Biblioteka Informatyki
Szkół Wyższych

Information Systems Architecture and Technology

Web Information Systems Engineering, Knowledge Discovery and Hybrid Computing

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

Web Information Systems Engineering, Knowledge Discovery and Hybrid Computing

Editors
Leszek Borzemski
Adam Grzech
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

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

ISBN 978-83-7493-630-9

CONTENTS

PART III: INFORMATION SYSTEMS DESIGN PARADIGMS

PART IV: HIGH PERFORMANCE HYBRID ARCHITECTURES

INTRODUCTION

Web information systems engineering, knowledge discovery and hybrid computing are very important challenges in today's research and development. New IT developments and design paradigms in this domain are very essential in business practice as they provide innovative tools to enhance business performance to be competitive on the market.

The growth of the Internet explodes with the development of Web Information Systems (WIS) built with the use of Web technologies. In this book we present issues related to different design aspects of WISs, including definition of the overall architecture, problems with legacy applications interrelationship, user navigation, application functionality and performance, security as well as the implementation issues. Knowledge Discovery provides a key technology for intelligent WISs and has been used in many newly developed intelligent WISs. Many of them need support of high performance computing developments including hybrid processing platforms such as GPUs (Graphics Processing Units) and multicore processors like Cell Broadband Engine (Cell B.E.).

This book consists of 28 chapters presenting a balanced coverage of four challenges of current IT technologies: Web systems engineering, knowledge discovery, information systems design paradigms and hybrid computing.

In Part I, seven chapters cover Web and Internet engineering problems.

In Part II, eight chapters discuss knowledge discovery and engineering problems and solutions.

Part III covers selected problems in information systems design paradigms and applications.

Part IV discusses certain problems of high performance hybrid computing systems architectures and applications.

PART I. WEB SYSTEMS AND INTERNET ENGINEERING

Chapter 1 illustrates Decisional DNA knowledge structure and its main features working together with a web crawler in order to manage web information. It can be used to extract required information from websites and make it reusable, transportable and shareable. Decisional DNA is a domain-independent and flexible knowledge representation structure. Its main features of acquiring and storing experiential knowledge of formal decisional events are used to deal with unexpected situations, especially, convert unstructured data into well-structured knowledge, including, information from websites. Consequently, website owners, users and organization leaders can use it for future decision-making processes based on previous decision events. Experiments are presented to prove the approach.

Chapter 2 brings up the problem of e-commerce Web server performance evaluation using a simulation-based approach. A motivation for choosing this approach is discussed and a simulation tool developed for e-commerce Business-to-Consumer (B2C) scenario is presented. A methodology for carrying out simulation experiments using the tool is discussed, including a new way of collecting statistics necessary to derive performance metrics oriented towards user sessions and revenue. Such business-oriented metrics are of great importance for online retailers.

Chapter 3 describes a novel proposal of the spatio-temporal Web performance forecasting with Sequential Gaussian Simulation (SGS) method belonging to group of the geostatistical simulation methods. The database was created by Multiagent Internet Measurement System MWING. Investigations consider connections of the agent from Gliwice with many www servers located in Europe. A preliminary analysis of measurement data was conducted. Subsequently, the structural analysis, which contains the description of used Gaussian anamorphosis and directional variogram approximated with the theoretical model, was performed. Both of these analyzes are necessary to realization forecast by SGS method. Next, a spatial forecast of the total time of downloading data from Web servers with a one-week time advance was calculated. The analysis of server activity on a particular week day for a period of a few weeks in selected time intervals and considered forecasted errors was performed. Results of forecast were analyzed in detail, followed by the determination of subsequent research directions to improve Web performance forecasts.

Chapter 4 presents a solution to the problem concerning one of the most important issues of today's Internet infrastructure development, namely time efficient getting of same Web resources which are located on different Web sites hosted on Web servers located all over the world. Therefore, the prediction algorithm should be applied in order to make an appropriate choice of a target server to get required resource. This chapter considers such prediction based on the structure of the Internet at the Autonomous Systems level.

Chapter 5 deals with the issues related to the efficient getting of resources on the Web based on the Internet model using the "best performance hit" algorithm. The experimental system has been built that uses Web search engine performance algorithms to determine access time for searched resources. Implemented search engine allows searching for specific terms making the download time estimation of the corresponding resources on the basis of geographical location data and distance at the level of autonomous systems and also based on performance data gathered from group of servers deployed in random locations on the surface of the world.

Chapter 6 provides an evaluation of two currently available open source solutions for Content Aware Networking: Content Centric Network and Network of Information. Available applications and scenarios in both implementations were tested and summarized. Research results include download speed measurements in CCN implementation in 3 different topologies. Best performance achieved in CCN is 3.6 times worse than a reference test made with SSH protocol.

Chapter 7 paper presents new, general concept of data storage system based on network environment with a special emphasis on data security. This problem is considered as a one of the basic problems to be developed to support current new networking based processing paradigms like cloud computing and Web computing. The concept is based on set of user requirements such as level of security, performance, cost and set of security measurements and technology features (quantitative as well as qualitative). The question is if it is possible to meet the user requirements with a use of available security measurements, technologies and acceptable cost. Security is usually defined with a use of the requirements: confidentiality, integrity and availability. The requirements are well known but new features, factors and architectures of storage system encourage to look for new, pragmatic explanations of them. It may be simply said that the requirements should be always met. In network storage systems this means fulfilling the requirements in every data location (local, remote), in every data state (storage, transmission) and in every operation performed on data.

PART II. KNOWLEDGE DISCOVERY AND ENGINEERING

Chapter 8 introduces a new approach called Decisional DNA Digital TV in order to capture, reuse, and share the viewers' TV watch experience and preference. Interactive television is an evolutionary integration of the Internet and digital TV. It allows viewers to interact with television content and services. Thanks to the booming of digital TV, viewers' TV watch experience could be extremely valuable. By running customized applications either inside the digital TV or at viewers' set-top boxes, it is possible to capture viewers' TV watch experience. This chapter presents the features, architecture and initial experimental results of this approach as well as an introduction to the Decisional DNA which is a domain-independent, flexible, smart knowledge representation structure which allows its domains to acquire, reuse, evolve and share knowledge in an easy and standard way.

Chapter 9 proposes to add knowledge management perspective to all five main Business Process Management (BPM) activities: identifying, mapping, measuring, analyzing and redesigning the processes in an organization. BPM as a holistic management is one of the most promising approaches to management in general, but there is still a question of how can we address knowledge management issues in this approach. Knowledge Management and Business Process Management have both similarities and differences, but in authors' opinion they should be used together to help with the conversion into self-learning organizations in knowledge based society.

Chapter 10 discusses the issue of information resources integration. The problem of organisational integration as well as various solutions are described. Four levels of integration are defined, existing strategies are discussed and some examples of models and systems integrating various information resources are mentioned. The proposed Heterogeneous Information Resources Integrating Model (HIRIM) is characterised: its architecture, possible versions and properties. The author proves that designing such a model or system it is necessary to compromise which leads to the solution never meeting all expected needs.

Chapter 11 presents the problem of extraction of elementary facts from complex sentences in natural language. In learning stage the system is processing complex sentence and corresponding set of simple sentences, reflecting elementary facts expressed in complex sentence. Link Grammar is used for syntactic analysis. Obtained syntactic structures are supplemented with semantic features and thus constitute syntactic-semantic images. During process of image matching, patterns of elementary facts are identified within the image of complex sentence and stored in knowledge base. In working stage, similar image of the new, analyzed sentence is created. This image is matched with patterns of already known elementary facts taken from knowledge base. Successful matching means that new elementary fact was discovered.

Chapter 12 presents the problem of monitoring computer system operation based on various event logs. The authors concentrate on Linux systems and available standard log programs. The basic idea is to collect event logs from many computers and correlate them with operational and user profiles. To deal efficiently with the bulk of collected data a special own software module QLogAnalyser was used. This software provides useful capabilities to deal with variety of log formats and facilitates detecting interesting situations using regular expressions. It supports also visualisation of various statistics. The practical usefulness of the developed approach to system monitoring has been illustrated with results related to one of didactic laboratories in the Institute.

