwidth demand request,
- identifiers of resource layer are transported using LMP procedures [10],
- Fixed Filter (FF) reservation style is applied [7].

Resource Terminals (RTs) emulate optical resources. For this reason they store information regarding state of the emulated device in a local Oracle 10g XE database. Terminal T is used for configuration and verification of reservation state in emulated OXCs. In order to perform these operations dedicated WWW interface which presents database content of Resource Terminals is provided.

ID	T_START	T_STOP	NUM_REQ	NUM_REQ_LOST	LP	MIN_T23_1	MAX_T23_1	AVG_T23_1	STD_DEV_T23_1	MIN_T56_4	MAX_T56_4	AVG_T56_4	STD_DEV_T56_4
90	2011-01-05 18:45:24.33	2011-01-05 18:50:00.00	70	0	0	20,56	64,97	30,32	10,43	12,61	53,53	21,95	10,76

Fig. 2. Fragment of the CALL_STAT table content with exemplary statistics (all values from MIN_T23_1 to STD_DEV_T56_4 are given in milliseconds)

Communication between cooperating layers is performed over well defined interfaces. Service control layer and connection control layer communicate over I_C interface. Information between Connection Control Server and Resource Terminal is exchanged over I_R interface. In both cases Diameter protocol [11] with dedicated messages and appropriate AVP (Attribute Value Pairs) elements is used [8]. Due to space limitation, comprehensive description of the control software in the implemented testbed is beyond the scope of this work and can be found in [12,13,14].

3. CONNECTION CONTROL PERFORMANCE

3.1. TEST ENVIRONMENT

The developed and implemented software for the ASON/GMPLS network elements has been installed and validated in laboratory testbed, which was created to study not only the basic functionality, but also to investigate communication of connection control layer with the remaining layers of the architecture and carry out per-

formance tests. The structure and configuration of the testbed is described in the next part of the section. The results of performed functional tests are presented in [12,13]. Executed performance tests are thoroughly described in section 3.2.

For the purpose of ASON/GMPLS control software performance testing network architecture from Fig. 1 has been implemented. Due to the fact that system platform hardware parameters have strong influence on the performance of the implemented ASON/GMPLS architecture we present brief description of the utilized equipment. Connection Control Server software has been installed on NTT TYTAN computers with the following hardware parameters:

- Supermicro X8DTL-3F motherboard,
- Intel XEON E5506 (2,13 GHz) quad core processor,
- 4GB DDR3 ECC R RAM memory,
- 2x500GB SATA HDD.

Resource Terminal software has been installed on NTT computers with the following hardware parameters:

- Gigabyte GA G31M-ES2L motherboard,
- Celeron E3300 (2,5GHz) dual core processor,
- 2GB DDR2 DIMM memory,
- 250GB SATA HDD.

Network configuration of the implemented testbed (along with IP addresses of all equipments) is described in Fig. 3. Execution of performance tests required proper configuration of Debian Linux operating system and implemented software.

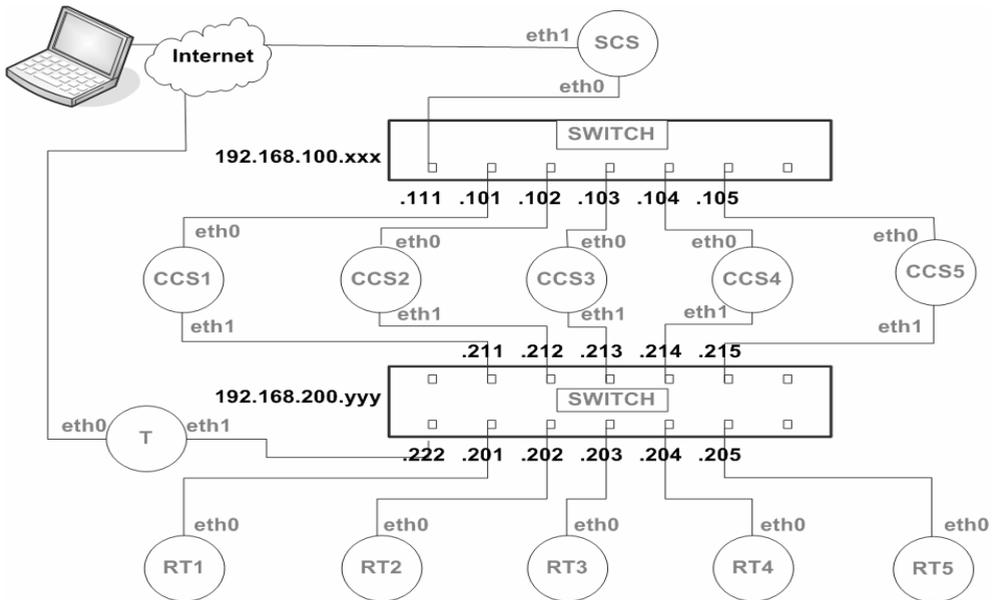


Fig. 3. Architecture and configuration of the implemented ASON/GMPLS testbed [12,13]

Due to physical network interfaces limitations in the hardware, implementation of the connection control layer from Fig. 1 involved configuration of Virtual LAN interfaces VLANs. The structure of the used virtual networks and the sets of optical resources identifiers allocated to particular interfaces are presented in Fig. 4.

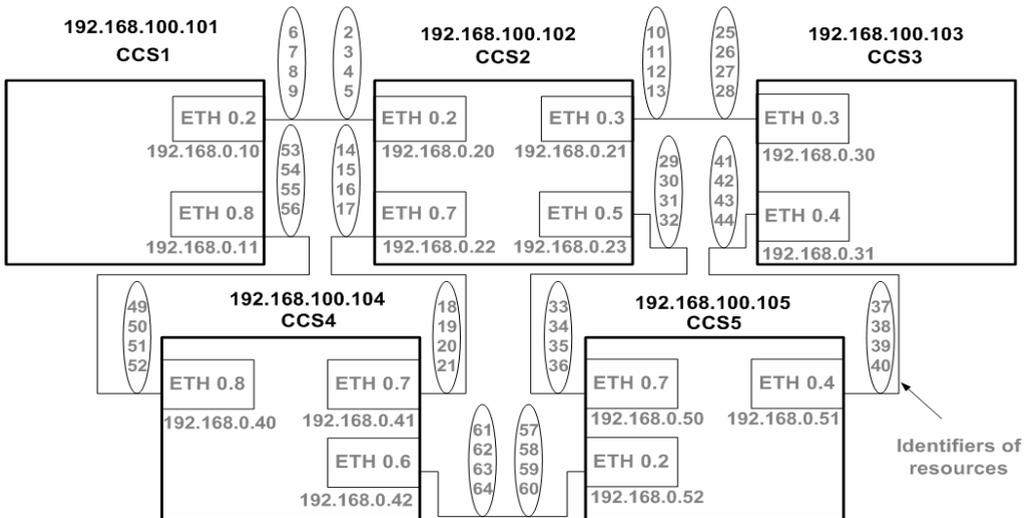


Fig. 4. Architecture of the implemented ASON/GMPLS connection control layer. ETH x.y stands for virtual network interface y based on physical network interface ethx [12,13]

3.2. TEST RESULTS

In the implemented testbed a set of performance tests regarding connection set-up and release time was carried out. Every stage of control software development was validated by execution of comprehensive functional and performance tests. Development of the software was performed into two parallel directions: optimization of connection set-up time in the connection control layer and implementation of the corresponding layers as well as inter-layer communication. For this reason executed performance tests consisted of two stages concerning connection control layer and then the three-layer ASON/GMPLS architecture (Fig. 5), in which different database solutions in Connection Control Servers were implemented:

- STAGE 1 (performance tests of ASON/GMPLS connection control layer):
 - “PSQL” variant: PostgreSQL 8.3 database,
 - “PL/pgSQL” variant: PostgreSQL 8.3 database with PL/pgSQL procedures,
 - “C/C++ struct” variant: dedicated C/C++ structures,
- STAGE 2 (performance tests of three-layer ASON/GMPLS architecture):
 - “PL/pgSQL” variant: PostgreSQL 8.3 database with PL/pgSQL procedures,
 - “C/C++ struct” variant: dedicated C/C++ structures.

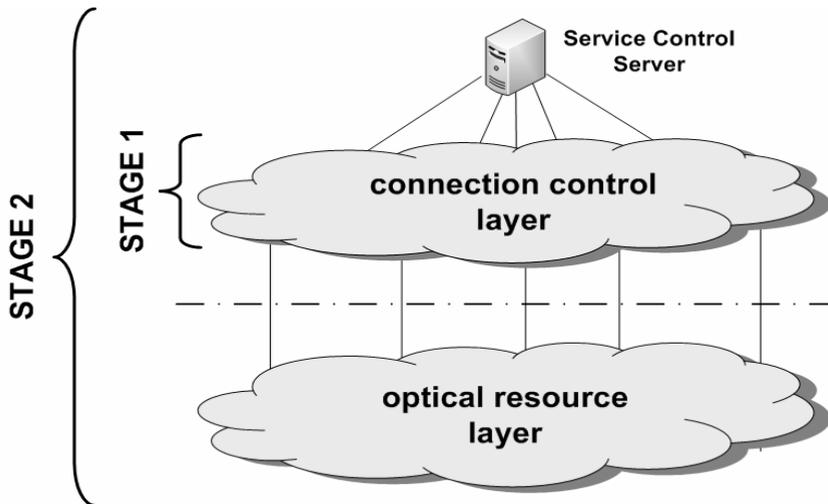


Fig. 5. Stages of software development and performance tests execution

The first implementation of connection control layer utilized PostgreSQL 8.3 database [15] (“PSQL” variant). Executed tests demonstrated limited performance of that solution. Thorough analysis of system elements revealed that the performance of the database in CCS is the most factor influencing connection set-up time. That conclusion resulted in the search for PostgreSQL database optimization. Several SQL SELECT queries per single reservation of optical resources used in initial version of the system were replaced by PL/pgSQL [15] procedures (“PL/pgSQL” variant) stored in the database and retrieved by execution of single SQL query. That optimization lead to reduction of connection set-up time but obtained results were still not sufficient and for that reason dedicated structures written in C/C++ language using STL library were implemented.

Exemplary results of each solution performance tests regarding connection set-up times between two and three Connection Control Servers are presented in Fig. 6. Presented results were obtained basing on more than 30 measurements of connection set-up time with assumed 0.95 confidence interval. Software variant utilizing PostgreSQL 8.3 database with several SQL SELECT queries is marked as “PSQL”. Variant with PL/pgSQL procedures is signed as “PL/pgSQL”. Database system using C/C++ structures is indicated as “C/C++ struct”. As it has already been mentioned, connection set-up time can be improved by about 50% by replacing several time consuming SQL SELECT queries to the PostgreSQL 8.3 database with internal PL/pgSQL procedures. However, the most significant improvement (about 10 times) can be observed when dedicated structures are used instead of regular SQL database. In that case contents of the whole database are retrieved at the start of the CCS software and stored in the C/C++ structures. Performed tests also indicated connection set-up time dependence

on the number of Connection Control Servers involved in the process of reserving optical resources. When three CCSs are involved set-up time is considerably higher than in the case of utilizing two servers independent of the used database solution.

Implementation of the three-layer ASON/GMPLS architecture (Fig. 1) was based on “PL/pgSQL” variant of the connection control layer development in stage 1 (Fig. 5). In the architecture we measured connection set-up (Fig. 7) as well as release times (Fig. 8) between two and three Connection Control Servers. Presented results were obtained basing on more than 30 measurements of connection set-up and release time with assumed 0.95 confidence interval. It is worth noting that results obtained for the three-layer architecture (Fig. 7, Fig. 8) include inter-layer communication as well as request processing times in optical resource layer. For that reason mean connection set-up and release times are slightly higher than in the case of only connection control layer. Similarly to Fig. 6, implementation of the dedicated database system (C/C++ structures, marked as “C/C++ struct”) resulted in significant request response time reduction comparing to the regular database (indicated as “PL/pgSQL”). Moreover, connection set-up time is dependent on the number of CCS servers involved in request processing.

In order to investigate the influence of the request processing in the optical resource layer on the total connection set-up time, further optimization of Resource Terminal was performed by excluding database operations. The results of the measurements (Fig. 7) with (marked as “RT DB on”) and without (marked as “RT DB off”) processing in database indicated that database system in RT has negligible impact on the total response time. Such small discrepancy can be justified if we consider that in the RT database only a small number of SQL UPDATE operations is performed. In the case of the Oracle database used in Resource Terminals, UPDATE operations are also optimized by application of fast commit mechanism [16], which reduces to the minimum the number of hard disk input/output operations.

Fully featured ASON/GMPLS architecture allows to measure also connection release time (Fig. 8). Release times are smaller than set-up times and independent of the number of the involved CCS servers, which is caused by release request processing procedure implemented in the system. Connection release request is generated by user via WWW/PHP interface provided by Service Server and sent using Diameter protocol to the CCS which initiated the process of optical resource reservation. The request is further sent to the next CCS in the path using RSVP protocol and simultaneously the procedure of releasing optical transport resources in the corresponding Resource Terminal is executed. In the connection release scenario it is assumed that the process of releasing optical resources is always successful and for this reason initiating CCS do not expect RSVP message confirming resources releasing in the next RTs in the path. After receiving Diameter protocol message confirming resource release in the corresponding RT, initiating CCS server sends final response to Service Control Server completing resource release request. The procedures of connection set-up and release in the implemented architecture are thoroughly described in [12,13].

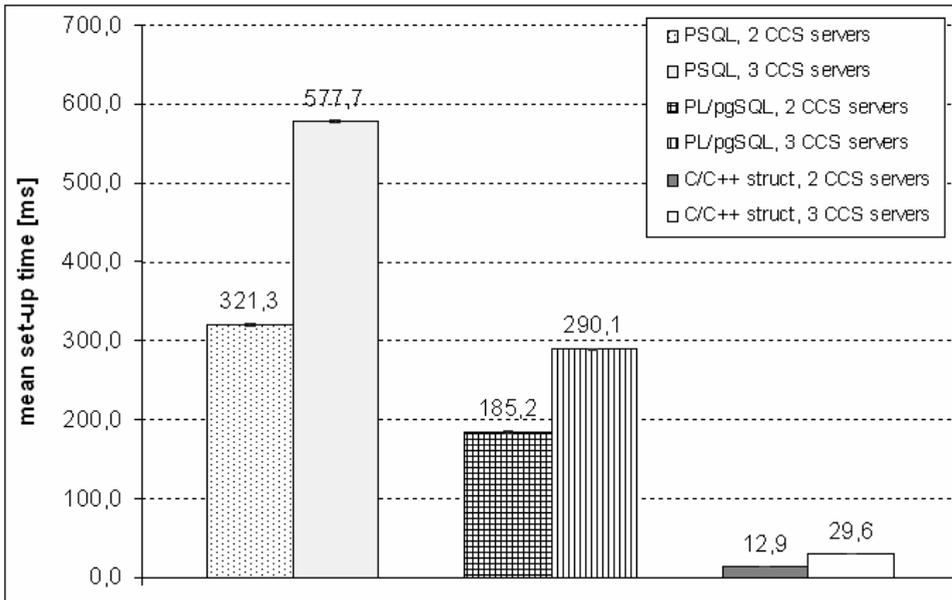


Fig. 6. Mean connection set-up time measured in the connection control layer (Stage 1, Fig. 5) for different variants of database systems in CCS; 0.95 confidence intervals are marked at the top of the bars

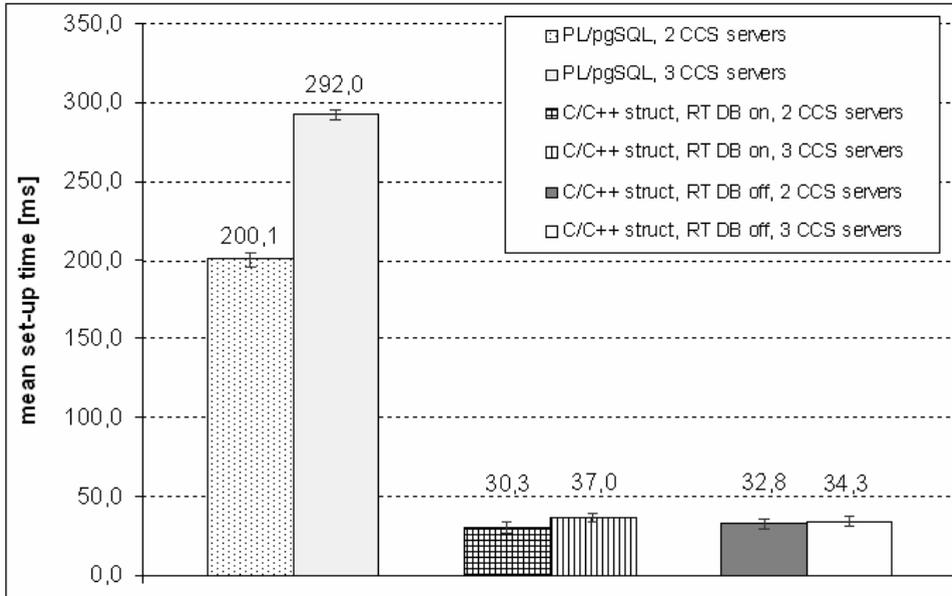


Fig. 7. Mean connection set-up time measured in the three-layer ASON/GMPLS architecture (Stage 2, Fig. 5) for different variants of database systems in CCS; 0.95 confidence intervals are marked at the top of the bars