Chapter 13 presents the system developed to aid modern Internet advertising. The goal of the system is to provide an ad based on the previous activity of the user. Gathering and processing various data concerning this activity is the core task of the system.

Chapter 14 indicates the need to support University Competence Centers. These are specialized centers which aim to support teaching and the development of the latest software technologies, educating students, PhD students and scientific workers and also raising the competitiveness of the university on the higher education market. The work presents the characteristics of: IBM Rational RequisitePro, IBM Rational Requirements Composer, IBM Rational Team Concert as systems to manage requirements around the product development cycle.

Chapter 15 presents the analysis of ten domestic companies from the point of view of the considered relation. Web pages were used to collect required financial information. Capital structure is an important factor of financial management of the firm. Capital structure decisions have a strategic meaning and for that should be well prepared and using of debt capital rational and reasonable. This chapter concerns making analysis of capital factor influencing the firm value and discusses relation between capital structure and company value, which future growth is the main reason for capital structure optimization.

PART III. INFORMATION SYSTEMS DESIGN PARADIGMS

Chapter 16 presents the advantages and disadvantages of the methodologies used in software companies as well as their popularity. The data have been obtained in a survey conducted among IT firms. It seems evident that agile methodologies are becoming more and more popular. It is due to the fact what it was strongly expressed by the IT firm managers that the client does not understand design diagrams and is mainly interested in system interface construction of a functional prototype. Furthermore, the most surprising observation is negligent attitude towards system documentation.

Chapter 17 proposes how to put together vectors of features for segmented image objects and a spatial relationship of the objects by constructing a multi-step search-engine, taking into account multi-set data mining and the object spatial relationship for the Content-Based Image Retrieval System (CBIR). The research presents a combination of two aspects of image representation, namely features of segmented objects at lower level and spatial relations of objects to compare image similarities at a higher level. The new representation of spatial relationships of the image objects is based upon the Principal Component Analysis (PCA). It makes the method invariant to image rotation. The efficiency of the system is being evaluated.

Chapter 18 presents a conception of interactive information and decision support system for urban and industrial air quality management. The emphasis of the project is on real-time analysis and multi-media information, and the support of distributed and mobile clients through the Internet. The approach integrates meteorological data and forecasts, air quality and emission monitoring, dynamic 3D simulation modeling and forecasting, GIS, expert systems, decision support and reporting tools in a unified, modular client/server framework implemented as a range of web accessible application services.

Chapter 19 shows how the way of counteraction is offered to not authorized access with use of an algorithm filtration audit events information system. The control of integrity confidential data and audit of events in information system allows to prevent and identify a significant amount of attacks and also to reduce the financial losses connected with information leakage.

Chapter 20 discusses the problem how information systems allow the enterprise to carry out as the internal financial analysis, so external from a position of the budget, supervising departments, creditors and shareholders.

Chapter 21 deals with the contract-based programming. The Object Constraint Language (OCL) supports contracts assigned to models. Using OCL, invariants of classes, pre- and post-conditions of operations, as well as constraints in behavioral models can be specified. Various tools provide object-oriented modeling with UML and OCL, but only scarce of them can generate code of OCL contracts. This chapter evaluates their capabilities to transform OCL contracts into the corresponding code. The tools were compared in respect to different structures of OCL, contract inheritance, target languages, programming techniques realizing contracts, and system reactions in case of not satisfied contracts. Limitations in realization of contracts in the OCL tools were discussed. The reviewed tools were related to T.O.F.I.C – a new tool that transforms UML class models, refined with the C#-aimed profile and specified with OCL expressions, into the appropriate C# code. In this approach, OCL contracts are established with the Microsoft Code Contracts library of .NET.

Chapter 22 touches the fault injection systems which are valuable tools for evaluation of target systems' dependability if the proper set of faults are injected into the system under tests (SUT). Among many advantages, one typical drawback of many fault injection systems is their limited functionality enclosed by the hardcoded fault models or specific target properties. New fault injection scenarios usually require tool modification. The chapter presents the novel scripting language that overcomes this issue, called InScript that is dedicated for software implemented fault injection systems. It allows to flexibly define the desired fault model and the after-injection behavior of the fault injection tool for optimal observation of the fault effects in the SUT. The proposed InScript language was successfully implemented in some fault injection tools and exemplary experiments are reported.

PART IV. HIGH PERFORMANCE HYBRID ARCHITECTURES

Chapter 23 introduces a library that simplify programming in OpenCL. OpenCL applications can be written with 5–12 times less code by using this library. In recent years powerful multi-core processors and programmable graphics processing units (GPUs) have become cheap and ubiquitous. It is common to see commercial gaming consoles or desktop PCs achieve peak single precision Gflop/s ratings in the range of hundreds, if not thousands, by means of employing various parallel architectures. This influx of high-performance yet relatively cheap, readily available compute devices fuels both the consumer market and the scientific supercomputing niche. It is desirable to exploit their high-performance and low unit cost by writing applications for scientific, engineering and commercial use. OpenCL is a framework for developing portable parallel applications for a wide variety of hardware devices (multi-core CPUs, programmable GPUs). Developing in native OpenCL is, however, difficult and requires writing a large amount of code even for simple programs. Therefore a library presented in this chapter may be very valuable for programmers.

Chapter 24 concerns design and implementation of parallel algorithms in computation system based on multicore processor Cell B.E. The problem of processing of parallel algorithms for logic-algebraic method on Cell B.E. processor is exploited. The specific structure of processor and its communication mechanisms impose a careful design of processing architecture. Presented solutions can be regarded as guidelines for design of parallel processing of any task with a similar pattern of computation. Proposed architecture was implemented in real system and some results of measurements of processing efficiency is finally presented.

Chapter 25 contains the comparison of CISC and RISC architectures in Web-based decision support systems. Not only does its first part elucidates those two architectures, but it also illuminates operating systems that can be used on the latter. The second part of this chapter introduces four computational programs solving problems that may be found in above-mentioned systems: the aggregation of autonomous systems, the prediction of a time needed to download resources, elementary arithmetic operations and sorting algorithms. The final part embodies obtained results and introduces an absolute quality indicator.

Chapter 26 presents a model and simulation experiment of a computer system survivability. Model parameters were taken from legal regulation and risk analysis. Requirements to the system recovery time and accessibility are set by regulation, based on the computer system category. The simulation was done by using stochastic activity networks. Simulation results show that the modelled computer system security mostly depends on the incident occurrence probability, on the strength of protection mechanisms, while the occurring incident severity has the least effect on the protected computer system.

Chapter 27 shows a model of a system for automated tests based on the concept of exchangeable, hierarchical resources that are described using specially created rule system. In order to make test cases work in different environments a multi- layer structure of test scripts was introduced that separates general test steps from their platform dependent implementation. A prototype was constructed to verify the proposed concept and algorithms. Results showed that it is possible to create such tools that can generate all valid configurations and flexibly control test environment coverage by implementing the paradigm: write once, run on many.

Chapter 28 deals with a Snapshot technique of preserving consistent in time state of selected object. This technique is often used to create archive containing consistent in time files backup from filesystem, that continuously on-line updating data. When snapshot storage space is fulfilled, snapshot does not provide consistent in time filesystem state and is useless. This disadvantage can occur only when snapshot storage space size is smaller than filesystem storage space size, in practice it is the most frequent case. The obtained results allow users using snapshots to choose an appropriate filesystem in storage space preparation process and also estimate the required size of storage space used by the snapshot before file backup creation.

This book contains the contributions accepted after the review of authors' submissions. We hope that the book will be considered as a forum for presentation of original and professional work in up-to-date research areas including Web systems, Internet, knowledge discover and engineering; information systems design paradigms; and high performance processing on hybrid architectures.

We would like to express many thanks to revisers who helped to evaluate the submissions.

We thank all the authors who have submitted their contributions to be published in this book.