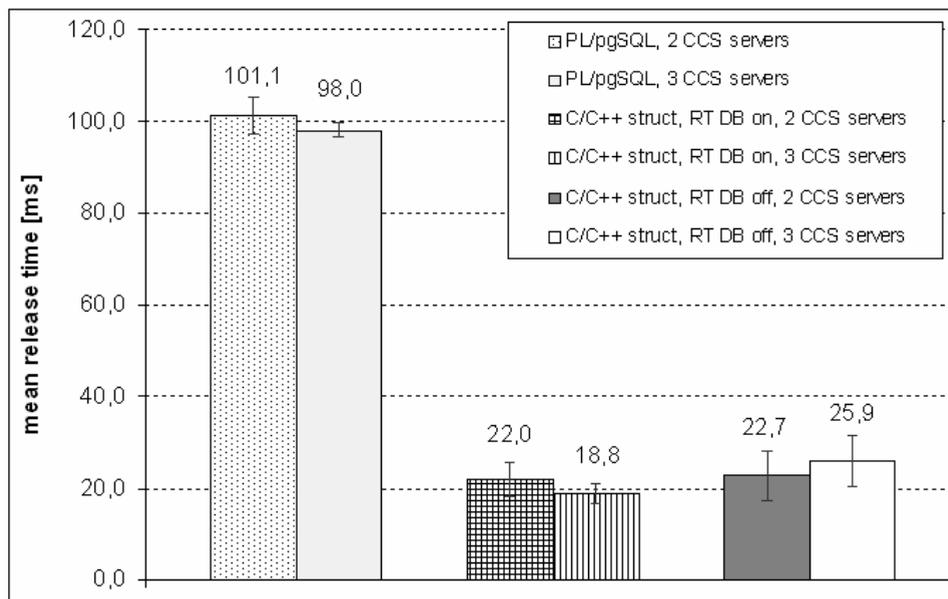


Fig. 8. Mean connection release time measured in the three-layer ASON/GMPLS architecture (Stage 2, Fig. 5) for different variants of database systems in CCS; 0.95 confidence intervals are marked at the top of the bars

Analogically to connection set-up time, implementation of the database system as dedicated structures (“C/C++ struct” variant) considerably improves connection release time. Furthermore, the database in Resource Terminal has very little influence on total resource release time due to the fact that performed SQL UPDATE also utilize Oracle fast commit mechanism.

4. CONCLUSIONS

The goal of the work described in the work was to investigate the performance of connection control layer of the ASON/GMPLS architecture implemented in laboratory testbed. Due to development of the software, performance tests were carried out in two stages: for connection control layer and then for three-layer architecture. The measures of the system performance were connection set-up time and connection release time.

Executed tests demonstrated that the performance of ASON/GMPLS architecture highly depends on the used database type and its optimization. By application of proper optimization mechanisms satisfactory performance can be achieved. In the implemented architecture connection set-up time depends on the number of involved Connection Control Servers whereas connection release time do not, which is con-

cerned with applied communication procedures. Tests performed with Resource Terminal database indicated that standard database systems are sufficient for simple queries. However, for complicated operations dedicated solutions should be applied, which is proved by CCS performance tests.

To sum up, we investigated the performance of the implemented telecommunication system and optimized connection set-up and release times. Achieved results are sufficient for telecommunication control systems [17]. However, in the ITU-T recommendation [17] performance metrics include propagation delay, which is not considered in the implemented testbed environment, due to the fact that this delay component depends on the distance and can be easily included in the calculations for real network environments. For example, the distance of 1000 km introduces propagation delay of about 5ms for one direction.

Performance aspects are crucial for telecommunication control systems, particularly for providing real time services. For this reason we are planning to extend our research in this area. One of the parts of the further work with the implemented testbed will include performance tests according to the ETSI methodology described in [18].

ACKNOWLEDGEMENTS

This work was partially supported by the Ministry of Science and Higher Education, Poland, under the grant PBZ-MNiSW-02-II/2007.

REFERENCES

- [1] ITU-T Recommendation Y.2012, *Functional Requirements and architecture for next generation networks*, April 2010.
- [2] ITU-T Recommendation G.8080/Y.1304, *Architecture for the Automatically Switched Optical Network (ASON)*, June 2006.
- [3] MANNIE E., *Generalized Multi-Protocol Label Switching (GMPLS) Architecture*, IETF RFC 3945, October 2004.
- [4] ITU-T Recommendation G.7713.2/Y.1704.2, *Distributed Call and Connection Management: Signalling mechanism using GMPLS RSVP-TE*, March 2003.
- [5] AWDUCHE D., et al., *RSVP-TE: Extensions to RSVP for LSP Tunnels*, IETF RFC 3209, December 2001.
- [6] *Oracle Database 10g Express Edition*, <http://www.oracle.com/technology/products/database/xe/index.html>
- [7] BRADEN R., et al., *Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification*, IETF RFC 2205, September 1997.
- [8] KACZMAREK S., NARLOCH M., MLYNARCZUK M., SAC M., *Communication protocol extensions for control of ASON/GMPLS network*, accepted for publication at Polish KSTiT 2011 Conference and as paper in *Telecommunication Review and Telecommunication News (Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne)*, 2011 (in Polish).

- [9] KACZMAREK S., MŁYNARCZUK M., WALDMAN M., *RSVP-TE as a reservation protocol for optical networks*, PWT 2010, XIV Poznań Telecommunications Workshop, Poznań, Faculty of Electronics and Telecommunications at Poznan University Technology, 2010, pp. 28–31.
- [10] FREDETTE A., et al., *Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems*, IETF RFC 4209, October 2005.
- [11] CALHOUN A., et al., *Diameter Base Protocol*, IETF RFC 3588, September 2003.
- [12] KACZMAREK S., NARLOCH M., MŁYNARCZUK M., SAC M., *Next generation services and teleinformation networks - technical, application and market aspects; Network architectures and protocols*, PBZ-MNiSW-02-II/2007/GUT/2.6, Gdansk, December 2010 (in Polish).
- [13] KACZMAREK, S., NARLOCH M., MŁYNARCZUK M., SAC M., *The realization of NGN architecture for ASON/GMPLS network*, paper submitted to Journal of Telecommunications and Information Technology.
- [14] KACZMAREK S., MŁYNARCZUK M., WALDMAN M., *The realization of ASON/GMPLS Control Plane*, Information Systems Architecture and Technology, System Analysis Approach to the Design, Control and Decision Support, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2010, pp. 313–324.
- [15] *PostgreSQL: The world's most advanced open source database*, <http://www.postgresql.org/>
- [16] *Oracle Database Concepts 10G*, http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/process.htm#sthref1530
- [17] ITU-T Recommendation Y.1530, *Call processing performance for voice service in hybrid IP Networks*, May 2004.
- [18] ETSI Technical Specification TS 186 008: *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IMS/NGN Performance Benchmark*, Part 1–Part 3, v1.1.1, October 2007.

Mariusz GŁĄBOWSKI*, Michał STASIAK*

INTERNAL BLOCKING PROBABILITY CALCULATION IN SWITCHING NETWORKS WITH ADDITIONAL INTER-STAGE LINKS

Switching networks form a base for the operation of many systems and devices, including that of exchanges or routers that constitute nodes in telecommunications networks. It is their effectiveness in performance that is decisive in enabling the maximum amount of traffic that a network can carry. Most of switching networks used in network nodes are blocking networks. In order to reduce the blocking probability in switching networks with multi-service traffic, this work proposes the application of overflow links. This mechanism requires a slight modification in the physical architecture of a switching network and is based on an application of switches with an appropriately higher number of inputs and outputs. The present chapter also proposes the first analytical approach to blocking probability calculation in multi-service switching networks with overflow links. In order to evaluate the accuracy of the proposed analytical method, the results obtained on the basis of the proposed method are compared with simulation results.

1. INTRODUCTION

Parameters of modern telecommunications networks depend on the effectiveness of switching devices that constitute network nodes and are their components. These devices are usually digital telephone switching centres, switches and routers. The operation of switching devices is based on switching networks. Switching networks, in turn, can be divided into blocking and non-blocking switching networks [1],[2]. Due to costs involved, most devices employ blocking networks. An application of blocking networks is, however, followed by a loss of

*Chair of Communication and Computer Networks, Poznań University of Technology, pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland.

part of traffic offered to the network due to the occurrence of the phenomenon of internal blocking. There are methods that can effect a considerable decrease in the internal blocking in switching networks. Such methods include, for example, the application of overflow links [3], re-packing and re-arranging [2]. The techniques based on re-packing and re-arranging do not interfere with the physical structure of a network. A decrease in the internal blocking results from the application of appropriate control algorithms that are decisive in adopting the way of setting up a connection by the system. A disadvantage of such an approach is the following increase in the load of devices that control switching networks. This results from a high complexity of algorithms for setting up connections. A solution based on an application of overflow links entails a change in the structure of a network following the application of additional links between switches of the same stage (section). An introduction of overflow links leads then to an increase in the number of inputs and outputs of switches of the stage in which overflow links have been applied. The idea of overflow links was applied for the first time in Pentaconta dial telephone switching systems [3] in the 1960s through the 1980s. In the switching networks of the Pentaconta switching centre overflow links were used in the switches of the first stage. This resulted in a several percent decrease in the probability of internal blocking [4]. The possibility of an introduction of overflow links was also seriously considered for digital switching networks [5],[6].

Modern network devices employ networks that carry multi-service traffic. The simulation studies that have been carried out so far [7],[8] indicate that the application of overflow links in multi-stage switching networks leads to a substantial increase in their traffic effectiveness because of the elimination of the internal blocking.

This chapter attempts to construct an analytical model of a multi-service switching network with an ideal system of overflow links. The model assumes that the capacity of the overflow links is high enough to eliminate the phenomenon of blocking in such links entirely. Such an approach stems from the observations obtained throughout the simulation studies and experiments carried out in [7],[8], where it was observed that a two-fold increase in the capacity of overflow links, as compared to inter-stage links, led to a virtual elimination of internal blocking.

The chapter is organized as follows: Section Two provides a description of the structure and the operation of a three-stage Clos-network with overflow links in the first stage. Section Three presents a model of a switching network with overflow links. Section Four presents the results of a comparison of the analytical calculations with the results of the simulation experiments for a selected multi-stage switching network with overflow links. Section Five sums up the results of the investigations presented in the chapter.

2. SWITCHING NETWORKS WITH OVERFLOW LINKS

Consider the blocking model of Clos switching network presented in Fig. 1. The network is constructed of n symmetrical switches $n \times n$ links in each of the stage. Each link has a capacity of f BBUs (Basic Bandwidth Units [9]). The network is offered multi-rate traffic that is composed of a number of call stream classes. A call of each class requires a different number of BBUs to set up a connection. The call stream of each class is a Poisson stream. It was adopted in the study that the point-to-group selection had been applied in the network. The outputs of the switching network were grouped into outgoing directions in such a way that each i^{th} output of each of the switches in the last stage belonged to the i^{th} output direction (Fig.1).

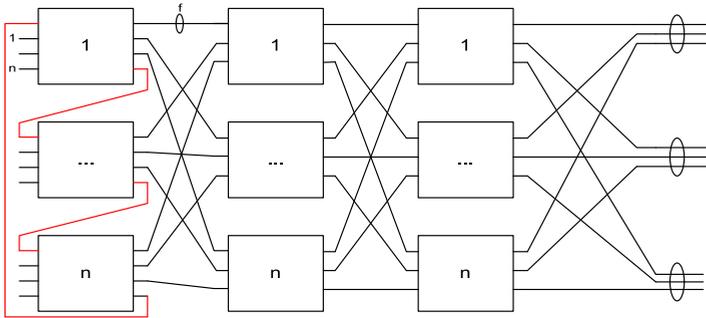


Fig. 1. Clos switching network with a system of overflow links in the first stage.

In the point-to-group selection, once a call appears in a given input of the switch of the first stage, the controlling algorithm selects the switch of the last stage that has a free link in the demanded direction. Then the controlling algorithm attempts to set up a connection between the selected switches of the first and the last stages. If the connection between these switches cannot be set up, the controlling algorithm attempts to set up a connection with another switch of the last stage that has a free output in the given direction. This procedure is repeated until the connection can be set up or all switches that have outputs in the given direction have been checked. If, after checking all available switches, the connection still cannot be set up, the controlling algorithm rejects the call due to the internal blocking in the network. If all outgoing links of a given direction are busy, the controlling algorithm rejects the call on account of the external blocking in the network.

In a three-stage Clos network, overflow links can be introduced in a number of ways [7]. In the course of the study it was adopted that a system of overflow links would be introduced to the first stage in such a way that the overflow links would connect an additional output of a given switch with an additional input of the neighbouring switch of the same stage (Fig. 1). The output of the last switch is con-

nected with an additional input of the first switch. This system of overflow links can be applied to each stage of the network. The results of the simulation study on three-stage Clos networks [7],[8] clearly and unambiguously indicate that the highest decrease in the internal blocking results from the introduction of an overflow system in the first stage of the switching network. Moreover, it has been observed that the higher increase in the capacity of the overflow links, the lower internal blocking occurs in the system. An application of overflow links with the capacity that is two-fold higher than the capacity of inter-stage links stabilizes the internal blocking at a level of insignificant values as compared to the external blocking (at least by a lower order of magnitude). Ultimately, the network becomes a quasi-non-blocking network.

3. ANALYTICAL MODEL WITH OVERFLOW LINKS

3.1. BASIC ASSUMPTIONS

In this chapter the authors propose the first approach to a determination of the internal, external and total blocking probability in multi-stage switching networks with overflow links. The basis for the proposed approach is to use an effective availability method – the PGBMT method (Point-to-Group Blocking for Multichannel Traffic) [10],[11],[12], worked out for switching networks without overflow links. The idea of the proposed method is based on an introduction of certain changes in the determination of the probability graph that is used for a determination of the internal blocking.

In order to fully present the proposed method for a determination of the blocking probability in networks with overflow links, the next part of the chapter will be devoted to the basic assumptions of the PGBMT method (Section 3.2), and then, in Section 3.3, a method for a determination of the parameters of the probability graph in the switching networks under consideration will be proposed.

3.2. EFFECTIVE AVAILABILITY METHOD – THE PGBMT METHOD

The effective availability method – the PGBMT method – makes it possible to determine analytically the blocking probability in a switching network servicing multi-rate traffic. The method is based on the idea to reduce the multi-stage network model to the single-stage network model – the non-full-availability group. Additionally, it is assumed that the capacity of the non-full-availability group is equal to the capacity of the output group of the network. Thus, to determine the blocking probability in the network, formulas that determine blocking in a non-full-availability group for a given value of the effective availability parameter are used. This parameter is determined on the basis of the structure and the load of the switching network. Effective availability

is then defined as such an availability in the network for which the blocking probability is equal to the blocking probability in a non-full-availability group. Additionally, the assumption is that the parameters of the traffic stream are the same, both for the switching network and the group.

To determine the effective availability in the PGBMT method it is necessary first to determine, separately for each class of call, the so-called equivalent network. Equivalent network is a network with the same topological structure as a multi-service network in which the capacity of links is equal to 1 BBU. The next assumption is that the network services only one traffic class and that the load in each link of the network is equal to the blocking probability of a given traffic class in a real network. The next step is to construct the probability graph (on the basis of the equivalent network) and to determine the effective availability for a given traffic stream. Effective availability for calls of class i can be determined on the basis of the following formula:

$$d_e(i) = [1 - \pi(i)]v + \pi(i)\eta Y_1(i) + \pi(i)[v - \eta Y_1(i)]y_z(i)\sigma_z(i) \quad (1)$$

where: $d_e(i)$ – effective availability for stream of class i in the equivalent network, $\pi(i)$ – probability of direct non-availability of the switch of the last stage for calls of class, v – number of links in a given direction of the equivalent network, $Y_1(i)$ – average fictitious traffic carried by the switch of the stage, $\sigma_z(i)$ – secondary availability coefficient that determines this part of the switches of the last stage of the of the equivalent network that is secondarily available for calls of class i [10].

The probability $\pi(i)$ results from an analysis of the probability graph. For the exemplary 3-stage Clos network presented in Fig. 2a:

$$\pi(i) = \{1 - [1 - E(i)]^2\}^k \quad (2)$$

where $E(i)$ is the load of the equivalent network, equal to the blocking probability $E(i)$ for calls of class i in the inter-stage link with the capacity f . This parameter is determined on the basis of the Kaufman-Roberts model [14], [15]:

$$n[P_n]_f = \sum_{i=1}^M A_i t_i [P_{n-t_i}]_f, \quad (3)$$

$$E(i) = \sum_{n=f-t_i+1}^f [P_n]_f, \quad (4)$$

where: n – the number of busy BBUs in the full-availability group (link) with the capacity f , A_i – traffic of class i offered in the inter-stage link, t_i – number of BBUs de-

manded by a call of class i , M – number of offered call classes, $[P_n]_f$ – occupancy distribution in the full-availability group, i.e. the occupancy probability of n BBUs in a link with the capacity f BBUs.

The parameter $\sigma_z(i)$ for the three-stage Clos network in Formula (1) is determined by the following formula:

$$\sigma_3(i) = 1 - E(i) \tag{5}$$

According to the PGBMT method, the total blocking probability in the switching network can be determined in the following way:

$$E_i(i) = E_e(i) + E_i(i)[1 - E_e(i)] \tag{6}$$

where the internal blocking probability $E_i(i)$ and the external blocking probability $E_e(i)$ in the switching network is determined by the following formulas:

$$E_i(i) = \sum_{s=1}^{v-d(i)} \frac{P(i,s)}{1 - P(i,0)} \left[\frac{\binom{v-s}{d(i)}}{\binom{v}{d(i)}} \right] \tag{7}$$

$$E_e(i) = P(i,0) \tag{8}$$

The distribution $P(i,s)$ in Formulas (7) and (8) is a distribution of free links in the limited-availability group [16]. This group is a model of separated links that approximates output links of the switching network in the PGBMT method. The distribution $P(i,s)$ determines the probability of an event that s output links in a given direction (from among V all links of the direction) can service a call of class i :

$$P(i,s) = \sum_{n=0}^V [P_n]_V P(i,s | V - n) \tag{9}$$

where $[P_n]_V$ is the occupancy distribution in a limited-availability group:

$$n[P_n]_V = \sum_{i=1}^M A_i t_i \zeta_i(n - t_i) [P_{n-t_i}]_V \tag{10}$$

The parameter $\zeta_i(n)$ is the so-called conditional probability of passing (conditional. transition probability) that, in the limited-availability model, is determined by the following formula:

$$\zeta_i(n) = F(V - n, k, f, 0) - F(V - n, k, t_i - 1, 0) / F(V - n, k, f, 0) \quad (11)$$

The parameter $F(x, k, f, t)$ in Formula (11) is the number of distributions x of free BBUs in k links, each with the capacity f BBU. The next assumption is that in each free link t free BBUs are allocated. Note that the total capacity of the group is equal to $V=kf$. The probability $P(i, s | x)$ in (9) is the conditional distribution of free links in the limited-availability group determined with the assumption that x links in the group are free:

$$P(i, s | x) = \binom{k}{s} \sum_{w=st_i}^{\varphi} F(w, s, f, t_i) F(x - w, k - s, t_i - 1, 0) / F(k, x, f, 0), \quad (12)$$

where $\varphi = sf$, if $x \geq sf$, and $\varphi = x$, if $x < sf$.

The limited-availability group model determined by Formulas (10) and (11) and applied to the PGBMT method makes it possible to evaluate (and assess) the internal and external blocking probability.

3.3. THE PROPOSED METHOD FOR MODELLING NETWORK WITH OVERFLOW LINKS

To determine the internal blocking probability in the network with overflow links on the basis of the PGBMT method it is necessary to first determine the probability graph that corresponds to the modified structure of the network. The probability graph for a three-stage Clos network, without overflow links, is presented in Fig. 2a. The graph shows all connection paths between switches of external stages. The introduction of an overflow link increases the number of possible connection routes in the graph (Fig. 2b) because the switches in the second stage k_1, k_2, \dots, k are additionally available both directly and indirectly – from the point of view of a given switch A in the first stage – through another switch in the first stage (switch C) that is connected to switch A by an overflow link AC. Assuming that the overflow link is lossless, the graph shown in Fig. 2b can be converted to the form presented in Fig. 2c. Therefore, on the basis of the graph presented in Fig. 2c, the non-availability probability of a given switch of the third stage $\pi(i)$ can be determined by the following formula:

$$\pi(i) = 1 - \{[1 - E^2(i)][1 - E(i)]\}. \quad (13)$$

The secondary availability coefficient for the switching network with overflow links (determined by the graph in Fig. 2c) can be determined by the following formula:

$$\sigma_3(i) = 1 - E^2(i), \quad (14)$$

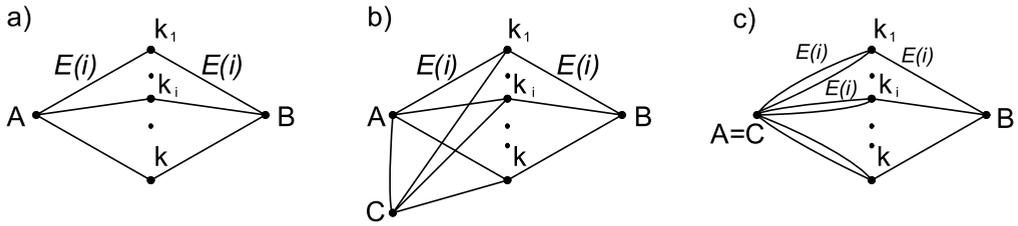


Fig. 2. Probability graphs for Clos switching networks a) without overflow links, b) and c) networks with overflow links

The parameters $\pi(i)$ and $\sigma_3(i)$, determined on the basis of Formulas (13) and (14), are the requisite base for determining the effective availability in the network with overflow links on the basis of Formula (1).

With the parameter of the affective availability for calls of individual traffic classes at hand it is possible to initially determine the internal blocking probability (Formula (7)) and external blocking probability (Formula (8)) in a switching network with overflow links that services multi-service traffic.

4. COMPARISON OF THE SIMULATION AND ANALYTICAL RESULTS

In order to determine the accuracy of the proposed method for modelling networks with overflow links the results of the analytical calculations of the blocking probability were compared with the results of the simulation experiments. A three-stage Clos network was investigated. The network was composed of 4×4 switches with a system of overflow links (Fig. 1). It was adopted that the capacity of input, output and inter-stage links of the switching network was 30 BBUs. Another assumption was that the capacity of the overflow links was infinitely high. The network was offered multi-rate traffic that was composed of three classes that demanded 6, 2 and 1 BBUs, respectively.

Figure 3 shows the percentage decrease in the value of the internal blocking probability – resulting from the introduction of overflow links – for each traffic class in the case of the analytical calculations (Fig. 3a) and in the case of the simulation experiments that followed (Fig. 3b). It is noticeable that the nature of the changes in the internal blocking probability remains the same both for the analytical calculations and for the simulation experiments. This proves that the modified PGBMT method allows to initially estimate and assess changes in the internal blocking probability in a switching network with overflow links.

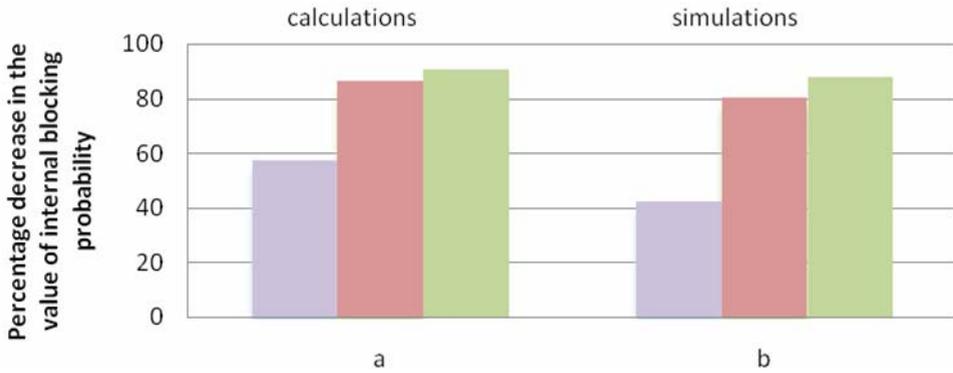


Fig. 3. Percentage decrease in the internal blocking, a) calculations, b) simulation

Figure 4, in turn, presents a comparison of the results of analytical calculations of the total blocking probability with the data obtained during the simulation experiments. The results of the experiments are presented with 95% confidence interval, determined on the basis of the t-Student distribution, for 10 series with 100,000 calls of the oldest class each.

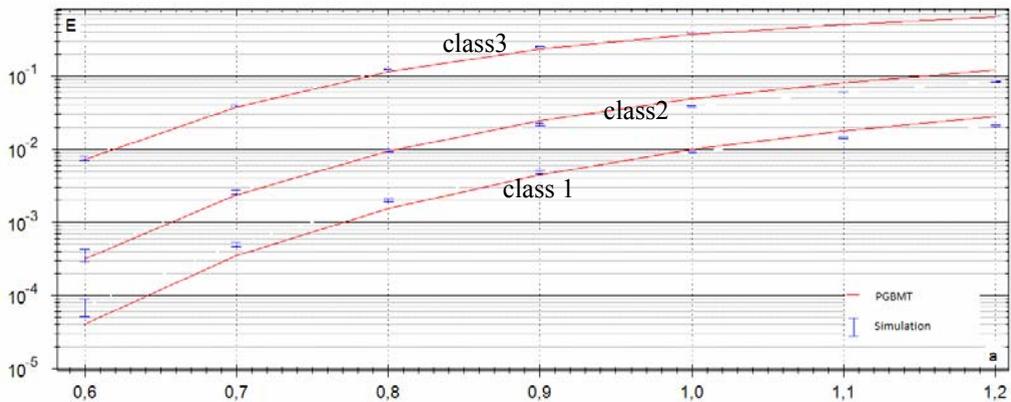


Fig. 4. The total blocking probability in the switching network with overflow links

5. CONCLUSIONS

The introduction of overflow links between each two neighbouring switches of the first stage is followed by a substantial decrease in the blocking probability in the switching network. For the analytical estimation of the blocking probability in networks with overflow links, the chapter proposes an application of the PGBMT method. It should be highlighted that, according to the best knowledge of the authors,

this is the first attempt to model multi-service switching networks with overflow links analytically. The obtained results make it possible to initially evaluate the effectiveness of the introduction of overflow links to Clos switching networks. It turns out that a connection of two neighbouring switches of the first stage by an overflow link leads to at least 50% decrease in the value of the internal blocking probability. The results show then that there is an increasing need for further studies on switching networks with overflow links to be carried out with the aim of developing even more effective overflow systems and of increasing the accuracy and precision of analytical methods.

REFERENCES

- [1] CLOSE C., *A study of non-blocking switching networks*, Bell System Technical Journal, Vol. 32, No. 2, 1953, 406–424.
- [2] KABACIŃSKI W., *On nonblocking switching networks for multichannel connections*, IEEE Transactions on Communications, Vol. 43, No. 1, 222–224.
- [3] FORTET R., (Ed.), *Calcul d'orange, Systeme Pentaconta*, L.M.T Paris, 1961.
- [4] STASIAK M., *Computation of the probability of losses in commutation systems with mutual aid selectors*. Rozprawy Elektrotechniczne, Vol. XXXII, No. 3, 1986, 961–977.
- [5] INOSE H., SATIO T., KATO M., *Three-stage time-division switching junctor as alternate route*. Electronics letters, Vol. 2, No. 5, 1966, 78–84.
- [6] KATZSCHNER L., LORCHER W., WEISSCHUH H., *On a experimental Local PCM Switching Network*, Proc. International Seminar on Integrated System for Speech, Video and Data Communication, Zurich, 1972, 61–68.
- [7] STASIAK M., ZWIERZYKOWAKI P., *Performance Study in Multi-rate Switching Networks with Additional Inter-stage Links*, In: Proc. The Seventh Advanced International Conference on Telecommunications (AICT 2011), St. Marteen, Holland, 2011.
- [8] STASIAK M., ZWIERZYKOWAKI P., *Multi-service switching networks with overflow links*, Image Processing and communications, Vol. 15, No. 2, 2010, 61–71.
- [9] ROBERTS J., MOCCI V., VIRTAMO I., (Eds.), *Broadband Network Teletraffic, Final Report of Action COST 242*, Berlin, Commission of the European Communities, Springer, 1996.
- [10] STASIAK M., GŁĄBOWSKI M., *Point-to-point blocking probability in switching networks with reservation*, Proc. 16th International Teletraffic Congress, Vol. 3a, Edinburgh, 1999, 518–528.
- [11] STASIAK M., *Systemy ze stratami w sieciach z ruchem zintegrowanym*, Wydawnictwo Politechniki Poznańskiej, Poznań, 1995.
- [12] GŁĄBOWSKI M., *Modelowanie systemów multi-rate ze strumieniami zgłoszeń BPP*, Wydawnictwo Politechniki Poznańskiej, Poznań, 2009.
- [13] LEE C., *Analysis of switching networks*, Bell Systems Tech. J., Vol. 34, No. 6, 1955, 1287–1315.
- [14] KAUFMAN J., *Blocking in a shared resource environment*, IEEE Transactions on Communications, Vol. 29, No. 10, 1981, 1474–1481.
- [15] ROBERTS J., *A service system with heterogeneous user requirements-application to multi-service telecommunications systems*, In: Proceedings of Performance of Data Communications Systems and their Applications, Pujolle G. (Ed.), North Holland, Amsterdam, 1981, 423–431.
- [16] STASIAK M., *Blocking probability in limited-availability group carrying mixture of different multichannel traffic streams*, Annals of Telecommunications, 1993, Vol. 48, No. 1–2, 71–76.

César DE LA TORRE*

DESIGN OF SECURE AND COST EFFICIENT NETWORKS TO SUPPORT CLOUD COMPUTING APPLICATIONS

Secure and cost efficient networks to support cloud computing services over public infrastructure, is the next step for a network designers. The network interfaces become the best point to analyze the network performance; it will enable administrators to inventory all applications and baseline their performance, before moving into private or public cloud. To guarantee optimal service visibility, is important identify and classify all deployed applications and services running on the network, design a map dependencies among the elements involved in delivering services, measure and classify traffic along each service delivery path and measure cost of each service delivery path. Finally, integrate security technologies in critical networks points to enforce security policies. As business applications recede from desktops into data centers and from there into great beyond that is public cloud, still the principal task of IT administrators is guarantee visibility and delivers high performance in network core and network edges for optimal access to services.

1.INTRODUCTION

In the past, principal ideas of IT security have concentrated primarily on perimeter defenses firewalls, proxies, and content filtering. The idea of cloud computing has dramatically changed before mentioned concepts, as effect of new ideas to implement corporate IT services over the Internet. Cloud computing is a general term for an old idea that evolves delivering hosted services using remote rather than local computing power and storage, to run virtualized applications that users access via corporate WAN/LAN or Internet, Table 1. Clouds are classified in publics and private. A public cloud sells services to anyone on the Internet and the administration is out of enterprise's control, besides a private cloud is a proprietary data center or network that supplies hosted services to a limited number of people.

* ESPE, Escuela Politécnica del Ejército, Av. Gral. Rumiñahui S/N, Quito, Ecuador.

Private or public, the principal goal of cloud computing is to offer easy, scalable access to applications in real time, at lowest possible cost and enforcing the security paradigms.

Table 1. Distribution of software and hardware in a infrastructure

CORPORATE LOCATIONS			EXTERNAL ADMINISTRATION
User office/Desk	Remote office	Data center	Third-party location
Desktop Computing	Department Servers	Consolidated Servers (Private Cloud)	Public Cloud
Client Agents	Local services	Virtualized applications and services	Cloud services

The process generated by virtualized applications running in hosted servers and accepting network connections are denominated cloud services. As a cloud service visibility intends of active open connections that virtualized application accept and respond to satisfy user requirements.