Wrocław, September 2011

Leszek Borzemski

PART I

WEB SYSTEMS
AND INTERNET ENGINEERING

Decisional DNA, Set of Experience Knowledge Structure, Web Data Mining, Web Crawler

Peng WANG*, Cesar SANIN*, Edward SZCZERBICKI*

WEB CRAWLING EXPERIENCE MANAGEMENT: DECISIONAL DNA PERSPECTIVE

Internet is becoming the most important information source. However, website information is primarily semi-structured and unstructured, and consequently, it is difficult to obtain the desired information. Decisional DNA is a domain-independent and flexible knowledge representation structure. Its main features of acquiring and storing experiential knowledge of formal decisional events are used to deal with unexpected situations, especially, convert unstructured data into well-structured knowledge, including, information from websites. This work illustrates Decisional DNA knowledge structure and its main features working together with a web crawler in order to manage web information. It can be used to extract required information from websites and make it reusable, transportable and shareable. Consequently, website owners, users and organization leaders can use it for future decision-making processes based on previous decision events. Experiments are presented to prove its efficiency and efficacy.

1. INTRODUCTION

Nowadays, the Internet is one of the main sources through which the "information age" can become a reality. However, extracting useful and explicit information from massive unstructured and semi structured data is still a contentious topic. Furthermore, user's desired information hidden in abundant and useless spam on websites needs to be extracted precisely [6]. It has become increasingly necessary for users to utilize appropriate techniques in finding the desired information and usage patterns. On this basis, web data mining research has become a hot spot in the high technology domain. It confronts extraction of useful knowledge in order to guide the decision-making from web-based data [6]. This work introduces a novel and explicit way, combining web

---

* School of Engineering, Faculty of Engineering and Built Environment The University of Newcastle, Australia.

crawler and Decisional DNA [7, 8], which has the ability to easily obtain and reuse different structured knowledge from the web.

2. BACKGROUND

2.1. SET OF EXPERIENCE KNOWLEDGE STRUCTURE (SOEKS) AND DECISIONAL DNA

Web Data Mining is currently working with different types of knowledge. The idea behind it is to store and manage knowledge in some manners. In other words, mining web data is the process of storing, retrieving, distributing and sharing knowledge. However, web information is mostly unstructured or semi-structured in huge quantities. Thus, a technology which can be used to capture and store formal decisional events as explicit knowledge is necessary. The Set of Experience Knowledge Structure (SOEKS or shortly SOE [9, 11, 14]) as a flexible and independent knowledge representation is a suitable tool for this task. Moreover, it also has been used to collect and store formal decisional events in an explicit manner [9]. Therefore, the SOEK can be a pattern based on existing and available knowledge offered by a formal decision event with dynamic structure. It can be expressed in XML or OWL as ontology in order to make it shareable and transportable [10-12].

The SOEKS is composed of variables, functions, constraints and rules [8]. Variables commonly use an attribute-value language to represent knowledge (i.e. by a vector of variables and values) [4]. It is the starting point for the SOEKS and the infrastructure of the SOE because they are the source of other components. Functions are made up of interactions of variables which include dependent variables and a set of input variables. On the other hand, according to the tasks of the decision event, functions are brought to reasoning optimal states. Therefore, this second component of the SOE establishes the relations between variables restricting experience on decision-making. Constraints are another factor of association amongst the variables. Though constraints are another way of functions, they have a different purpose. They limit the performance and possibility of a system and restrict the feasible solutions in a decision problem. Lastly, rules are another form of expressing links among variables. They condition the relationships that operate the universe of variables. In other words, they use the statements IF-THEN-ELSE to connect a consequence with a condition.

Additionally, the SOEKS is structured in view of some important features of DNA. Firstly, the combination of the four components of the SOE offers distinctiveness, just corresponding to the combination of the four nucleotides of DNA. Moreover, the elements of the SOEKS imitate a gene to connect with each other. In the same way as a gene produces a phenotype, the SOE yields a value of decision with their elements. Each SOE can be categorised and acts as a gene in DNA [9]. A set of SOE in a same category makes up of a decisional chromosome which stores decisional strategies for that category. After this, each module of chromosomes establishes an entire inference tool to offer a blue print of knowledge inside an organization [8].

2.2. WEB DATA MINING

Web Data Mining is the process of discovering and extracting useful information or knowledge from the Web including web hyperlink structure, page content and usage data [3]. It is an inclusive technology in which several domains are involved, such as Web, data mining, computational linguistics, statistics information standard and other fields of science. In other words, web data mining techniques can be used to analyse the content of documents, the use of available resources, to find effective, potential, valuable, understandable and explicit patterns of knowledge by combining methods of statistics and artificial intelligence with database management [6, 13]. According to different mining tasks, there are three important aspects of web data mining: web usage mining, web structure mining and content mining. Their detailed structure is illustrated after [6] as follows (see Fig. 1):

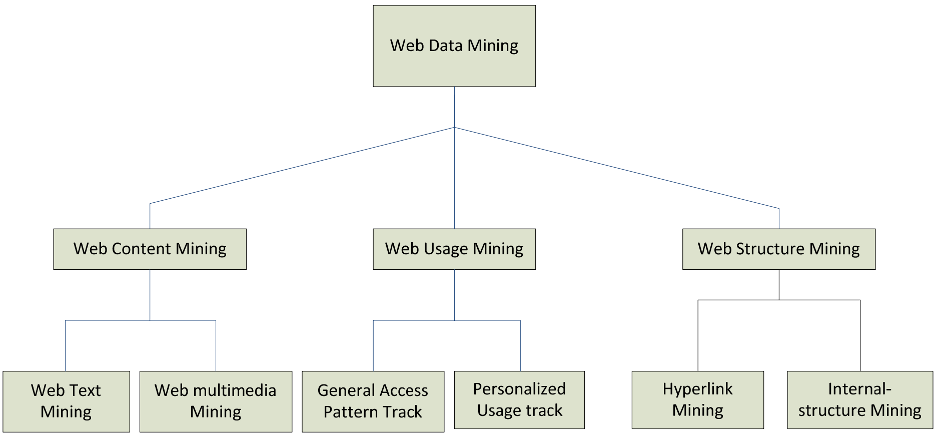


Fig. 1. Classification of Web Data Mining [6]

Web content mining comprises the process of discovering useful pinformation from text, image, audio or video data in the web [3]. Furthermore, it is also a data mining technique which is different from traditional data mining techniques because it is primarily for a variety of unstructured data such as text, voice, video and graphic image and so on. The current study of Web content mining is mostly based on document data.

Web usage mining is the ability to automatically discover web user access patterns from web server log files which record the users' access to data. General access pattern track for user groups and personalized use record track for single user are used to analyse users' browsing patterns. Generally, servers log data should be aimed to mine. These data includes a client-IP, server-side data, authoritative page and data-side proxy. Generally, it uses server log files to find interesting patterns of visiting web sites, which helps on understanding users' behaviour. And, in consequence, it supports websites improvements or personalization of users' service.

Web Structure Mining is the process of analysing node and connection structures of a website by using graph theory. In other words, it acquires knowledge from the organizational structure of a website and the relations among the links. For instance, web structure mining techniques can be used to index pages and seek the most useful pages among them. Web structure data mining is composed of two kinds. One is extracting hyperlinks between web pages or documents and the other one is mining the internal document by analysing the page structure's tree link structure [6].

2.3. WEB CRAWLER

Web crawlers are tools to automatically gather webpages from certain web sites with an orderly pattern. The process of crawling is to use seed URLs to download web pages related to these URLs. Then, they recursively extract and download web pages according to any hyperlinks identified from the URLs. One core component of web search engine is the Web Crawler. Thus, it can be used to assemble the web pages sorted by the search engine. Therefore, many applications apply it to deal with large numbers of web pages including web data mining, comparison shopping engines among others. Major engineering challenges have been bright worth by implementing high-performance web crawler, though its principle is simple [5].