Note the importance of cloud security for a visibility of virtualized applications. For a facility of analysis we assume that cloud security is the sum of cloud components. Where components of cloud security are: network security, application security, service security, user security, and so on.

Service visibility could be interrupted as result of network attacks or a excessive traffic volume. Therefore, for high service availability is essential the optimal distribution and allocation of IT security elements which increase bandwidth efficiency.

1.1. THE CHALLENGE OF NETWORKING CLOUD SECURITY

While we consider that links capacities in all path routes, that connect corporate networks to cloud service provider, have enough capacity to satisfy all user requirements; these are not important for data flow the influence of traffic congestion, hops and distances, packet delivery resiliency and retransmissions.

Assuming that corporate network security is the most important factor of cloud security, the procedures and tools applied to attend local security issues will extend to corporate services deployed in public clouds, without any additional risk for corporate security. It is a fact that security of selected service provider is better than security of corporate network.

The Unified Threat Management (UTM) elements has multiple functions that contribute to avoid degradation of cloud performance, such as detect and block threats, inte-

grated intrusion prevention systems, eliminate spam mails and URLs spam, content filtering; in addition integrate services like identity management, user authentication and administrative controls; therefore are principal components to optimize network flows. The scope of this work is the *optimal location of K UTM elements* in a given network, where the function of UTMes is increase *bandwidth efficiency*, because detect and eliminate unauthorized and undesired network traffic; it is *contribute to improve cloud services visibility*.

2. DEFINITIONS

Virtualized applications in a cloud consist of three elements: production, transport and consumption. The production is related to computing and storage process, consumption is the work performed in client devices and transport is the networking technology that binds production to consumption. In next paragraphs we assume that production and consumption are the result of agents which have the ability to interconnect themselves.

When virtualized application process (agents) are liable to accept and respond the requirements of other agents denominate service visibility; this parameter is principal to evaluate the optimal deployment of virtualized application in a cloud, *see Fig. 1*.

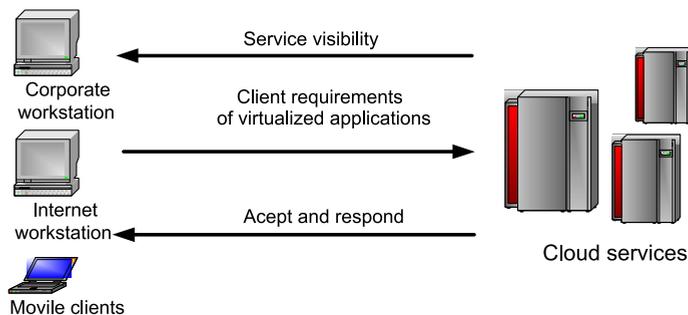


Fig 1. Service visibility

2.1 WAN OPTIMIZATION TECHNIQUES

Proactive management of virtualized application performance over LAN/WAN require of new components, such as: *intrusion prevention and detection network systems, tools for security analysis, and tools for monitoring, analyzing network and application performance*.

IT administrators to evaluate optimal network performance considered the following criteria: *average improvement in network throughput, average improvement in*

application availability and average improvements in application response time. Although, average improvement in application response time is a critical element in network design is not considered in present work, because could increase the network vulnerability and it is possible to offset this parameter through the optimization of the other two parameters. Consequently, the benefit of improving network performance is calculated as the minimization of threats in network flows. It is the reduction of total cost of transmission between nodes trough discard of unsafe packets.

The gain of this work is to *optimize the visibility of services, through the optimization of network performance and bandwidth efficiency.* Whereas the optimization of network traffic on all links is a solution to the need for greater bandwidth and security in the end nodes.

$$\max_{flow} Benefit = \min_{networkThreats} f(\text{givenNetwork}, \text{matrixFlow}, \text{networkThreats})$$

2.2 PRINCIPAL INDICATORS FOR THREATS DETECTION

The principal indicators applied to threat detection are: *false alarm rates and probability of detection rate*; of course a unified threat management system is better when the number of false alarm is close to 0 and the probability of hit detection is maximum. For next paragraphs analyses assume that the *false alarm rate is zero and the probability of hit detection rate is 1 and constant in all links.* So, there exists an invariable hit detection rate for the whole network.

Definition 1. Given a set of cloud services P , where $P=(p_1, \dots, p_m)$ and m is the total number of cloud services, exist a set of security policies O , where $O=(o_1, \dots, o_n)$ and n is the total number of security elements, the result of joints the both sets $P \cup O$ is a set of communication agents $A=(a_1, \dots, a_m)$ resistant to any network attacks.

2.3 GRAPHS AND NOTATIONS

Indeed exist many different families of graphs, we will analyze only the *random graphs* because are the most common graphs found in practice and has a wide application for computer network analysis. In random graphs arcs connect vertices without any structural regularity and the probability of an arc connect two vertices is independent of the vertices.

Since all the graphs are directed in this work, because network connections will be most probably bidirectional and the bandwidth in links are different for each way (asymmetric). Therefore, in real network computers the different forms how two agents in a network connect can be represented as a graph isomorphism.

Let direct graph $G=(V,E)$ consist of a finite non-empty set V of vertices and a binary relation E , i.e. a subset $E \subseteq V \times V$. The elements E are called *edges or arcs*. An

edge $(i,j) \in E$ is considered to be oriented from i to j . An undirected graph is a graph whose edges set E is symmetrical, i.e $(i,j) \in E$ and $(j,i) \in E$. In next paragraphs the term graph will be used to refer to a direct graph, because consider that undirected graphs are just a particular case of direct graphs.

Given a graph $G=(V,E)$, E can be represented by an adjacency matrix $Adj(G)=A$ with size $|V| \times |V|$, where $a_{ij}=1$ if $(i,j) \in E$ and $a_{ij}=0$ if $(i,j) \notin E$.

2.4 NETWORK DEFINITIONS

It will be necessary to introduce some specific notation to be used in our network definitions.

Definition 2. Let each agent $a_i \in A$ represent one of two different types of process:

- 1) The traffic volume of services visibility (processes) generated by virtualized applications.
- 2) The traffic volume of user requirements for virtualized application services in end nodes.

The whole network is represented as a direct graph $G=(V,E)$, where $V=\{v_1, \dots, v_n\}$ is the set of vertices, E is the set of edges $E \subseteq V \times V$ and represents network connections. Links between vertices are considered directed for the sake of generality, because connections will be most probably bidirectional, and the bandwidths in links are different for each way (*asymmetric*).

Definition 3. Let E is a set of links connections between vertices V , such for all links between vertices (i,j) the following statement holds:

- 1) If exist a link between vertices (i,j) , then $e_{ij}=1$ and $e_{ij} \in E$
- 2) If not exist a link between vertices (i,j) , $e_{ij}=0$ and $e_{ij} \notin E$

We consider a general wide area network, where *internal nodes are gateways* and *external (border) nodes are servers, clients or gateways to different subnets*. The agents are deployed in external network nodes V and exist for each ordered pair of communications agents (a_i, a_j) a static route path in the graph G .

Definition 4. Let a graph $G=(V,E)$ can be represented by adjacency matrix $Adj(G)=A$ with size $|V| \times |V|$, then we assume for element $a_{ij} \in A$ is true the following statements:

- 1) The routes are statics in time.
- 2) The communication between agents is trough the shortest path.
- 3) The time delay is 0 and not depended of hops and distances.
- 4) The capacity of each link is enough for any routed traffic, even if others communication process (a_b, a_i) are using the same link at the same time.
- 5) In each vertex v_i is deployed only one agent a_i .

The (*static*) route between two agents, see *Definition 4*, from *external vertex* v_i to v_j is denoted as $Route(a_i, a_j)$, where (a_i, a_j) represent the respective agents deployed in

respective vertices, and includes at least one internal node or gateway (*router or UTM*). The internal nodes form a core network represented as a direct graph $G_c=(W_c,R_c)$, where $(W_c \subset V, R_c \subset E)$.

Definition 5. Let a graph $G_c=(V,E_c)$; represent a core network, the following statements are hold:

- 1) The set $R = \left(r_{lk;l \neq k} \right)_{l,k=1}^{l,k=n}$ represent a set of static routes between internal nodes l and k .
- 2) For each pair of agents $(a_i, a_j) \in A$ is only one static route $r_{ik} \in R$
- 3) Because all links in our graph are directed, exist a routes $(r_{ik}, r_{ki}) \in R$

3. THE FORMAL MODEL

For each ordered pair of agent nodes $(a_i, a_j) \in A$, all the UTM will have the same hit detection ratio $H_r(a_i, a_j)$ and represent the fraction of total transmitted packets to be discarded by the UTM element because represent a security network risk (threats), in consequence $H_r(a_i, a_j) < 1$. The hit detection ratio determines how many of the packets that reach a gateway, in which is installed a UTM element, are discarded and not forwarded to their destination.

The brute average traffic that goes from agent a_i to agent a_j is the amount of packets in the network that are originated in node a_i and have a node a_j as destination. The traffic is measured in *packets/seconds*, and is denoted as $BruteTr(a_i, a_j)$. The net average traffic, denoted as $NetTr(a_i, a_j)$, that goes from agent a_i to agent a_j is:

$$NetTr(a_i, a_j) = (1 - H_r(a_i, a_j)) * Br$$

Although, packets flowing in Internet differ in dimension, for this work assumes that all packets have the same size. By constant C denotes the cost of transmitting a packet through a link r_{lk} . Thus, the cost of transmitting a packet through a *route path* will only depend on the length of the route $|Route(a_i, a_j)|$. Otherwise, routes will be at least of length 2 according to our premises.

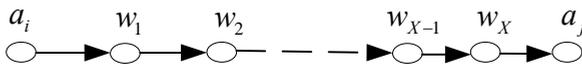


Fig. 2. Sample route from agent a_i to a_j

Let us consider the *route* from agent a_i to a_j , see Fig 2, Let $Route(a_i, a_j) = (a_i, w_1, w_2, \dots, w_x, a_j)$, there exist x_{ij} *hopes*, where: $x_{ij} \in |W_c|$, number of gateways in the path from a_i to a_j , where is possible install UTM elements. Besides, the cost of

transmitting the brute traffic demand from node a_i to a_j , denoted as $CostTr(a_i, a_j)$, should be:

$$CostTr(a_i, a_j) = BruteTr(a_i, a_j)(1 + x_{ij})C$$

For next paragraphs assumes that all UTMes to be installed are equals in sense of threat detection and analysis of packet flows. So, have the same definition threat files and the same algorithms to detect undesired packet flows, so the analysis of these elements is not necessary in next paragraphs. In addition, time required for doing updates is 0 in all elements. Consequently, the packet discards is desired to be done in the nearest UTM element to packets sources, the next UTMes that should be installed in route path will consider all packets as valid, because first UTM has flow packets forwarded.

This means that traffic between nodes (a_i, a_j) is the traffic flowing on the network immediately after having passed the first UTM that is on the route, see *Definition 1*.

$$\begin{aligned} & \forall_{\substack{i, j \in [1, n] \\ n = |A|}} (a_i, a_j) \in Ax A, \exists (a_i, w_1, \dots, w_m, a_j) \subset Route(a_i, a_j) \\ \Rightarrow & w_p \in W, p \in [1, m], BruteTr(a_i, a_p) \geq BruteTr(w_p, w_{p+1}) \\ \Rightarrow & NetTr(a_i, a_j) = BruteTr(w_p, w_{p+1}) \end{aligned}$$

Proof 1

However, if the UTM is located in the node nearest to the origin of packets (w_1 the first hope in the route from a_i to a_j), then the transmission cost will have the lowest possible value, see *proof 1*, because packets are discarded as quickly as possible and *equation 1* will be the following:

$$CostTr(a_i, a_j) = x_{i,k} BruteTr(a_i, a_k)C + x_{k+1,j} NetTr(a_{k+1}, a_j)C$$

Where: $i < k < j$; and k represent the node were is installed a UTM element

Of equation 1: $CostTr(a_i, a_j) = C * \left(\frac{x_{i,k} NetTr(a_i, a_k)}{1 - H_r(a_i, a_k)} + x_{k+1,j} NetTr(a_{k+1}, a_j) \right)$

$$\begin{aligned}
&= \frac{C^*(x_{i,k} \text{NetTr}(a_i, a_k) + x_{k+1,j} \text{NetTr}(a_{k+1}, a_j) - x_{k+1,j} \text{NetTr}(a_{k+1}, a_j) H_r(a_i, a_k))}{1 - H_r(a_i, a_k)} \\
& \quad x_{i,j} \text{NetTr}(a_i, a_j) = x_{i,k} \text{NetTr}(a_i, a_k) + x_{k+1,j} \text{NetTr}(a_{k+1}, a_j) \\
&= \frac{C^*(x_{i,j} \text{NetTr}(a_i, a_j) - x_{k+1,j} \text{NetTr}(a_{k+1}, a_j) H_r(a_i, a_k))}{1 - H_r(a_i, a_k)}
\end{aligned}$$

The expression $x_{k+1,j} \text{NetTr}(a_{k+1}, a_j) H_r(a_i, a_k)$ of *equation 3* represents the yield benefit from the existence of UTMes in route path from a_i to a_j . If UTM were placed in last hop of path, the yield benefit would have been of only $(\text{NetTr}(a_{k+1}, a_j) H_r(a_i, a_k))$, it is $x_{k+1,j}$ times less than if UTM were placed in the first hope in the path. This illustrates how important it is to locate the UTM near the source of traffic. If the distance between the source and the UTM decrees, the benefits of discard packet are more evident. Note for global network traffic the importance of locate the UTMes in principal routes to optimize flows in the maximum number of route paths; because connections between agents are asymmetric, the importance of calculate in separate ways reduction of flows for $\text{Route}(a_i, a_j)$ and $\text{Route}(a_j, a_i)$.

Installing more than one UTM per route path does not discard more packets, because UTMes have the same characteristics and packets forwarded for first UTM in route path are recognized as valid for next UTMes, as is assumed in *definition 1*. So the benefit is the benefit yield by the UTM nearest to the source, *see proof 2*.

3.1 INSTALLATION OF TWO OR MORE UTM ELEMENTS IN A GIVEN NETWORK

If there are on the $\text{path}(a_i, a_j)$ two or more UTM elements, the benefit is the gain of the performance of the first UTM, that is closest to the source; because the first UTM dismiss all malicious packets. Assuming that all the used UTMes have similar technical characteristics (*algorithms*) and the same detection rate, the second and so on UTMes on a particular $\text{path}(a_i, a_j)$ cannot exclude packages or introduce an additional delay. It is therefore essential to assign UTMes in all the major routes of transmission between different software agents, so that any process of communication between two agents located at different nodes must be on a compulsory cross at least one of the UTM elements in the form closest to the source as possible.

$$\forall_{\substack{i,j \in [1,n] \\ n=|A|}} \exists_{k \in [1,x]} w_k \Rightarrow Route(a_i, a_j) = (a_i, w_1, \dots, w_x, a_j) \wedge \exists_{p \in [1,m]} UTM_p \text{ } m < x \Rightarrow UTM_p = w_k$$

$$Route(a_i, a_j) = (a_i, UTM_1, \dots, UTM_m, a_j) = (R_1, R_2, \dots, R_{m+2})$$

$$\text{If } D = (R_1, R_2, \dots, R_{m+2}) \wedge BruteTr = (BT_1, BT_2, \dots, BT_{m+2})$$

$$RT \in (D \times BruteTr)^{m+2}; RT = \begin{pmatrix} R_1 & R_{m+2} \\ BT_1 & BT_{m+2} \end{pmatrix}$$

$$BT_y = \text{packets arriving node } y, \text{ where } y \in [1, \dots, m+2]$$

$$\text{If } BT_k \neq BruteTr(a_i, a_j), k \in [1, \dots, m+2] \Rightarrow BT_q = BT_{q+1}$$

$$\forall_{q \in (k, k+1, \dots, m+1)} BT_q = NetTr(a_i, a_j)$$

Proof 2

4. SOLUTION

The benefit from installing UTMs elements in a specific route (selected gateway) is the reduction of *vulnerability scanning*, *Web/URL filtering*, *SPAM traffic*, *indistinguishable voice and video that are not generated by agents (critical applications)*, because undecided packets can congest links after each gateway hop. It is undecided flows influence depend on several aspects of the number of nodes that their packets go through it, and UTM's benefits depends of relative position on each of the nodes that packets hops in the route path.

Let $(a_i, a_j) \in A$, and let $w \in Route(a_i, a_j)$. The relative position of any gateway w_{ij} in route path from a_i to a_j is denoted as $RelPos(w_{ij}, a_i, a_j)$. The relative position represent the number of gateways that all packets have to jump in its $path(a_i, a_j)$ before are filtered for a first UTM. Let $LenRoute(a_i, a_j)$ be the numbers of gateways in that route. Then, the benefit obtained from installing UTM in route hop w_{ij} for route $Route(a_i, a_j)$ is:

$$Benefit(w_{ij}, a_i, a_j) = BruteTr(a_i, a_j) H_r(a_i, a_j) (LenRoute(a_i, a_j) - RelPos(w_{ij}, a_i, a_j)) * C$$

Let $I \subseteq W$, where I is a set of gateways where an UTM has been installed. Then the benefit obtained from installing this set of UTMs in the network G will be:

$$Benefit(I, N) = \sum_{a_i, a_j \in A} \left(\max_{\{w \in I \cap Route(a_i, a_j)\}} Benefit(w, a_i, a_j) \right)$$

Problem 1: Optimization of cloud services visibility

The application flow mapping, see Fig. 3., illustrate about dependencies among the distinct elements involved in delivering applications in a cloud.

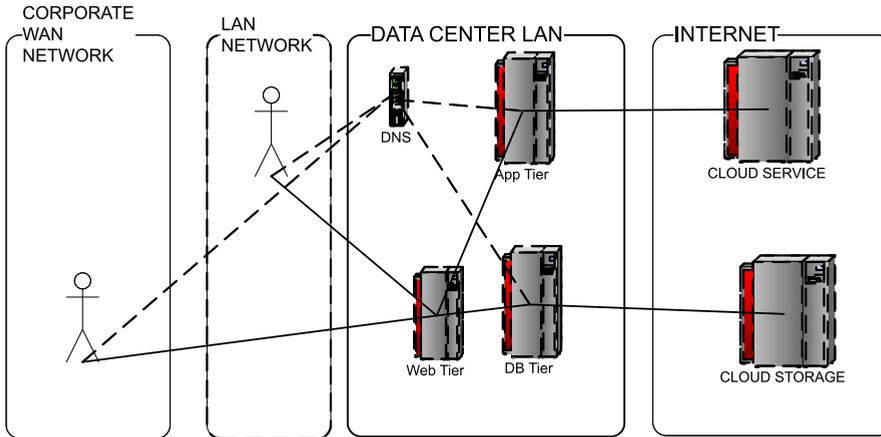


Fig. 3. Application Flow Mapping

Application is composed of agents connecting from local network or corporate WAN to corporate data center or to public infrastructure. Each of these components has subcomponents that are cited in Fig. 3. To guarantee visibility of virtualized application services over public infrastructure is important to know the following network characteristics, which can be measured or estimated for analysis using the uniform probabilistic distribution:

- 1) Identified agents in border nodes interconnecting between themselves by the network, denoted in the form of $A = \{a_1, a_2, a_3, \dots, a_m\}$
- 2) Diagram of map dependencies, see Fig 3, among the elements involved in delivering application as a network graph $G = (V, E)$
- 3) Measure traffic volume along each application delivery path, traffic matrix between nodes $V \times V$, $A = Adj(G)$
- 4) Measure cost of each application delivery path for 1G package per minute.
- 5) Analysis estimated that 25% of network flow is garbage and represent threats for network performance and service visibility.
- 6) Number of devices to enforce security policies (UTM) are 2

TASK TO ANALYZE

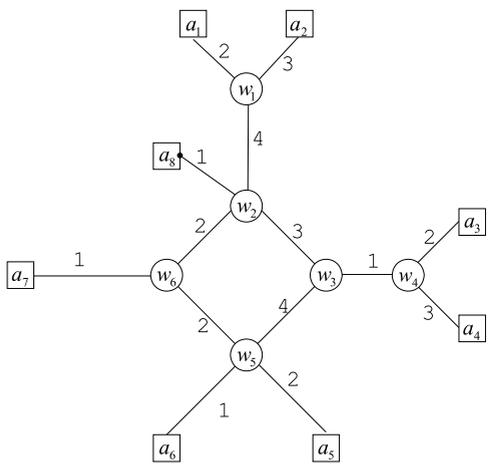
Determine the optimum node in path, between virtualized application and client

agents, to install UTM devices that reduce network vulnerabilities and enforces cloud security. Consequently, UTMs deployments contribute to service visibility.

SOLUTION

Given

- 1) Hit detection rate $H_r = 0,25$
- 2) Number of UTM elements to be allocated $K = \{2\}$
- 3) Traffic volume along each application delivery path (a_i, a_j) , given in Gbps
- 4) Communication agents $A = \{a_1, a_2, a_3, \dots, a_m\}$
- 5) Set of gateways to interconnect application components $W = \{w_1, w_2, w_3, w_4, w_5, w_6\}$
- 6) The network metrics to each link is the cost, value in USD
- 7) Transform application flow mapping, see Fig. 1 in a graph $G = (V, E)$, see Fig 2.



Where:

- a_1 - represent the users of corporate LAN
- a_2 - represent the users of corporate WAN
- a_3 - cloud service
- a_4 - cloud storage
- a_5 - DB Tier
- a_6 - App Tier
- a_7 - Web Tier
- a_8 - DNS

Fig. 4. Graph representation of Application Flow Mapping

Traffic volume matrix shows between nodes $V \times V$ is presented in Table 3, i.e. the result of measures in a network interfaces using the software *Ethereal version 0.99*, besides for scientist analysis could be also generate by *random number generator* based on *uniform probability distribution*.

Table 3. Traffic flow composition

Component	% of transmission flow	% of garbage
Data transfer	30	20
E-mail	20	80
Web URL	40	35
Others	10	20

Dijkstra algorithm provides the main paths that interconnect agents A , applying

equation 5 to analysis of the given network, recognized that principal gateways are $(w_2, w_3, w_4, w_5, w_6)$ because interconnect principal process between software agents.

To find out the optimal location for two UTMs appliances in a given network took advantage of method of reviewing the complete, and the results are presented in Fig 5. For analysis were considered four commodity IT functions, with the flow composition in links showed in Table 3.

Table 3. Traffic matrix between agent nodes VxV

V/V	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈
a ₁	0	0	8	12	0	4	7	1
a ₂	6	0	5	10	0	3	5	1
a ₃	10	8	0	20	12	10	6	1
a ₄	20	15	12	0	15	8	5	2
a ₅	0	0	4	10	0	20	6	1
a ₆	8	6	12	8	10	0	20	2
a ₇	7	6	10	15	0	9	0	2
a ₈	1	1	1	1	1	1	1	1

Simulation, see Fig 5, demonstrate that best result is when UTMs are located in nodes w_2 and w_4 because the threat flows is the least of all possible, this mean that the UTM are cleaning much of threats transmitted and consequently contribute significantly for optimal service visibility.

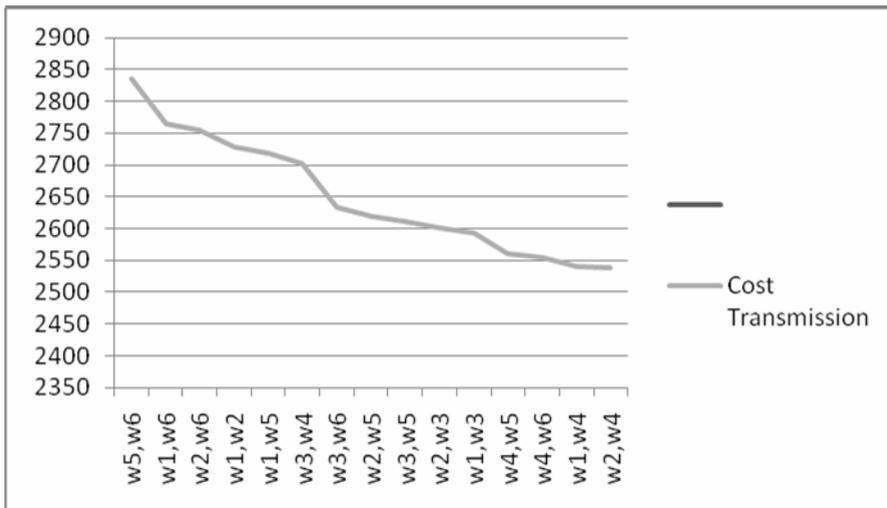


Fig 5. Simulation of cost transmission as function of UTMs location

If UTMs are located in nodes w_2 and w_4 , the optimal transmission cost is: 2538, 62 USD per day, see Fig 6, besides if the cloud is designed without UTM elements the normal cost of transmission is 3223 USD per day.

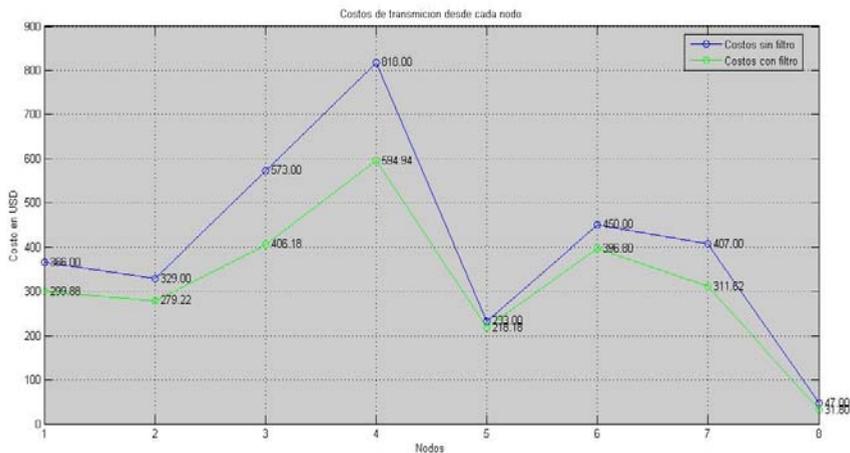


Fig 6. Simulation of transmission cost in nodes for data of Table 3, with UTM installed in nodes $\{w_2, w_4\}$ and without UTM

5. CONCLUSION

Before move applications in a cloud private or public is important a *substantial reduction of vulnerabilities and undesired network flows*. Indeed, a very important element to consider selecting a cloud service provider is the highest security level, still is fundamental that *Infrastructure as a Service IaaS (hardware and networking)* will have efficient proactive defenses.

The correct inventory of all applications and baseline their optimal performance, as well as continuous observation of network flows is fundamental for cloud efficiency, contribute to take the right decisions referents to expand resources on demand and satisfy user expectative. For instance, the best solution to guarantee service visibility of *Software as a Services SaaS* (cloud virtualized applications) is the method for deployment of UTM elements in principal gateway nodes.

Finally, the method presented in this work helps to optimize the quality and quantity of network flows along principal route paths; so has direct influence in cloud virtualized application performance and contributes to *optimal service visibility*.

REFERENCES

- [1] ALLEN, J., CHRISTIE, A., FITHEN, W., MCHUGH, J., PICKEL, J., AND STONER, E., *State of the Practice of Intrusion Detection Technologies*. CMU/SEI-99-TR-028, Carnegie Mellon University, Software Engineering Institute, January 2000.
- [2] ALMOND C, *A Practical Guide to Cloud Computing Security*. Avanade, 27 August 2009
- [3] ANDERSON, JAMES P., *Computer Security Threat Monitoring and Surveillance*. James P. Anderson Co., Fort Washington, Pa., 1980.
- [4] AIDE, *Manual*, <http://www.cs.tut.fi/~rammer/aide/manual.html>.
- [5] BACE, REBECCA GURLEY. *Intrusion Detection*. Indianapolis, IN: Macmillan Technical Publishing, 687-704, 2000.
- [6] CISCO. *Fundamentals of an Internetworks*, 2002.
- [7] GRZECH A, *Sterowanie ruchem w sieciach teleinformatycznych*, Wroclaw 2002.
- [8] GRAHAM CLULEY, *Security Threats: A Look Ahead to 2009*, Sophos Labs 2008.
- [9] KRISHNAN P, DANNY RAZ, YUVAL SHAVITT, *The Cache Location Problem*, IEEE/ACM Transactions On Networking, VOL. 8, NO. 5, 2000, 568–582.
- [10] SWIATEK J, *Wybrane zagadnienia identyfikacji statycznych systemow zlozonych*, Politecnika Wroclawska, Wroclaw 2009

*threshold mechanism, static reservation,
priorities, dynamic reservation,
blocking probability, carried traffic.*

Mariusz GŁĄBOWSKI*, Maciej SOBIERAJ*,
Maciej STASIAK*, Piotr ZWIERZYKOWSKI*

MODELING OF RESOURCE MANAGEMENT MECHANISMS FOR VIRTUAL NETWORKS

The aim of this chapter is to evaluate the effectiveness of resource management mechanisms within the context of virtual networks. The chapter presents resource management mechanisms for full-availability systems, e.g., dynamic reservation, static reservation, threshold mechanisms and priority mechanisms. The effectiveness of the mechanisms under consideration is evaluated on the basis of analytical methods that made it possible to determine the value of the blocking probability and the value of offered traffic in the full-availability systems with BPP traffic and different resource management methods.

1. INTRODUCTION

Currently in the EU many projects concerning virtualization of network resources are being carried out. The goal of the largest Polish project “Future Internet Engineering” (FIE, Polish: IIP) [1] is to construct a prototype of a modern network that would address and eliminate all the shortcomings and flaws of the Internet network that is currently used. The concept of the network used in the project that is under way is based on a multi-level virtualization [2].

Working out effective and efficient methods for managing resources in nodes of communications networks, especially in the case of multi-service networks that rely on virtualization of resources, is a complex issue. One of the fundamental difficulties arises from the necessity of servicing different classes of traffic streams by a network. The next important element influencing the complexity of the resource management

* Chair of Communication and Computer Networks, Poznań University of Technology, pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland.

mechanisms in virtual networks is the fact that resources can be executed both with the utilization of a single physical resource and with the utilization of many physical resources. The initial analysis of the problem related to designing feasible strategies of resource management in multi-service network indicates that the ones of the most effective strategies can be reservation mechanisms of resources, both dynamic (executed on-line and securing well-balanced access to network resources) [11,12], and static (executed appropriately ahead of time with time advancement) [11,12], as well as threshold mechanisms [12,13] and priority mechanisms [10].

The mechanisms adopted by telecommunications operators (e.g., reservation and threshold mechanisms) require performing appropriate traffic analysis of the operating network systems and their optimal dimensioning. Until recently, the main emphasis has been put on the blocking probability calculation in systems without virtualization. Nowadays, in order to fully determine the influence of resource management mechanisms on the effectiveness of telecommunications networks as well as in order to determine the optimal share of virtual operators in physical resources, it is necessary to work out analytical methods that would enable us to model traffic characteristics of networks (especially the value of carried traffic) on the basis of virtualization.

This chapter presents resource management mechanisms that can be used in links of the networks based on virtualization of resources and servicing traffic streams generated by BPP (Binomial-Poisson-Pascal) calls streams. For this purpose, static and dynamic bandwidth reservation algorithms are elaborated. Additionally, threshold mechanism and priority mechanism are also considered. The effectiveness of the presented mechanisms has been evaluated using analytical methods [3,4].

The remaining part of the chapter is organized as follows. Section 2 presents the analytical models of the full-availability systems with BPP traffic and different strategies of resource management. In Section 3, the results of blocking probability and carried traffic in exemplary full-availability systems are presented and compared for different resource management mechanisms. Section 4 concludes the chapter.