Heritrix is an instance of web crawlers. It is an extensible, web-scale and archival-quality open source. It is divided into three striking aspects such as the Scope, the Frontier and the Processor Chains [1]. It brings initial information to creating the Scope with seeds. The seeds contain initial URIs which can be consulted by the Frontier. The Frontier is responsible for which URIs should be ordered to be visited according to the Scope seeds. It maintains a series of internal queues of URIs, ensuring URIs to be not already-scheduled and only choosing the URIs scheduled to be collected.

3. THE DECISIONAL DNA-BASED WEB CRAWLER

Nowadays, Internet has been developing very rapidly. There is a huge requirement for sharing, storing, reusing and analysing knowledge among the websites. Using the Web Crawler technique with Decisional DNA is a novel and explicit way for organizations or website owners dealing with their increasingly unstructured number of information. It not only shares knowledge, but also assists in the decision making process.

3.1. ARCHITECTURE MODULE DESCRIPTION

SOEKS can be implemented by an architecture that contains four Macro Processes [7, 9] as shown on the top of Fig. 2 introducing the architecture of the proposed Decisional DNA based web crawler. Those processes are respectively diagnosis, prognosis, solution and knowledge. This work describes the necessary key components for a Decisional DNA based Web Crawler by using the above four macro processes. Functions and responsibilities of components are explained as follows.

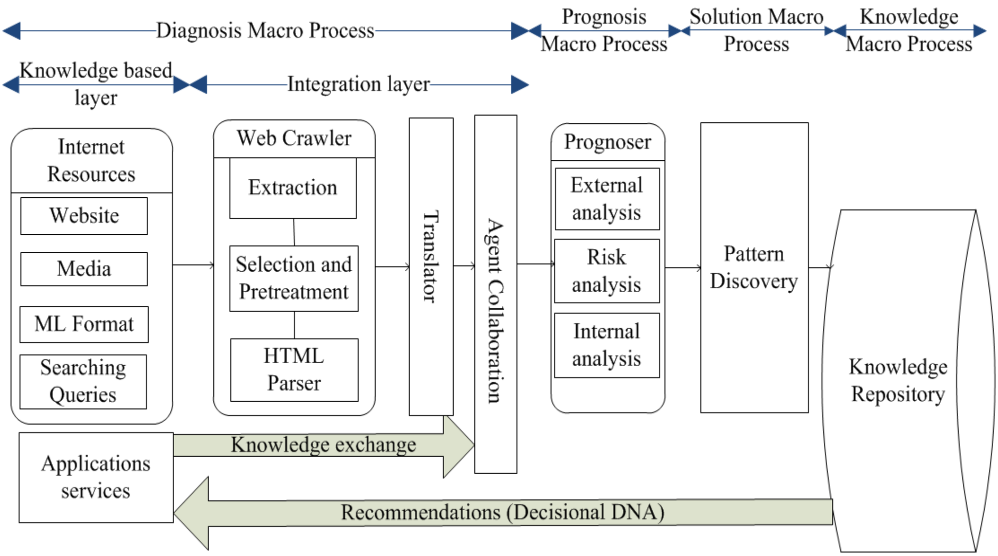


Fig. 2. Architecture of Decisional DNA-Based Web Crawler

The diagnosis Macro-process is composed of knowledge based layer and integration layer. Internet Resources are a component in the knowledge-based layer. As internet contains enormous information, it is useless if information is infinite and non-structured. However, internet sources, as well, appear in several forms. For example, website documents, emails, metadata, XML files, users' visited logs. Several resources can be related to an organization's knowledge discovery. Thus, the purpose of this component is to define and analyse the scope of the website. Then, using the Web Crawler component, it extracts knowledge in the integration layer as shown in Fig. 2.

Organizations, users or web owners may need to cooperate with multiple applications inside or outside a company. It is increasingly important to exchange information and knowledge quickly and safely between different applications. Therefore, it needs unified and shareable knowledge. XML is a simple, widely used, transportable and applicable language for sharing knowledge among applications; Decisional DNA formatted in XML language can be exchanged to multiple application services through the Agent Collaboration component (see Fig. 2).

Web Crawler is a component in the integration layer. It extracts certain information from given Internet Resources seeds. Meanwhile, it needs to analyse each given website hyperlinks to find valuable information. Heritrix is used at this stage. It is also responsible for removing useless information like ads or redundant tag format from each page. Afterwards, by using HTML Parsers, it extracts required information from web pages and then passes it to the XML Parser component (see Fig. 2).

Translator is used to convert gathered information from the Web Crawler to the Decisional DNA format. In this case, we use it to transform crawled and selected knowledge into the Decisional DNA XML format. Such process ensures that collected information is formatted accordingly to the SOEKS. At next step, that information can be stored through the agent collaboration component.

Agent Collaboration (Fig. 2) is a container used to collect knowledge from the Translator or application services. Such knowledge must be already in Decisional DNA format. Then, it transfers collected information to the Prognoser for further extraction.

Prognosis process performs a homogenization and unification of information to implement a Multi-objective Evolutionary Algorithm (MOEA) [9]. It generates a holistic group of Sets of experience from which a solution can be chosen. The Prognosis process can be divided into three analyser layers. The Internal analysis layer is responsible for evaluating variables which can be controlled and modified for the website owners. However, the External analysis layer focuses on uncontrollable and unmodified variables. The Risk analysis layer is used to deal with uncertainty, imprecision and incompleteness of the models produced by the previous two layers.

The Pattern discovery layer is to find best solution from a set of solutions offered by the Prognosis guiding organizations' leader to make decisions. This layer offers a range of indexes from where the user can choose priorities of the variables within the Decisional DNA such as value of truth, imprecise index, important variables and weights associated with variables.

Knowledge is stored in the Knowledge Repository after the pattern discovery process. At this stage, information becomes desired knowledge which can be shared and transferred among different applications. Knowledge is stored according to the Decisional DNA structure. In other words, a single set of experience is a gene of knowledge. A decisional chromosome is composed by many of these genes. And many chromosomes comprise a Decisional DNA. The purpose of the Knowledge Repository is to store and maintain several different Decisional DNAs in order to make them reusable, shareable and transportable among application services.

3.2. EXPERIMENT AND CASE STUDY

Our plan was to mine the movie website http://www.imdb.com/ and find useful knowledge which can be reused, shared and transported among diverse applications. Three techniques were implemented in this experiment: the Heritrix, the DOM4J Parser and the Set of Experience Knowledge Structure (SOEKS), providing as a result web mined extracted knowledge and placed it in a SOEKS form in order to construct a DDNA for movies.

Methodology of the Experiment: In the proposed platform, the diagnosis process includes two layers: knowledge-based layer and integration layer. The first step starts in the knowledge-based layer. The purpose of the experiment is to gather information about the top 250 movies from the website Imdb (http://www.imdb.com). Hyperlinks of the desired web pages are in the web page IMdb Top 250 (http://www.imdb.com/chart/top) (see Fig. 3). It can be seen that there are many ads, tags and pictures that are not of our interests in this page. We only want to gain access to desire hyperlinks pages inside the red rectangle. Thus, we need to find a pattern for those hyperlinks. If we click any one of those hyperlinks, all pages are from the same URL (http://www.imdb.com/title/). This result in three URL seeds must be provided to the Heritrix. They are: http://www.imdb.com, http://www.imdb.com/chart/top and http://www.imdb.com/title/. This task can be done by establishing a Frontier class which simply inherits the FrontierScheduler class first within the Heritrix software. We rewrite the schedule method of this new Frontier to accomplish our goal. When this Frontier class is established, we modified the “Process.option” file in the folder of “conf\modules” in order to be configured by the Heritrix’s web interface.

Fig. 3. Web page-IMDB Top 250 [2]

Now, we already got the interest web pages (see Fig. 4). The next step is to select and extract necessary information from those pages. The organizations or the website owners may only have interest in information like “description, title, stars, director, genre, score and ranking” which is showed in Fig. 4 in red rectangles. Therefore, this can be done by using an HTML Parser to extract desired information from tags. For example, when we view the source code of the desired page, the desire information is inside tags such as: “<title>The Godfather (1972) – IMDb</title>”. And it can be simply extracted by the HTML parser.