2. RESOURCE MANAGEMENT MECHANISMS

2.1. STRUCTURE OF OFFERED TRAFFIC

Let us assume that to the full-availability system (a full-availability link in real or virtual networks) traffic streams of the three following types are offered: m_I Erlang traffic streams (Poisson call streams) from the set $I = \{1, \dots, i, \dots, m_I\}$, m_J Engset traffic streams (binomial call streams) from the set $J = \{1, \dots, i, \dots, m_J\}$ and m_K Pascal traffic streams (negative binomial call streams) from the set $K = \{1, \dots, i, \dots, m_K\}$. The total

number of traffic classes is then equal to: $M = m_i + m_j + m_k$. It has been adopted in this chapter that the letter „ i ” denotes any class of Erlang traffic, the letter „ j ” any class of Engset traffic, and the letter „ k ” any class of Pascal traffic, whereas the letter „ c ” – any traffic class¹. The number of the so-called basic bandwidth units (BBUs) demanded by calls of class c is denoted by the symbol t_c (BBU can be defined as the greatest common divisor of the required call resources of all service classes; BBU can be determined for both real networks and for networks with virtualization capabilities). The mean traffic offered by class i Erlang stream can be expressed by the following formula:

$$A_i = \lambda_i / \mu_i, \tag{1}$$

where λ_i is the call intensity (arrival rate) for Erlang traffic of class i , and μ_i^{-1} is the mean holding (service) time of class i calls. In the case of Engset and Pascal streams, the mean traffic $A_j(n)$ offered by class j Engset stream and the mean traffic $A_k(n)$ offered by class k Pascal stream depend on the state of the system (i.e. the number of n busy BBUs):

$$A_j(n) = N_j \alpha_j \sigma_{j,T}(n), \tag{2}$$

$$A_k(n) = N_k \beta_k \sigma_{k,T}(n), \tag{3}$$

where the transition coefficients $\sigma_{j,T}(n)$ for class j Engset stream and $\sigma_{k,T}(n)$ for class k Pascal stream are defined as follows:

$$\sigma_{j,T}(n) = (N_j - y_j(n)) / N_j, \tag{4}$$

$$\sigma_{k,T}(n) = (N_k + y_k(n)) / N_k, \tag{5}$$

In Equations (2)–(5) we have adopted the following notation: N_j – the number of class j Engset sources, α_j – the mean traffic offered by a single Engset source of class j , $y_j(n)$ – the mean number of class j Engset calls serviced in state n , N_k – the number of class k Pascal sources, β_k – the mean traffic offered by a single Pascal

¹ In traffic theory the term BPP (Binomial–Poisson–Pascal) is used to describe this group of call/traffic streams.

source of class k , $y_k(n)$ – the mean number of class k Pascal calls serviced in state n . The intensity of Engset traffic α_j and Pascal traffic β_k of class j and k , respectively, offered by one free source is equal to:

$$\alpha_j = \gamma_j / \mu_j, \quad (6)$$

$$\beta_k = \gamma_k / \mu_k, \quad (7)$$

where γ_j and γ_k are the arrival rates of calls generated by a single free source of class j and k , respectively.

Further on in the chapter we look more closely at the resource management mechanisms – for the system under investigation – and at the analytical methods that make it possible to determine their influence on traffic characteristics.

2.2. DYNAMIC RESERVATION MECHANISM

Let us consider the full-availability group with dynamic reservation [12]. According to the adopted reservation mechanism in the system under consideration in this scenario, a quantity Q_c called the reservation threshold is introduced to a given class c . The system admits a given call of class c for service only when the number of free BBUs in the group is higher or equal to the value of the reservation space $R_c = V - Q_c$. Note that, in the case of the model of the system with reservation under consideration, the operation of the reservation mechanism introduces a dependence between the service stream in the system and the current state of the system. This dependence will be determined by the parameter $\sigma_{c,R}(n)$:

$$\sigma_{c,R}(n) = \begin{cases} 1 & \text{for } n \leq Q_c, \\ 0 & \text{for } n > Q_c. \end{cases} \quad (8)$$

Taking the above dependence into consideration, we are in position to determine the occupancy distribution in the full-availability group with reservation on the basis of the following equation:

$$\begin{aligned}
 n[P_n]_V &= \sum_{i=1}^{m_i} A_i \sigma_{i,R} (n-t_i) t_i [P_{n-t_i}]_V + \\
 &+ \sum_{j=1}^{m_j} N_j \alpha_j \sigma_{j,T} (n-t_j) \sigma_{j,R} (n-t_j) t_j [P_{n-t_j}]_V + \\
 &+ \sum_{k=1}^{m_k} S_k \beta_k \sigma_{k,T} (n-t_k) \sigma_{k,R} (n-t_k) t_k [P_{n-t_k}]_V.
 \end{aligned} \tag{9}$$

Having the occupancy distribution, calculated on the basis of Formula (9) at hand, we can determine the value of the parameters $y_j(n)$ and $y_k(n)$ that are necessary to determine the transition coefficients $\sigma_{j,T}(n)$, determined by Equation (4), and $\sigma_{k,T}(n)$ determined by Equation (5) [3, 4]:

$$y_j(n) = \begin{cases} N_j \alpha_j \sigma_{j,T} (n-t_j) \sigma_{j,R} (n-t_j) [P_{n-t_j}]_V / [P_n]_V & \text{for } n \leq V, \\ 0 & \text{for } n > V, \end{cases} \tag{10}$$

$$y_k(n) = \begin{cases} S_k \beta_k \sigma_{k,T} (n-t_k) \sigma_{k,R} (n-t_k) [P_{n-t_k}]_V / [P_n]_V & \text{for } n \leq V, \\ 0 & \text{for } n > V. \end{cases} \tag{11}$$

The blocking probability E_c for calls of individual traffic classes is expressed by the following equation:

$$E_c = \begin{cases} \sum_{n=V-Q_c+1}^V [P_n]_V & \text{for } Q_c < V-t_c \\ \sum_{n=V-t_c+1}^V [P_n]_V & \text{for } Q_c \geq V-t_c \end{cases} \tag{12}$$

whereas the value of carried traffic is determined by the formula:

$$L = \sum_{c=1}^M \sum_{n=0}^V y_c(n) t_c [P_n]_V. \tag{13}$$

2.3. STATIC RESERVATION MECHANISM

Let us consider the full-availability system with static reservation [11,12]. Static reservation consists in dividing the resources of a group between calls of different traffic classes. The full-availability group with the capacity of V BBUs has been di-

vided into two groups with the capacities V_1 and V_2 , where $V = V_1 + V_2$. Group V services $M = m_l + m_j + m_k$ call streams. Group V_1 services $M_1 = k_l + k_j + k_k$ call streams, and Group V_2 the remaining $M_2 = (m_l - k_l) + (m_j - k_j) + (m_k - k_k)$. traffic streams. Capacities for the groups V_1 and V_2 are, in the main, selected in such a way as to make the service quality parameters in both groups, e.g. the blocking probability, similar. Calculations of the blocking probability for calls of particular classes can be reduced to the two models of the full-availability group. Occupancy distributions in the first and the second group can be presented by the following equations:

$$n[P_n]_{V_1} = \sum_{i=1}^{k_l} A_i t_i [P_{n-t_i}]_{V_1} + \sum_{j=1}^{k_j} N_j \alpha_j \sigma_{j,T} (n-t_j) t_j [P_{n-t_j}]_{V_1} + \sum_{k=1}^{k_k} S_k \beta_k \sigma_{k,T} (n-t_k) t_k [P_{n-t_k}]_{V_1}. \quad (14)$$

$$n[P_n]_{V_2} = \sum_{i=k_l+1}^{m_l} A_i t_i [P_{n-t_i}]_{V_2} + \sum_{j=k_j+1}^{m_j} N_j \alpha_j \sigma_{j,T} (n-t_j) t_j [P_{n-t_j}]_{V_2} + \sum_{k=k_k+1}^{m_k} S_k \beta_k \sigma_{k,T} (n-t_k) t_k [P_{n-t_k}]_{V_2}. \quad (15)$$

After a determination of the occupancy distributions in Groups V_1 and V_2 , it is possible to determine the blocking probability for calls of class c :

$$E_c = \sum_{n=V-t_c+1}^V [P_n]_V \quad (16)$$

and – based on Formula (13) – determine the value of carried traffic.

2.4. THRESHOLD MECHANISM

Let us consider a full-availability group with threshold mechanism [12,13]. In the considered system, for each class of calls a set of threshold is individually introduced. For class c , a set of established thresholds can be written as follows: $\{Q_{c,1}, Q_{c,2}, \dots, Q_{c,q_c}\}$, where it is also assumed that: $\{Q_{c,1} \leq Q_{c,2} \leq \dots \leq Q_{c,q_c}\}$. In the system, a given threshold area u of class c limited by thresholds $Q_{c,u}$ and $Q_{c,u+1}$ is defined by its own set of parameters $\{\lambda_c, t_{c,u}, \dots, \mu_{c,u}\}$, where $t_{c,0} > t_{c,1} > \dots > t_{c,u} > \dots > t_{c,q_c}$ and $\mu_{c,0}^{-1} \leq \mu_{c,1}^{-1} \leq \dots \leq \mu_{c,u}^{-1} \leq \dots \leq \mu_{c,q_c}^{-1}$. In threshold systems, with the increase in the load of the system, the number of BBUs assigned to service

calls of particular classes decreases and, at the same time, the average holding time of the calls can be extended. In pre-threshold area, class c calls require $t_{c,0}$ BBUs to set up a new connection and the mean holding time equals $\mu_{c,0}^{-1}$. When the load of the system increases above threshold $Q_{c,1}$, the system will be in the threshold area 1. The number of required BBUs decreases from $t_{c,0}$ to $t_{c,1}$, and the mean holding time increases to $\mu_{c,1}^{-1}$. The situation looks similar when the load of the system exceeds next thresholds.

The traffic intensity $A_{i,u}$ of the class i Erlang stream offered to the group in area $Q_{c,u} < n < Q_{c,u+1}$, equals:

$$A_{i,u} = \lambda_i / \mu_{i,u}. \quad (17)$$

In the case of Engset sources, arrival rate of particular traffic classes decreases with the occupancy state of the system, while in the Pascal case arrival rate of particular traffic classes increases with the occupancy state of the system. Consequently, the value $A_{j,u}(n)$ of traffic offered by Engset sources in area $Q_{c,u} < n < Q_{c,u+1}$ and the value $A_{k,u}(n)$ of traffic offered by Pascal sources in area $Q_{c,u} < n < Q_{c,u+1}$, in state of n BBUs being busy, depends on the number of calls being serviced. To help calculate traffic intensities $A_{j,u}(n)$ and $A_{k,u}(n)$ we introduced the passing (transition) coefficients $\sigma_{j,u,T}(n)$ and $\sigma_{k,u,T}(n)$:

$$A_{j,u}(n) = N_j \alpha_{j,u} \sigma_{j,u,T}(n), \quad (18)$$

$$A_{k,u}(n) = N_k \beta_{k,u} \sigma_{k,u,T}(n), \quad (19)$$

$$\sigma_{j,u,T}(n) = (N_j - y_{j,u}(n)) / N_j, \quad (20)$$

$$\sigma_{k,u,T}(n) = (N_k + y_{k,u}(n)) / N_k, \quad (21)$$

The threshold coefficient of passing (transition coefficient) $\sigma_{c,u,Q}(n)$ determines occupancy states of the system in which offered traffic is defined by the parameters $\{\lambda_c, t_{c,u}, \dots, \mu_{c,u}\}$:

$$\sigma_{c,u,Q}(n) = \begin{cases} 1 & \text{for } Q_{c,u} < n \leq Q_{c,u+1} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

One of the basic methods for modeling the considered system is a method based on the approximation of service processes occurring in the system by reversible processes. As the consequence of the above, it is possible to analyze the process on the basis of the local balance equations, which, in turn, makes it possible to approximate the occupancy distribution in the system on the basis of the generalized Kaufman-Roberts recursion [5, 6]:

$$\begin{aligned}
 n[P_n]_V &= \sum_{i=1}^{m_i} \sum_{u=1}^{q_i} A_{i,u} \sigma_{i,u,Q}(n-t_{i,u}) t_{i,u} [P_{n-t_{i,u}}]_V + \\
 &+ \sum_{j=1}^{m_j} \sum_{u=1}^{q_j} N_j \alpha_{j,u} \sigma_{j,u,T}(n-t_{j,u}) \sigma_{j,u,Q}(n-t_{j,u}) t_{j,u} [P_{n-t_{j,u}}]_V + \\
 &+ \sum_{k=1}^{m_k} \sum_{u=1}^{q_k} S_k \beta_{k,u} \sigma_{k,u,T}(n-t_{k,u}) \sigma_{k,u,Q}(n-t_{k,u}) t_{k,u} [P_{n-t_{k,u}}]_V.
 \end{aligned} \tag{23}$$

Having the occupancy distribution, calculated on the basis of Formula (23), we can determine the value of the parameters $y_{j,u}(n)$ and $y_{k,u}(n)$, applying the approach based on the local balance equations [3], on the basis of the following equations:

$$y_{j,u}(n) = \begin{cases} N_j \alpha_{j,u} \sigma_{j,u,T}(n-t_{j,u}) \sigma_{j,u,Q}(n-t_{j,u}) [P_{n-t_{j,u}}]_V / [P_n]_V & \text{for } n \leq V, \\ 0 & \text{for } n > V, \end{cases} \tag{24}$$

$$y_{k,u}(n) = \begin{cases} S_k \beta_{k,u} \sigma_{k,u,T}(n-t_{k,u}) \sigma_{k,u,Q}(n-t_{k,u}) [P_{n-t_{k,u}}]_V / [P_n]_V & \text{for } n \leq V, \\ 0 & \text{for } n > V. \end{cases} \tag{25}$$

Let us determine now the blocking probability E_c of class c call stream in the system. Usually, the values of the threshold are set, so that each state of the threshold area (except the last post-threshold area q_c) can service an appropriate call [7, 8, 9]. With such assumptions, blocking states in the considered system will occur only in the last threshold area and the blocking probability can be written in the following form:

$$E_c = \sum_{n=V-t_{c,q_c}+1}^V [P_n]_V. \tag{26}$$

The value of carried traffic in the system will be determined on the basis of the following equation:

$$L = \sum_{c=1}^M \sum_{n=0}^V \sum_{u=0}^{q_c} y_{c,u}(n) t_{c,u} [P_n]_V. \quad (27)$$

2.5. PRIORITIES MECHANISM

Let us consider a model of the full-availability group with priorities. In this model we assume that the arrival of a new call with a higher priority can, in the case of the lack of free resources, terminate the currently serviced connections with a lower priority. Additionally, we assume in the model that: all classes offered to the system are ranked by priorities; each of the classes is characterized by a different priority; first class of service is characterized by the highest priority whereas class M ($M = m_l + m_j + m_k$) has the lowest priority; the system services three classes and the first class is characterized by the highest priority, whereas the last class M by the lowest priority; traffic from a lower priority classes does not have any influence on the blocking probability of higher priority classes.

Let us discuss the full-availability group with priorities carrying M traffic classes. The system can be then presented as an $c-1$ step calculation algorithm; in each step, the system with two priorities is considered. In the first step, the calls of the lowest priority c are considered, under the assumption that the remaining classes of calls $(1, \dots, c-1)$ have a higher priority and push out the calls of class c . In the successive steps, traffic of class $c-1$ has the lowest priority, while the remaining traffic classes $(1, \dots, c-2)$ have a higher priority and can push out traffic of class $c-1$.

Hence, to facilitate similar considerations as with the case of the system with two priorities, we can determine the blocking probability for calls of class c with $M-c+1$ priority on the basis of the following formula [10]:

$$E_c^P = \frac{A_c t_c E_c + \sum_{i=1}^{h-1} A_i t_i (E_c - E_{c-1})}{A_c t_c}, \quad (28)$$

where E_c^P is the blocking probability of class c in the system with priorities that services M traffic classes, and E_c is the blocking probability of class c in the system without priorities that also services M traffic classes.

The value of carried traffic of class c in the system will be determined on the basis of the following dependence:

$$L_c = (1 - E_c^P) A_c t_c. \quad (29)$$

3. NUMERICAL RESULTS

In order to determine the effectiveness of the proposed algorithms, the authors investigated their influence on the value of the blocking probability in individual traffic classes and on the value of carried traffic. The research study was carried out for the two systems described below.

System 1 with capacity of $V = 100$ BBUs:

- Structure of offered traffic - call classes: $M = 3$, $t_1 = 1$ BBU, $\mu_1 = 1$ (Erlang), $t_2 = 2$ BBUs, $\mu_2 = 1$, $N_2 = 250$ (Engset), $t_3 = 4$ BBUs, $\mu_3 = 1$, $S_3 = 250$, (Pascal); Dynamic reservation mechanism: $Q_1 = Q_2 = 25$ or $Q_1 = Q_2 = 75$; Static reservation mechanism: $V_1 = 25$, $V_2 = 75$ or $V_1 = 75$, $V_2 = 25$; Threshold mechanism: $t_{1,0} = 1$ BBU, $\mu_{1,0} = 1$, $t_{2,0} = 2$ BBUs, $\mu_{2,0} = 1$, $t_{3,0} = 4$ BBUs, $\mu_{3,0} = 1$, $t_{3,1} = 2$ BBUs, $\mu_{3,1} = 2$; $Q_3 = 25$ or $Q_3 = 75$; Priorities mechanism.

System 2 with capacity of $V = 100$ BBUs:

- Structure of offered traffic - call classes: $M = 3$, $t_1 = 1$ BBU, $\mu_1 = 1$ (Erlang), $t_2 = 4$ BBUs, $\mu_2 = 1$, $N_2 = 250$ (Engset), $t_3 = 16$ BBUs, $\mu_3 = 1$, $S_3 = 250$, (Pascal); Dynamic reservation mechanism: $Q_1 = Q_2 = 25$ or $Q_1 = Q_2 = 75$; Static reservation mechanism: $V_1 = 25$, $V_2 = 75$ or $V_1 = 75$, $V_2 = 25$; Threshold mechanism: $t_{1,0} = 1$ BBU, $\mu_{1,0} = 1$, $t_{2,0} = 4$ BBUs, $\mu_{2,0} = 1$, $t_{3,0} = 16$ BBUs, $\mu_{3,0} = 1$, $t_{3,1} = 8$ BBUs, $\mu_{3,1} = 2$; $Q_3 = 25$ or $Q_3 = 75$; Priorities mechanism

The following graphs 1-2 present the percentage change in the value of carried traffic in the systems under consideration. It is noticeable that different resources management mechanisms have a different influence on the volume of carried traffic. From the perspective of the volume of carried traffic, the best mechanism is the threshold mechanism. When an increase in the reservation threshold from 25 BBUs to 75 BBUs ensues, it is observable that the differences in the volume of carried traffic in systems with different mechanisms tend to decrease. Similar results are also obtained with increasing differences in demands of individual traffic classes.

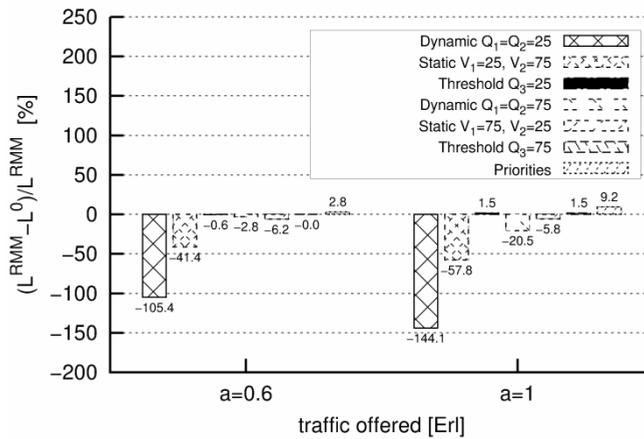


Fig. 1. Percentage change in the value of carried traffic in the system 1 with and without resource management mechanisms

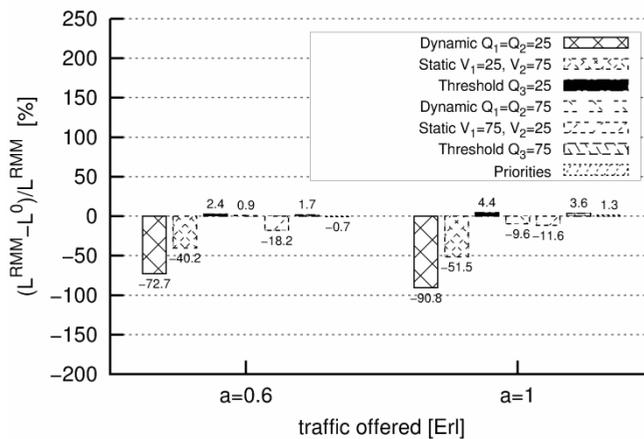


Fig. 2. Percentage change in the value of carried traffic in the system 2 with and without resource management mechanisms

By analysing the graphs 3-5, it is possible to determine the dependencies that occur in the system and that relate to the blocking probability for individual traffic classes in System 2. A decrease in the blocking probability for class 1 is observable with the case of the application of the threshold mechanism, static reservation and priorities in the system with the load of $a = 0.6$. In other instances, an increase in the blocking probability for calls of class 1 ensues. Identical response of the system is also observable with the case of calls of class 2, except for the priorities for the load $a = 1$. For calls of class 3, a decrease in the blocking probability is to be observed in almost

all cases. The highest decrease in the blocking probability creates favourable conditions for the application of the dynamic reservation mechanism.

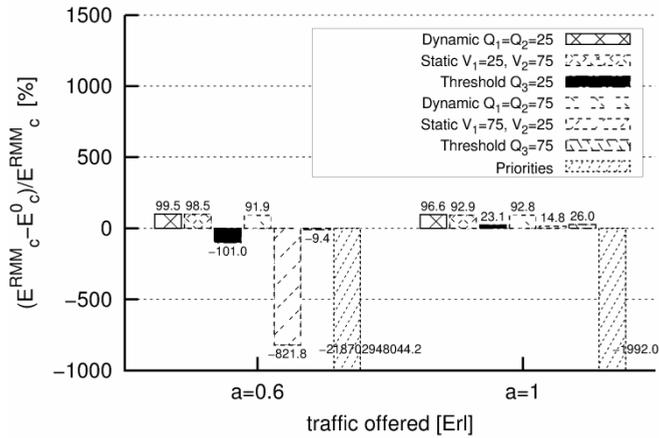


Fig. 3. Percentage change in the value of blocking probability for class 1 calls in the system 2 with and without resource management mechanisms

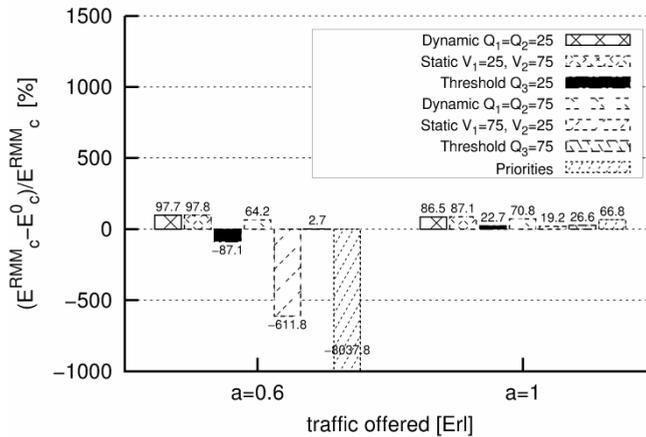


Fig. 4. Percentage change in the value of blocking probability for class 2 calls in the system 2 with and without resource management mechanisms

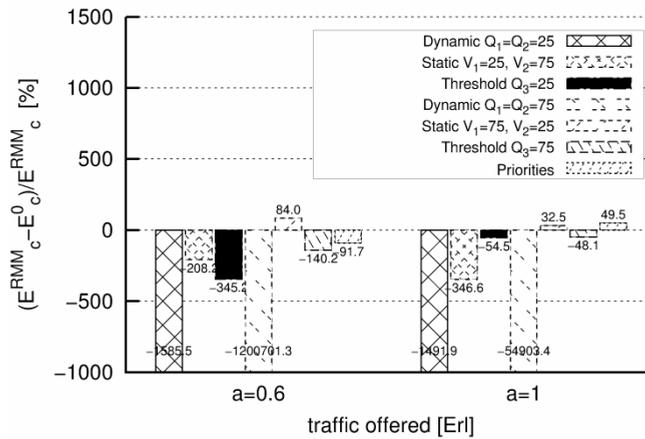


Fig. 5. Percentage change in the value of blocking probability for class 3 calls in the system 2 with and without resource management mechanisms

4. CONCLUSIONS

The chapter presents the results of analytical studies on the influence of resource management mechanisms on such traffic characteristics as the value of carried traffic and the blocking probability. The study was carried out for full-availability systems that serviced multiservice BPP traffic streams. The study proves and demonstrates that only the threshold mechanism and priorities have active influence on the increase in the value of carried traffic. The changes in the value of blocking probabilities observed in other systems are related to a simultaneous decline in the use of the resources of links, i.e. a decrease in the value of carried traffic in the system. Due to the limited space of the chapter, only part of the results is presented here. The results obtained in the course of the study of other systems corroborate the researchers' conclusions.

REFERENCES

- [1] Projekt Inżynieria Internetu Przyszłości – <https://www.iip.net.pl/project>.
- [2] Network Virtualization – <http://www.arl.wustl.edu/netv/main.html>.
- [3] GŁĄBOWSKI M., KALISZAN A., STASIAK M., *Modeling product-form state-dependent systems with BPP traffic*, Journal of Performance Evaluation, vol. 2010, no. 67, pp. 174–197, 2010.
- [4] GŁĄBOWSKI M., KALISZAN A., STASIAK M., *Iterative Algorithm for Blocking Probability Calculation in Erlang-Engset-Pascal Multi-rate Systems*, Theoretical and Applied Informatics, Vol. 19, No. 4, 2007, pp. 281–295.
- [5] KAUFMAN J., *Blocking in a shared resource environment*, IEEE Transactions on Communications, vol. 29, no. 10, pp. 1474–1481, 1981.

- [6] ROBERTS J., *A service system with heterogeneous user requirements-application to multi-service telecommunications systems*, in Proceedings of Performance of Data Communications Systems and their Applications, G. Pujolle, Ed. Amsterdam: North Holland, 1981, pp. 423–431.
- [7] KAUFMAN J., *Blocking with retrials in a completely shared resource environment*, Journal of Performance Evaluation, 15:99–113, 1992.
- [8] MOSCHOLIOS I., LOGOTHETIS M., NIKOLAROPOULOS P., *Call blocking probabilities in a multirate loss model of quasi-random input*, In Proceedings of First International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks HET-NETs'03, pages 6/1–6/10, Ilkley, July 2003.
- [9] MOSCHOLIOS I., LOGOTHETIS M., KOUKIAS M., NIKOLAROPOULOS P., *Call burst blocking probabilities in an ON-OFF multi-rate loss model of quasirandom input*. in Proceedings of First International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks HETNETs'03, pages 21–23, Ilkley, July 2003.
- [10] STASIAK M., WIEWIÓRA J., ZWIERZYKOWSKI P.: *The analytical model of the WCDMA interface with priorities in the UMTS network*, International Symposium on Information Theory and its Applications (ISITA2008), Auckland, New Zealand, December, 2008
- [11] STASIAK M.: *Efektywna dostępność w zagadnieniach modelowania pól komutacyjnych*. Wydawnictwo Politechniki Poznańskiej, 2005.
- [12] STASIAK M., GŁĄBOWSKI M., HANCZEWSKI S. ZWIERZYKOWSKI P.: *Podstawy inżynierii ruchu i wymiarowania sieci teleinformatycznych*. Wydawnictwo Politechniki Poznańskiej, 2009.
- [13] GŁĄBOWSKI M.: *Continuous threshold model for multi-service wireless systems with PCT1 and PCT2 traffic*, in Proceedings of 7th International Symposium on Communications and Information Technologies, Sydney, Oct. 2007, pp. 427–432.

Jan KWIATKOWSKI*, Grzegorz PAPKALA*

SERVICE AWARE VIRTUALIZATION MANAGEMENT SYSTEM

Efficient use of available computing resources is currently one of the major challenges facing designers of systems based on the SOA paradigm. Request Distribution Manager is a tool for allocating computing resources to implement the service. To indicate the computing resources, knowledge about the allocated communication resources and the current loading of computing resources are used. The tool allows to apply to the service of various hardware resources, dynamically matched to satisfy the requirements of the service. The decision on the allocation of computing resources is then confronted with the utilization of resources allocated using data from the monitoring of the service execution. This allows to gather knowledge about the quality of the methods used for allocating resources and the need to modify them.

1. INTRODUCTION

In recent years the evolution of software architectures led to the rising prominence of the Service Oriented Architecture (SOA) concept. This architecture paradigm facilitates building flexible service systems. The services can be deployed in distributed environments, executed on different hardware and software platforms, reused and composed into complex services. Adopting the concept of services SOA takes IT to another level, one that's more suited for interoperability and heterogeneous environments. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services".

On the other hand the Service Oriented Architecture and virtualization come closer to each other, then the need to combine them in an efficient way becomes one of the key challenges for designers of systems based on SOA paradigm. Protocols and languages describing the base framework for services have been already well described,

* Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

standardise and established. Similarly virtualization which is often pointed as an SOA enabling technology has been quite matured. The number of offered solutions is not small and as reported in [6] it is hard to find one which outstands all the others. Not only the performance of the hypervisors but also the management tools associated with them are becoming better and better. Nonetheless, management of the virtualized environment with hundreds of thousands of virtual machines is becoming the crucial issue especially in terms of capacity and security [1].

In the chapter an architecture of Request Distribution Manager (RDM) a tool for efficient allocating computing resources to services is proposed. The service requests are examined in accordance to the SOA request description model. To ensure efficient resource utilization, incoming SOA requests are attributed to execution classes. The functional and non-functional request requirements in conjunction with monitoring of execution and communication links performance data are used to distribute requests and allocate the services and resources. RDM exploits the combining the service orientation with automatic management using constraints attached to the requests what increases overall reliability, response time and constraints fulfilment, reducing the need for manual work in the same time. RDM itself is built using independent modules communicating with each other using XML-RPC protocol. This means that they can be easily exchanged as long as the interfaces of the modules are compatible what makes this relatively flexible solutions. On the other hand this yields the set of problems connected with debugging or configuration.

The chapter is organized as follow. Section 2 briefly presents similar tools as RDM and pointed the main differences between them and RDM. Detailed description of the RDM is presented in section 3. It covers general structure of the RDM with short presentation of each module and its role. In section 4 focus is put at the creation of test application with strong capabilities of checking the system behaviour under specific conditions. The test application does not perform real services, it is only designed to simulate the resources usage. The possibility to evaluate RDM performance under virtually any conditions (including features such as CPU or memory usage, service duration, etc.) is, as for now, limited to CPU usage. Finally, section 6 outlines the work and discusses the further works.

2. AN OVERVIEW OF RELATED SOLUTIONS

Overall number of offerings devoted to manage the cloud is increasing constantly. Apart from commercial software offered by the virtualization solutions vendors such as Microsoft's System Center Virtual Machine Manager 2008 R2 or VMware's vSphere there is a number of open solutions for virtualization (cloud) management

which will be described shortly here. These are Nimbus, OpenNebula, OpenStack and Eucalyptus.

Open solutions are in most cases hypervisor agnostic and offer wide range of features to ease the deployment and management of private, public or hybrid clouds. Their architecture is modular which gives high customization opportunities. According to [3] each of those open solutions has slightly different target thus, provides different capabilities. That makes them more suitable either for private, public or scientific purposes.

Nimbus project has a very clear area of interest which is the scientific community [9]. It concentrates on capacity allocation and capacity overflow [3]. The project places a strong emphasis on the research and future cloud technology development. Different approach has been undertaken by the Eucalyptus project which is a reverse engineering of Amazon's offering. It uses the same EC2 (for computing) and S3 (for storage) APIs. Amazon as the leader on the market created a de facto standard API which is supported by all open projects mentioned in this chapter. The compatibility with Amazon's API is claimed as the reason to use Eucalyptus for hybrid clouds with local part and Amazon for needs going beyond own resources [8].

OpenNebula with centralized management is the solution for private clouds [3]. The project was started in 2005 and tries to deliver an "efficient and scalable management of virtual machines on large-scale distributed infrastructures" [10]. Last but not least is the OpenStack software which is the outcome of joint effort of NASA and RackSpace [11] with number of other parties involved in the cloud development. Currently OpenStack is often pointed as the best open solution to create public cloud [5]. The project is divided into three main parts: "Compute", "Storage" and "Image service". It seems to have the widest range of virtualization and image types support with Hyper-V, KVM, LXC, QEMU, UML, VMWare and Xen hypervisors and Raw, AMI, VHD, VDI, qcow2, VMDK and OVF image formats [12] available.

2.1 THE DIFFERENCES BETWEEN RDM AND OTHER APPROACHES

The largest difference between RDM and aforementioned solutions is coming from another targets standing behind the projects. While most of other solutions are strictly devoted to manage the infrastructure, RDM is devoted to properly dispatch the requests placing the virtualization management on the second place. Nonetheless one can point a number of similarities starting from common modular architecture with possibilities to customize the software easily. Furthermore just like other solutions *libvirt* is used to overcome the problem with communication with various hypervisors.

The role of RDM as a dispatcher means that some of the functionalities are redundant. Under such situation one may put offering Amazon compatible API, billing integration, number of control panels and so on. On the other hand the functionality is

extended to understand the SOAP messages, identify which services are capable of performing them and finally running those services and dispatching the requests.

Analogical software is found as an addition on top of e.g. OpenNebula and offers service orchestration and deployment (SVMSched) or service management as a whole (Claudia).

3. REQUEST DISTRIBUTION MANAGER

RDM is built as a component of a SOA. The architecture of RDM resolves the problem with traditional software lack of flexibility. Software composition and distribution in the traditional form, where applications are not open enough to follow rapidly changing needs of business, had to be replaced with something more flexible. The idea to compose the processes from services publicly or privately available, mix and match them as needed, easily connect to business partners, seems like the best way to solve it.