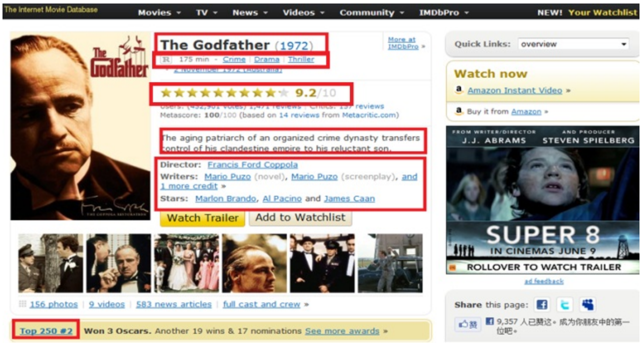


Fig. 4. Fields of interest in the web page [2]

In our experiment, we chose the MirrorWriterProcessor class to store the required files. Therefore, we had to rewrite MirrorWriterProcessor in order to filter the information. In this class, we use a HTML Parser to acquire certain information, and then use DOM4J to translate such information into XML with the required Decisional DNA structure. For our purposes and as an example, the title must be stored as a variable and follows the SOEKS variable’s structure [7].

```
<variable>
    <var_name>title</var_name>
    <var_type>CATEGORICAL</var_type>
    <var_cvalue> The Godfather (1972) </var_cvalue>
    <var_evalue> The Godfather (1972) </var_evalue>
    <unit></unit>
    <internal>false</internal>
    <weight>0.0</weight>
    <l_range>0.0</l_range>
    <u_range>0.0</u_range>
    <categories>
        <category></category>
    </categories>
    <priority>0.0</priority>
</variable>
```

There are six variables acquired as discussed above to separately indicate title, description, stars, director, ranking and genre. Those variables are iteratively and automatically stored in a SOEKS-XML file as a gene until crawling is finished. Next step, we use the Prognoser to analyse those variables and find best solutions for different purposes but that is part of a future work. Finally, the Decisional DNA is stored in the Knowledge Repository which can be reused or transported to other applications.

This decisional DNA-based web crawler is implemented purely in java on windows 7 operational system. It holistically traversed the website in 2 hours 15 minutes 58 seconds and totally discovered 27645 URIs, each of them consumed 0.2 second to gather the required movies' information. We identify that three factors affect the web crawling speed. They are internet connection, web services' limitations and capability of websites. All those factors occur in any web crawler component. However, the gathered information is effectively and efficiently converted into Decisional DNA structure with minimal time consuming. In consequence, there will be a better performance when those factors are reduced; nevertheless, reducing those factors is not part of our research. Finally, Those 250 movies were converted to decisional DNA-based structure which is able to be reuse for any purposes by multiple applications.

4. CONCLUSION AND FUTURE WORK

This work illustrates an experimental structure of web data mining combined Decisional DNA with a Web Crawler. This new structure can be used to extract information from websites and convert it into knowledge which can be reused or shared with different systems.

This research represents an initial stage of combining the Decisional DNA with web data mining techniques. Future research will focus on the following:

– Refinement of the requirements of Decisional DNA for dynamic web data mining. Interaction of web site with Decisional DNA and assistance of decision making needs to be researched in detail.

– Experiencing extraction and inference of multimedia data from web data mining.

– Exploration of Prognosis, Solution and Knowledge processes and findings a way to make websites able to automatically gain knowledge from visitors. Therefore, it can simulate user's behaviour to adapt different clients' needs.

REFERENCES

[1] MOHR M.K.G., STACK M., RANITOVIC I., *Introduction to Heritrix, an Archival Quality Web Crawler*, in 4th International Web Archiving Workshop (IWAW04), Bath (UK), 2004.

[2] I. IMDb.com. (2006). *IMDb Top 250*, Available: http://www.imdb.com/chart/top

[3] LIU B., *Web Data Mining:Exploring Hyperlinks, Contents, and Usage Data*, Springer Berlin–Heidelberg, 2007.

[4] LLOYD J.W., *Learning Comprehensible Theories from Structured Data*, in Advanced Lectures on Machine Learning, Vol. 2600, S. Mendelson, A. Smola, Eds., ed: Springer Berlin–Heidelberg, 2003, pp. 203–225.

[5] NAJORK M., *Web Crawler Architecture*, Springer Verlag, September 2009.
[6] MUKTHYAR M.K.K.S. azam, RASOOL Shaik, JAKIR Ajam S., *Web data mining Using XML and Agent Framework*, IJCSNS International Journal of Computer Science and Network Security, Vol. 10, p. 5, 2010.
[7] SANIN C., *Decisional DNA and the Smart Knowledge Management System: Knowledge Engineering and Knowledge Management applied to an Intelligent Platform*: LAP Lambert Academic Publishing 2010.
[8] SANIN C., MANCILLA-AMAYA L., SZCZERBICKI E., CAYFORDHOWELL P., *Application of a Multi-domain Knowledge Structure: The Decisional DNA*, in: Intelligent Systems for Knowledge Management, Vol. 252, N. Nguyen and E. Szczerbicki, Eds., ed: Springer Berlin–Heidelberg, 2009, pp. 65–86.
[9] SANIN C., SZCZERBICKI E., *Experience-based Knowledge Representation: SOEKS*, Cybernetics and Systems: An International Journal, Vol. 40, 2009, pp. 99–122,
[10] SANIN C., SZCZERBICKI E., *Extending Set Of Experience Knowledge Structure into a Transportable Language Extensible Markup Language, Cybernetics and Systems: An International Journal,* Vol. 37, 2006, pp. 97–117.
[11] SANIN C., TORO C., SZCZERBICKI E, *An OWL ontology of set of experience knowledge structure*, Journal of Universal Computer Science 13(2), 2007, pp. 209–223.
[12] TRONG HAI DUONG N.T.N., GeunSik JO, *Constructing and mining a semantic-based academic social network*, Journal of Intelligent & Fuzzy Systems, Vol. 21(3), 2010, pp. 197–207.
[13] WANG J., HUANG Y., WU G., ZHANG F., *Web mining: knowledge discovery on the Web*, in: Systems, Man, and Cybernetics, 1999. IEEE SMC ‘'99 Conference Proceedings. 1999 IEEE International Conference on, Vol. 2, 1999, pp. 137–141.
[14] ZHANG H., SANIN C., SZCZERBICKI E., *Decisional DNA applied to robotics*, presented at the Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part II, Cardiff, UK, 2010.

Web server, Web performance, Quality of Web Service, QoWS, electronic commerce, e-commerce, B2C, metrics, simulation

Grażyna SUCHACKA*, Leszek BORZEMSKI**

SIMULATION-BASED PERFORMANCE STUDY OF E-COMMERCE WEB SERVER SYSTEM – METHODOLOGY AND METRICS

The chapter brings up the problem of e-commerce Web server performance evaluation using a simulation-based approach. A motivation for choosing this approach is discussed and a simulation tool developed for e-commerce Business-to-Consumer (B2C) scenario is presented. A methodology for carrying out simulation experiments using the tool is discussed, including a new way of collecting statistics necessary to derive performance metrics oriented towards user sessions and revenue. Such business-oriented metrics are of great importance for online retailers.

1. INTRODUCTION

Evaluation of Web server performance is currently a topical research issue in the area of Quality of Web Service (QoWS), connected with a limited capacity of Web servers and their overloads. A lot of research in that area has been done in recent years. In particular, a number of mechanisms have been proposed to improve Web server performance under overload. This brought up the need to develop effective and inexpensive methods for evaluating effectiveness of new QoWS mechanisms for Web servers. This problem is especially apparent in the case of Web servers for e-commerce Business-to-Consumer (B2C) applications, which require taking user session- and revenue-related aspects into consideration.

* Opole University of Technology, Institute of Automatic Control and Computer Science, K. Sosnkowskiego 31, 45-272 Opole, Poland.

** Wroclaw University of Technology, Institute of Informatics, Wybrzeże Wyspianskiego 27, 50-370 Wroclaw, Poland.

The rest of the chapter is organized as follows. In Sect. 2 we argue a choice of the simulation-based approach to Web server performance evaluation. Section 3 presents our simulation tool, while performance metrics and a research methodology are discussed in Sect. 4 and 5, respectively. We conclude in Sect. 6.

2. MOTIVATION FOR A SIMULATION-BASED APPROACH TO WEB SERVER PERFORMANCE EVALUATION

Operation of a Web server may be described using analytical modeling. Analytical models of Web servers are based on a set of formulae and computational algorithms relate workload and system parameters to performance metrics. Such models are typically based on the queuing theory and queuing networks. However, the analytical approach is rarely applied to the request service in Web servers due to the big complexity of these systems and the Web traffic. These factors require introducing many assumptions and simplifications, which only roughly correspond to request service in real servers (c.f. [1, 7]). That is why a common practice is to set up experimental verifications using prototype Web testbeds or to employ the simulation modeling.

The experimental approach consists in using a prototype testbed and computing Web server performance metrics in reaction to a specially prepared workload. Major disadvantages of this approach include high cost and difficulties with an access to the appropriate hardware, especially when the system efficiency has to be evaluated for a variety of system configurations. In such experiments, Apache Web server [13] is typically used, since its open source architecture allows one to easily implement control mechanisms in an operating system kernel or at the application level (c.f. [10, 11].

In the face of the above-mentioned disadvantages a very attractive research method is a simulation-based approach. In fact, it has been widely used in QoWS research in recent years, e.g. in [3, 5]. Simulation models are computer programs which emulate clients' requests processing at various components of a Web server system, including the request generator. A fundamental issue is an accurate modeling of system bottleneck resources and the client behavior. A simulation model is usually solved through a discrete event-driven simulator, which allows one to carry out experiments for a variety of system configurations and parameters. Such experiments may be generally very time-consuming. As opposed to the analytical and prototype-based approaches, however, the simulation-based approach gives a possibility of detailed modeling of very complex systems, such as e-commerce Web server systems, and does not involve high cost. That is why we decided to use this approach and to develop a simulation tool allowing one to evaluate the performance of a B2C e-commerce Web server system under different scheduling policies.

It has to be noticed that there are some Web benchmarks freely available for non-commercial use, e.g. httperf, SPECweb99, SURGE, S-Clients, WebBench, and WebStone. They are able to generate Web workload and collect some statistics on simulation results. However, the analysis of these benchmarks has indicated their low suitability for e-commerce Web servers, mainly due to very simplified workload models and an incapability of providing session- and business-oriented performance metrics. Only TPC-W benchmark specification defines a workload model oriented to e-commerce transactions [6]. However, it does not model neither HTTP-level Web workload nor details of the Web server resources usage at the HTTP level. Furthermore, available TPC-W implementations [14, 15] do not support different customer profiles nor provide business-oriented metrics.

3. SIMULATION TOOL

Based on up-to-date literature results we worked out a workload model and a Web server system model for a B2C scenario [2]. Then, we implemented the models in our simulation tool, which was to provide the following functions:

1) Generating many concurrent user sessions at a given session arrival rate and collecting session-oriented statistics.
2) Differentiating between two session classes: heavy buyers (who are key customers in a B2C scenario) and occasional buyers ("ordinary customers").
3) Modeling the interaction between users and a Web site, especially the impact of the Web server system performance on customer behavior (i.e. the impact of long page response times on the users' willingness to continue their sessions).
4) Generating highly variable, bursty Web traffic at an HTTP level for a mix of static, dynamic and secure requests typical of B2C Web sites.
5) Providing both system performance metrics (such as the system throughput or response times) and business-oriented metrics (related to the achieved revenue and successfully completed key customer sessions).

The simulation tool was implemented in C++ using a package CSIM19, a professional toolkit for modeling complex systems [4]. CSIM is a library of routines for use with C or C++ programs, which enables to develop process-oriented, discrete-event simulation models. Active system elements (such as HTTP requests) are represented by CSIM processes while passive elements (such as Web server system resources) are represented by CSIM resources, i.e. facilities and storages with their own queuing systems. During a simulation run the processes compete for use of the system resources in a realistic way.

The architecture of our simulation tool is presented in Fig. 1. It includes a workload generator and an e-commerce Web server system simulator with a number of service modules.

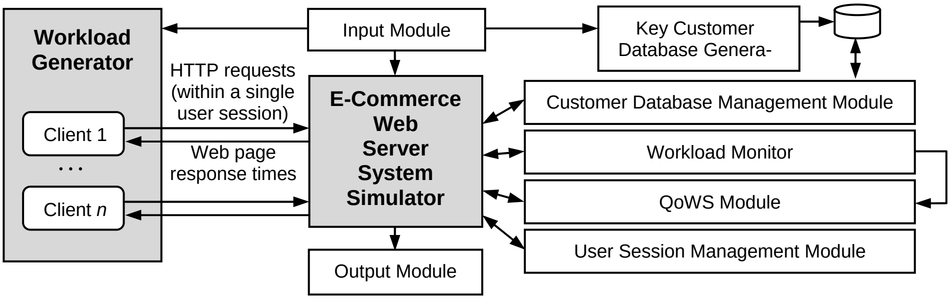


Fig. 1. Architecture of the simulation tool

The *workload generator* is an integral part of the simulator. It is responsible for generating and transmitting to the Web server system a sequence of HTTP requests emulating the session-based workload. The implemented workload model is discussed in [2] in detail. Based on parameters specified in an *input module*, the workload generator generates key customer and ordinary customer sessions at a given session arrival rate, i.e. it initializes a given number of new user sessions per minute.

After the initialization of each new session, the session description is generated in advance, including all HTTP requests in the session. The goal is to ensure exactly the same basic workload in all experiments for the same session arrival rate to objectively compare simulation results across different scheduling policies. The session requests are then sent to the system at given interarrival times. However, an actual system workload in each experiment depends on the Web server system performance: if page response times offered by the system are too long, the workload generator will stop sending requests within the corresponding sessions in order to simulate their failure.

The *e-commerce Web server system simulator* implements the queuing network model of the system (the model has been described in [2]). It simulates activities involved in request waiting in the system queues and their processing at the system resources. Other simulation services and auxiliary functions are implemented in software modules. An *input module* provides initialization and configuration parameters for the workload generator, a key customer database generator and the system simulator. The *key customer database generator* produces records characterizing key customer purchase histories and inserts them to the customer database at the beginning of each simulation experiment (this data matters in the case of some QoWS algorithms implemented in a QoWS module). A *customer database management module* is in charge of reading and updating customer records in the customer database during the simulation. A *workload monitor* collects statistics on the system current load and reports them to a *QoWS module*, which implements classification, admission control and scheduling algorithms. A *user session management module* implements user session semantics. An *output module* collects simulation statistics, computes performance metrics and summarizes them in a report at the end of the simulation.

4. PERFORMANCE METRICS

A key issue in evaluating Web server efficiency is defining the most appropriate performance metrics. Unlike most other QoWS approaches, we address the server system performance especially in terms of business-oriented metrics. Additionally, "conventional" system performance metrics are analyzed as well.

4.1. BUSINESS-ORIENTED METRICS

The business-oriented metrics include the following measures:

- *Revenue throughput* is defined as an amount of money per minute, generated through successfully completed buying sessions (c.f. [8]). The revenue is computed as the amount of money corresponding to the total value of goods purchased through the B2C site in a given observation window. Since we focus on the impact of computer system efficiency on the revenue, we do not take into consideration other factors affecting the actual retailer's profit (e.g. a fact that buyers may complain about products and demand refunds after some time).
- *Percentage of successfully completed key customer sessions* (in contrast to *percentage of aborted key customer sessions*) in the observation window. This metric is related to page response times offered by the system to key customers and it partially reflects their QoWS.
- *Percentage of achieved potential revenue* in the observation window. Potential revenue is computed as the total financial value of products in shopping carts of sessions which had ended with a purchase or had been aborted in the observation window. Achieved potential revenue is defined as the total financial value of products in shopping carts of the sessions ended with a purchase. Hence, the percentage of achieved potential revenue is defined as the percentage of potential revenue which turned into actual revenue. This metric gives the information on how effectively the system processed sessions with goods in shopping carts.
- *Potential revenue losses per minute* are computed based on the total financial value of products in shopping carts of sessions aborted due to poor QoWS[1].
- *B2V* (*Buy-to-Visit*) *ratio* relates the number of purchases to the number of all visits to the site in the observation window.