Nonetheless, although virtualization is already being used as a common and proven way to decrease the overall hardware needs and costs, still the hardware utilization is around 15-20% and storage utilization does not go above 60% [2]. Using virtualization gives very promising results, but as stated in [4] it is still not enough. Virtualization stopped at “low-hanging-fruit” and is not pushing forward. Mission critical services are used as before due to the easier maintenance, controlling and monitoring. What is more, reduced budgets and “do more with less” attitude made it much more complicated for real virtualization adaptation since - especially at the beginning - costs of implementation are higher than those of keeping everything as is.

RDM offers two interfaces to interact with the virtualized environment. One is XML-RPC based. More important is the possibility to direct SOAP calls to services to be handled by the RDM. Each and every request is then redirected to proper instance based on the requirements it has. Proper instance is either found from the working and available ones or the new one is started to serve the request.

Such an approach gives the possibility to manage the virtualization automatically with minimal manual interaction. The architecture of the RDM is presented in figure 1, as for now there is a number of independent modules offering the XML-RPC interfaces to interact with them.

- RDM-Manager – manages all other modules and routes the requests to the services (capsules),
- RDM-Virtualization – offers the access to hypervisor actions, uses *libvirt* to execute commands what gives the project independence from particular hypervisor,

- RDM-Database – module used to store all data required while request processing,
- RDM-Monitoring – offer information about particular physical servers as well as virtual instances running,
- RDM-Translate – handles SOAP requests extracting requirements and forwarding the request to proper services based on RDM-Manager answer,
- RDM-Matchmaker – module responsible to properly match the requirements of the request with capabilities of the environment and current state of it.

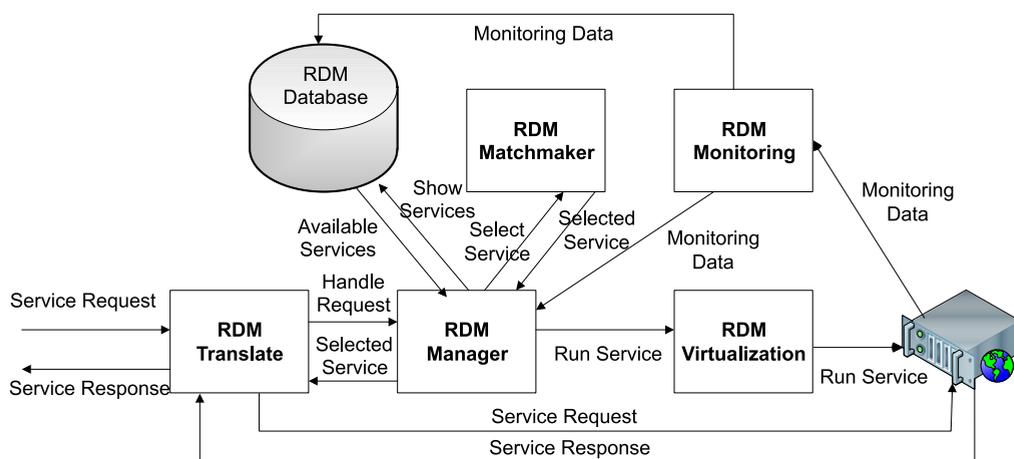


Fig. 1. The architecture of the RDM

3.1. REQUEST DISTRIBUTION

Each SOAP request coming to the system is directed to the RDM-Transform module which extracts requirements passed in the header section of the message to properly handle the request. Whole process of request handling is presented in figure 2. For the sake of simplicity and aim behind the RDM-Transform's interface is the only one available from outside.

Process of request handling starts with SOAP message coming from outside. In the figure 2 broker is taken as a sample outside actor initiating the process. This is in accordance to the general idea of placing RDM inside Service Oriented Architecture where service broker is common module for such purpose. RDM-Transform module has a role of being the gateway to the system and hiding all of the heavy lifting from outside world. It extracts the requirements passed in the SOAP message, in its header section (figure 3). The requirements are extracted and converted to simple text form which is used by RDM-Matchmaker module.

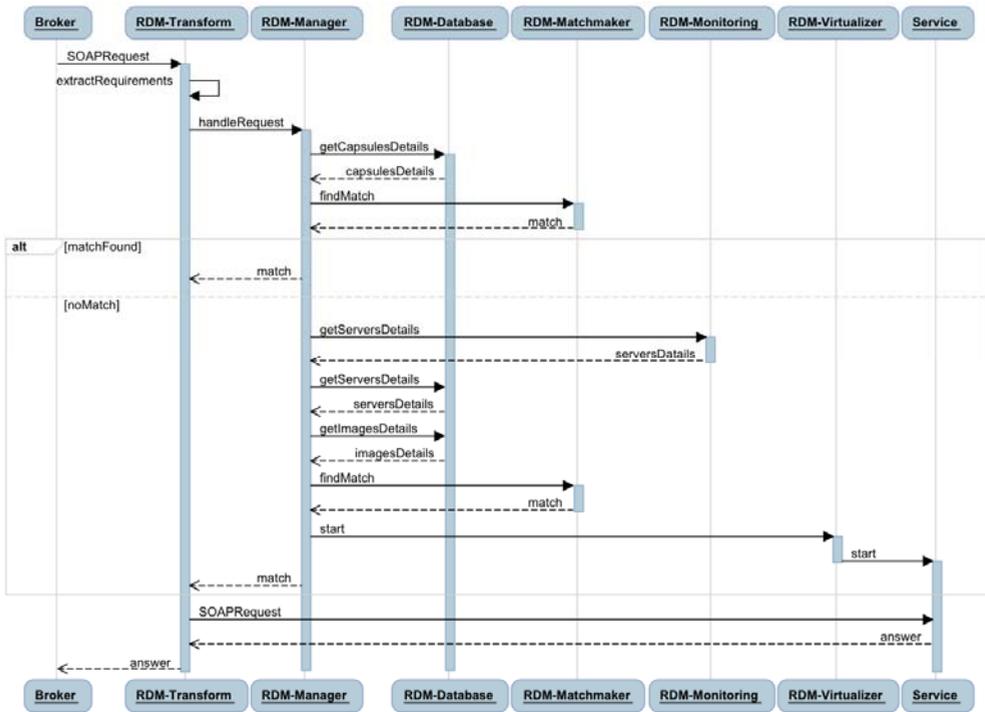


Fig. 2. Sequence diagram of request handling in RDM

Current list of running services (capsules) is stored in the database and it is taken from them using RDM-Database.

```

<env:Envelope xmlns:env=http://schemas.xmlsoap.org/soap
/envelope/>
  <env:Header>
    <pwr:executionControl xmlns:pwr="http://example.pwr.wroc.pl
/exec" env:role="http://www.w3.org/2003/05/soapenvelope/role
/ultimateReceiver">
      <pwr:param>assert system.executionTime<500</pwr:param>
      <pwr:param>assert system.executionTimeUnit==ms</pwr:param>
      <pwr:param>assert system.executionPriority==32</pwr:param>
    </pwr: ExecutionControl>
  </env:Header>
  <env:Body>
    <!-- service spcyfic code -->
  </env:Body>
</env:Envelope>

```

Fig. 3. Sequence diagram of request handling in RDM

Handling the request can lead to one of three situations. There is a running service (capsule) which can perform it and it will be returned as the target to which the SOAP request shall be forwarded. There is no running service (capsule), but there is an image which satisfies the conditions. In such a case the image will be instantiated and it will be used as the one to perform the request. Last possibility is the lack of proper service and image in which case the error will be thrown and finally returned as a SOAP Fault message to the client.

The overall management of service capabilities is manual task, what makes the system still more suitable for testing and research rather than for the production purposes.

3.2. VIRTUALIZATION MANAGEMENT

As it was already mentioned, virtualization management is based on open source *libvirt* toolkit. It offers the virtualization API supporting number of the most popular hypervisors. From the point of view of this chapter it is of a less important how technically the management is performed. It is more important to note what are the capabilities of the management and how it is understood here.

The virtualization management does not mean simply to start or stop the machine. As Jack Welch said about management “... *I only have three things to do. I have to choose the right people, allocate the right number of dollars, and transmit ideas from one division to another with the speed of light*”, what does imply managing being something more complex. The complexity is coming first of all from the decision making problem. Choosing the right people in our case mean finding the right service or image. Allocating right number of dollars is just allocation of proper resources, last but not least to transmit the ideas with the speed of light, so to make our processing as minimal footprint as possible.

3.3. FUTURE IMPROVEMENTS OF RDM

Automation of virtualization management while request dispatching is as for now just a half-ready solution. It creates the instances when needed, but there is no mechanism shutting them down when the number of requests is decreasing. This leads to the proposition of RDM-Controller module.

RDM-Controller tasks would be to prevent the system from taking virtual *sprawl* route, which is pointed as one of the biggest challenges in virtualization [1]. Use of data coming from monitoring could be used determine the instances which are not used and can be stopped. It could also determine the need increase or decrease the amount of memory assigned to particular machines. Furthermore controlling module shall understand service importance automatically starting the services which require constant accessibility. This feature of the service could be easy added as another entry describing the service, using the same notion.

4. TESTING ENVIRONMENT

To test the performance of the system some kind of test environment had to be created. Although most of the code has been written in Python (modules) sample services as well as sample requests are generated using Perl. Tests in current state has been limited to the CPU usage. The services are dummy and all they do is to utilize certain capacity of the CPU during certain amount of time. The solution to make this happen is somewhat not exact and limits the usage properly when set above 5%. It uses simple method to increase the CPU usage to 100% forking *perl* processes and starting never ending loops (figure 4). Such a process would always consume all of the processing power so it is being limited using *cpulimit* tool [7].

```
#!/usr/bin/perl -w
my $use_no_cpus = 1;
while (--$use_no_cpus and fork) {}; while () {}
exit;
```

Fig. 4. Perl code to increase the CPU usage

The number of CPU's to be stressed is fixed in the code, but it is not a problem to pass this number as an argument to the script. Time and CPU usage are limited in the directly in the dummy services implementation. It is done by first starting the script as a new process. It's *ID* is passed to the *cpulimit* tool with desired usage e.g. 50% and as the last step after desired number of seconds the process is killed to free the resources.

4.1. TEST CASE: VIRTUAL *SPRAWL*

The first test case exploits RDM lack of automatic resource freeing. Called service is configured to consume 80% of CPU for 60 seconds what simulates some relatively exhausting operation. The requests are incoming almost in the same time and they require at least 30% CPU capacity to be free (figure 5). The outcome of such a simple simulation is increasing number of instances of the service to handle sudden peak in requests, yet there is no mechanism to limit the number of the instances afterwards.

It turned out that there is also a couple of other issues coming from non existence of the RDM-Scheduler module. Requests coming in the same time (almost the same time) are being treated in similar way. The problem with that is appearing especially when the new instance is to be started. There is no queue to hold the requests from further processing until the instance is really started.

```
<soap:Header>
  <pwr:executionControl xmlns:pwr="http://
```

```

    example.pwr.wroc.pl/exec">
<pwr:param xsi:type="xsd:string">
    server.name=="mercury15";
</pwr:param>
<pwr:param xsi:type="xsd:string">
    capsule.systemCPU.idle>=30
</pwr:param>
</pwr:executionControl>
</soap:Header>

```

Fig. 5. Requirements passed with the SOAP message for test case” “virtual aprawl”

Another issue that appeared is the synchronous handling of the requests. All calls between modules are done this way, what sometimes causes the system to block waiting for some actions (e.g. starting new virtual machine) to finish. In many cases this could be resolved asynchronously, yet not every request can be handled that way and this is independent of the internal RDM processing.

5. CONCLUSIONS

Distribution of the request inside Service Oriented Architecture yields number of issues which involves virtualization management. Automation of such process requires well defined requirements handling, policies and description of service features to be properly matched with request. As it was pointed in this chapter RDM is trying to make such a capability real. Increasing the level of service awareness already on the lower level can make the system more flexible and better adapting to the changes in customers behavior.

To highlight the paths RDM may undertake it is worth to point better capacity management, inclusion of the policies support and asynchronous handling of requests when possible. Nonetheless RDM could already be used to manually set up the environment and automatically handle requests.

Currently, using RDM the following features are available:

- mechanism for the creation and use of services - using the SOA paradigm and virtualization. This makes services independent from the available hardware architecture, and ensures the efficient use of hardware resources
- method of scheduling delivery of services in a virtual machine environment - are taken into account the performance parameters of the virtual machine, service and equipment on which a virtual capsule is installed

- tool architecture and its constituent modules is open, communication takes place via defined interfaces using XML-RPC for internal communication and the SOAP protocol for external communication

The final goal of our work is to develop Service Aware Virtualization Management System which will integrate not only constraints coming from the request but also would be able to deal with service level agreements and adapt to changing conditions dynamically responding to the needs of the service clients.

ACKNOWLEDGEMENTS.

The research presented in this work has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

REFERENCES

- [1] ROSENBERG D., *Analyst: Virtualization management key to success*, http://news.cnet.com/8301-13846_3-10468343-62.html, March 15 2010
- [2] SARGEANT P., *Data centre transformation: How mature is your it?*, http://www.gartner.com/it/content/1282000/1282013/data_centre_transformation_phil_sargeant_17_feb2010.pdf, 2010
- [3] SEMPOLINSKI P., THAIN D., *A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus*, In: Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on , vol., no., pp.417–426, Nov. 30 -Dec. 3 2010
- [4] SNYDER B., *Server virtualization has stalled, despite the hype*. In: InfoWorld.com, <http://www.infoworld.com/t/server-virtualization/what-you-missed-server-virtualization-has-stalled-despite-the-hype-901>, 2010
- [5] SUMAYAH A., *Behind the scenes of IaaS implementations*, <http://salsahpc.indiana.edu/>
- [6] VENEZIA P., WITKOWSKI M., *Pojedynek platform wirtualizacyjnych*, In: Networld, June 2011
- [7] cpulimit, <http://cpulimit.sourceforge.net/>
- [8] Eucalyptus Home Page <http://open.eucalyptus.com/>
- [9] Nimbus Home Page. <http://www.nimbusproject.org/>
- [10] OpenNebula Home Page. <http://www.opennebula.org/>
- [11] OpenStack Home Page. <http://www.openstack.org/>
- [12] OpenStack Compute Administration Manual, <http://docs.openstack.org/cactus/openstack-compute/admin/content/>

BIBLIOTEKA INFORMATYKI SZKÓŁ WYŻSZYCH

- Information Systems Architecture and Technology, ISAT 2005*, pod redakcją Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2005
- Information Systems Architecture and Technology. Information Models, Concepts, Tools and Applications*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2006
- Information Systems Architecture and Technology. Information Technology and Web Engineering: Models, Concepts & Challenges*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2007
- Information Systems Architecture and Technology. Application of Information Technologies in Management Systems*, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2007
- Information Systems Architecture and Technology. Decision Making Models*, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2007
- Information Systems Architecture and Technology. Information Systems and Computer Communication Networks*, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2007
- Information Systems Architecture and Technology. Web Information Systems: Models, Concepts & Challenges*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008
- Information Systems Architecture and Technology. Information Systems and Computer Communication Networks*, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008
- Information Systems Architecture and Technology. Models of the Organisations Risk Management*, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2008
- Information Systems Architecture and Technology. Designing, Development and Implementation of Information Systems*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008
- Information Systems Architecture and Technology. Model Based Decisions*, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2008
- Information Systems Architecture and Technology. Advances in Web-Age Information Systems*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009
- Information Systems Architecture and Technology. Service Oriented Distributed Systems: Concepts and Infrastructure*, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009
- Information Systems Architecture and Technology. Systems Analysis in Decision Aided Problems*, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2009
- Information Systems Architecture and Technology. IT Technologies in Knowledge Oriented Management Process*, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2009
- Information Systems Architecture and Technology. New Developments in Web-Age Information Systems*, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010
- Information Systems Architecture and Technology. Networks and Networks Services*, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010
- Information Systems Architecture and Technology. System Analysis Approach to the Design, Control and Decision Support*, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2010
- Information Systems Architecture and Technology. IT TModels in Management Process*, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2010

**Wydawnictwa Politechniki Wrocławskiej
są do nabycia w księgarni „Tech”
plac Grunwaldzki 13, 50-377 Wrocław
budynek D-1 PWr., tel. 71 320 29 35
Prowadzimy sprzedaż wysyłkową
zamawianie.ksiazek@pwr.wroc.pl**

ISBN 978-83-7493-631-6