---

[1] The potential revenue losses metric does not include values of products which have been in shopping carts of customers who left the site due to unknown reasons, other than poor QoWS.

The revenue throughput provides information on the amount of dollars which have been achieved per minute. However, the revenue throughput is dependent on actual system load in each experiment. The workload generated in advance in various experiments for the same session arrival rate is the same, but actual Web traffic depends on the system performance (which in turn depends on a request scheduling policy) and thus, it may differ for different scheduling policies. For example, in the case of poor QoWS some sessions containing requests connected with adding a product to a shopping cart may be aborted at early stages; thus, they will not bring any item to a shopping cart and the resulting potential revenue will be lower. That is why the percentage of achieved potential revenue is introduced, giving the information on how effectively the system has processed all sessions with goods in shopping carts.

All the business-oriented metrics evaluate QoWS from the perspective of an online retailer and they reflect the system ability to support e-business profitability and key customers' loyalty.

4.2. SYSTEM PERFORMANCE METRICS

In this group one can distinguish two kinds of measures:

1) Metrics related to the system throughput and efficiency:
   - *The number of completed HTTP requests per minute*,
   - *The number of successfully completed sessions per minute*,
   - *Percentage of successfully completed sessions in the observation window*,
   - *Percentage of aborted sessions in the observation window*.

These metrics provide information on the system performance, which is particularly significant for the service provider and the Web site operator, who can see if the system capacity is enough to efficiently cope with the incoming Web traffic.

2) Metrics related to page response times:
   - *Mean page response time*,
   - *Median of page response time*,
   - *90-percentile of page response time*.

Commonly applied metrics are related to request response times. *Request response time* is defined as the time needed by the system to complete a single HTTP request. It encompasses an interval from receiving the first byte of the request until sending the last byte of a response. However, a metric much closer to user perception of latency is page response time, defined as the time needed by the system to complete a whole Web page, i.e. all HTTP requests for that page. *Page response time* t_p^s for the p-th page in session s is computed according to the following formula:

$$t_p^s = \sum_{x_{ip}^s \in O_p^s} t_i^s \quad (1)$$

where t_i^s is request response time provided by the system to the i-th HTTP request belonging to session s, x_{ip}^s is HTTP request i belonging to the p-th page in session s, and O_p^s is a set of Web objects making up the p-th page in session s. Page response time is computed only for successfully completed Web pages.

We apply various statistical measures to extrapolate a set of page response time values in the observation window. In QoWS studies the most common one is a mean value. However, it may not be a representative measure of a set of parameters having a heavy-tail distribution, i.e. when response times differ from one another with a few orders of magnitude. Thus, we apply the median and the 90-percentile of page response time as well. X-percentile of a random variable x is defined as $P(x \leq Y) = X$. For example, if 90-percentile of page response time is equal to 4 seconds, it means that in 90% of the system observations the page response time is less than 4 seconds.

The system performance metrics are computed for all monitored sessions and for key customer and ordinary customer sessions separately.

5. METHODOLOGY

In this Section, the methodology for carrying out simulation experiments using our simulation tool is discussed. We focus on the problem of registering the system behavior data during a single simulation. A new way of collecting statistics for session-based and revenue-based performance metrics in a simulation experiment is proposed.

Each simulation experiment is to emulate an operation of the e-commerce Web server system for some period of time. A single experiment is run for a given session arrival rate, i.e. the constant number of new user sessions initiated per minute.

In the typical case, when a Web server performance is evaluated on a request basis, an experiment run consists of two phases. The first one is a preliminary phase, lasting from the experiment start till the moment of the system behavior stabilization. The second one is a measurement phase, during which the system behavior is evaluated and some statistical data needed to compute performance metrics is collected.

However, in the case of applying session-based performance metrics, a problem with determining an observation window arises. Since we monitor user sessions from their beginning to end and compute per-session statistics, we have to properly determine bounds of the observation window and the way of collecting statistical data.

We propose distinguishing three phases of a single experiment run (Fig. 2).

1) The *preliminary phase* lasts from the simulation start at the moment $\tau_0 = 0$ till the moment τ_1, when the system operation is stable. In that phase no statistical data is collected.

2) The *measurement phase* lasts for a given period of time, delimited by moments τ_1 and τ_2. In that phase statistical data is collected only for user sessions initiated in that phase.
3) The *final phase* lasts from the moment τ_2 till the moment τ_{end}, when the last session started in the measurement phase has just finished. Only data for these sessions is collected, while sessions started in the final phase are processed without registering their service parameters.

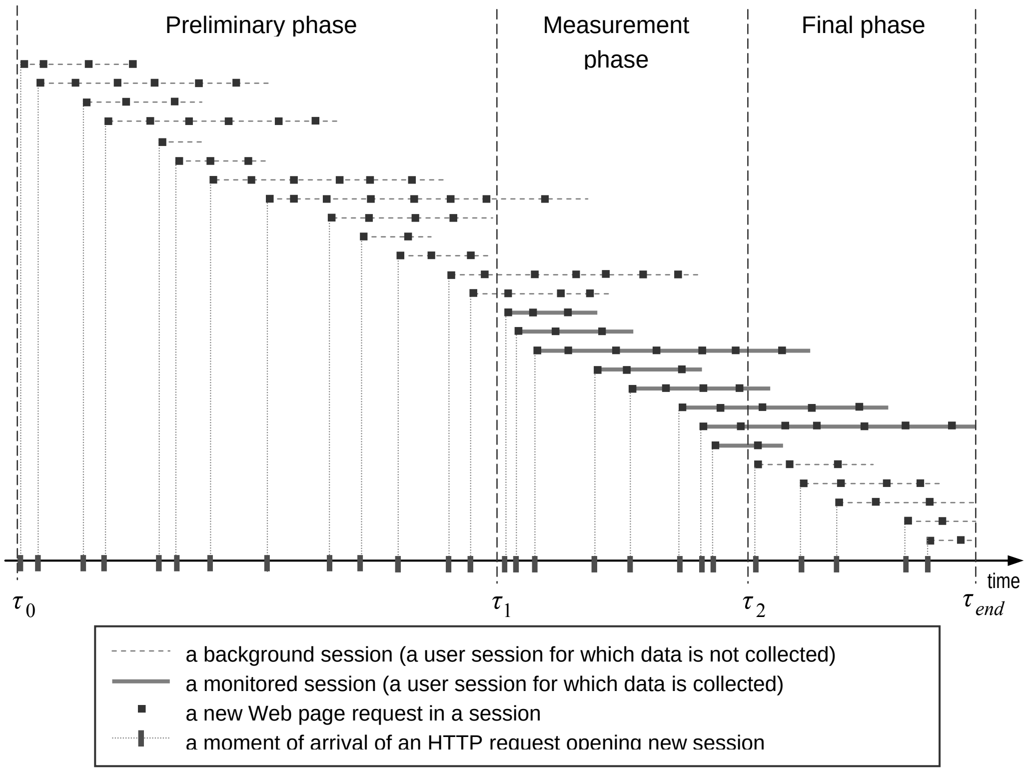


Fig. 2. Phases of a single simulation experiment (run for a constant session arrival rate)

It has to be noticed that the current time in a simulation is determined by the value of a simulation clock maintained by CSIM. Thus, the simulation time is different from the "real world" time and from the CPU time used in executing the simulation. Simulation time starts at zero and then advances along with the simulation proceedings according to program commands [9]. Thus, moments τ_0, τ_1, τ_2 and τ_{end} concern just CSIM simulation time, where τ_0 is equal to zero. Durations of the preliminary and the measurement phases amount to $\tau_1 - \tau_0$ and $\tau_2 - \tau_1$, respectively, and are given by input parameters. The final phase duration is not known in advance, because it is changeable and depends on the system efficiency and the incoming Web traffic during the simulation. We assume that the observation window is equal to the measurement phase, so its duration is equal to $\tau_2 - \tau_1$. This interval matters in the case of computing mean values of some performance measures.

Statistical data collected for the monitored sessions during the simulation is then used to compute system performance metrics (which have been discussed in Sect. 4). Since the generated workload is identical in all experiments for the same session arrival rate, such an approach enables us to compare system performance results across various scheduling policies.

A group of simulation experiments performed for the same workload and system parameters (i.e. the same scheduling policy) but for different session arrival rates, makes up a series. In consecutive experiments within a series, the session arrival rate is gradually increased. Values of a given performance metric for the whole series may be visually presented as a curve on a graph and depict a variation of the performance metric as a function of the session arrival rate.

As an example, lets consider the revenue throughput as a function of the session arrival rate varying from 20 to 300 sessions per minute (Fig. 3). The workload generated in experiments contained 10% of key customer sessions. The maximum page response time, after which a user session was considered aborted, was equal to 8 seconds. As it can be seen in Fig. 3, for lower system loads the revenue throughput grows with the increasing number of users interacting with the B2C site. However, above the point of 100 new sessions per minute the revenue rate decreases with the increase in the load. Such situation indicates the system overload, which results in long page response times (exceeding 8 seconds) and thereby in a greater number of aborted sessions (also sessions with goods in shopping carts, which could not be purchased).

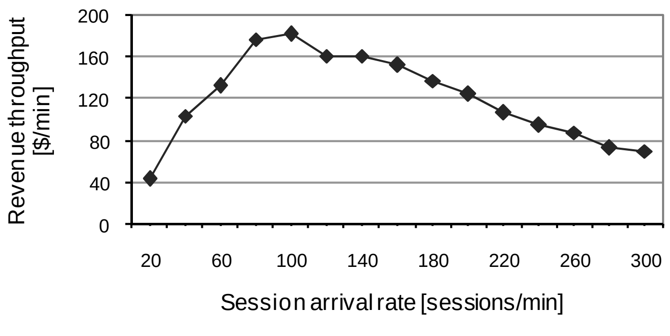


Fig. 3. Revenue throughput for FIFO scheduling (10% of key customer sessions in the workload)

A discussion on simulation results for a B2C Web server system performance under FIFO and priority-based scheduling using the tool and methodology presented in this work may be found in [2] and [12].

6. CONCLUDING REMARKS

The analysis of pros and cons of various approaches used to evaluate Web server performance motivated us to apply the simulation-based approach. We designed and implemented the simulator dedicated to the B2C environment and proposed a research methodology.

Our goal was to create a simulation tool, which allows one to carry out performance study of a B2C Web server system and provides information relevant to e-business profitability. Running such a performance study requires assuming a way of registering statistical data with regard to individual user sessions, as well as a way of deriving metrics oriented to user sessions and revenue in the whole observation window. We believe that the proposed methodology partially fills the gap in this area.

REFERENCES

[1] BHULAI S., SIVASUBRAMANIAN S., MEI van der R., STEEN van M., *Modeling and Predicting End-to-End Response Times in Multi-tier Internet Applications*, LNCS, Vol. 4516, 2007, 519–532.

[2] BORZEMSKI L., SUCHACKA G., *Business-Oriented Admission Control and Request Scheduling for E-Commerce Websites*, Cybernetics and Systems, Vol. 41, No. 8, 2010, 592–609.

[3] BORZEMSKI L., ZATWARNICKI K., *Fuzzy-Neural Web Switch Supporting Differentiated Service*, LNCS, Vol. 4252, 2006, 195–203.

[4] *CSIM19, Development Toolkit for Simulation and Modeling*, http://www.mesquite.com.

[5] DINKER D., SCHWETMAN H., *Using Java Method Traces to Automatically Characterize and Model J2EE Server Applications*, In: Winter Simulation Conference, Monterey, 2006, 1187–1194.

[6] GARCÍA D. F., GARCÍA J., *TPC-W E-Commerce Benchmark Evaluation*, IEEE Computer, Vol. 36, No. 2, 2003, 42–48.

[7] MATHUR V., APTE V., *An Overhead and Resource Contention Aware Analytical Model for Overloaded Web Servers*, The Journal of Systems & Software, Vol. 82, No. 1, 2009, 39–55.

[8] MENASCÉ D. A., ALMEIDA V. A. F., FONSECA R., MENDES M. A., *Business-Oriented Resource Management Policies for E-Commerce Servers*, Performance Evaluation, Vol. 42, No. 2-3, 2000, 223–239.

[9] Mesquite Software, Inc., *User's Guide: CSIM19 Simulation Engine (C++ Version)*, http://www.mesquite.com/documentation/documents/UsersGuide-CplusplusVersion.pdf.

[10] SCHROEDER B., HARCHOL-BALTER M., *Web Servers under Overload: How Scheduling Can Help*, ACM Transactions on Internet Technology (TOIT), Vol. 6, No. 1, 2006, 20–52.

[11] SHAN Z., LIN C., WEI Y., *Prototype Implementation and Performance Evaluation of a QoS-based Web Server*, In: Proc. of IEEE SOSE'05, Beijing, China, 2005, 191–196.

[12] SUCHACKA G., BORZEMSKI L., *Integrating a key customer-oriented strategy into the B2C e-commerce service*, MPER, Vol. 1, No. 4, 2010, 67–76.
[13] *The Apache Software Foundation*, http://www.apache.org.
[14] *TPC-W Java Implementation*, PHARM research team, University of Wisconsin-Madison, http://mitglied.multimania.de/jankiefer/tpcw/index.html.
[15] *TPC-W-NYU - A J2EE Implementation of the TPC-W Benchmark*, New York University, http://cs1.cs.nyu.edu/totok/professional/software/tpcw/tpcw.html.

Web performance forecast, spatio-temporal data mining, geostatistics, Sequential Gaussian Simulation

Leszek BORZEMSKI*, Anna KAMIŃSKA-CHUCHMAŁA*

SPATIAL WEB PERFORMANCE FORECASTING WITH SEQUENTIAL GAUSSIAN SIMULATION METHOD

This research describes a novel proposal of the spatio-temporal Web performance forecasting with Sequential Gaussian Simulation method belonging to group of the geostatistical simulation methods. The database was created on the basis of Multiagent Internet Measurement System MWING. Investigations consider connections of the agent from Gliwice with many www servers located in Europe. Base contains the measurements, which were taken every day at the same time, at 06:00 a.m., 12:00 a.m. and 6:00 p.m. during the period of month June 2008. A preliminary analysis of measurement data was conducted. Subsequently, the structural analysis, which contains the description of used Gaussian anamorphosis and directional variogram approximated with the theoretical model, was performed. Both of these analyses are necessary to realization forecast by Sequential Gaussian Simulation method. Next, a spatial forecast of the total time of downloading data from Web servers with a one-week time advance was calculated. The analysis of server activity on a particular week day for a period of a few weeks in selected time intervals and considered forecasted errors was performed. Results of forecast were analyzed in detail, followed by the determination of subsequent research directions to improve Web performance forecasts.

1. INTRODUCTION

Analysis of network measurements plays a very important role in research Web performance. The results of such analysis could be helpful for example in monitoring networks traffics or capacity of servers. In this work we propose to use the Sequential Gaussian Simulation (SGS) method to predict Web performance that is perceived by end users while downloading resources from Web servers. A great advantage of this

* Institute of Informatics, Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeże Wyspiańskiego 27, Poland, email: {leszek.borzemski, anna.kaminska-chuchmala}@pwr.wroc.pl

method is the possibility to make spatio-temporal forecasting, in which the minimum amount of input information is required, and at the same time it takes into account the geographical location of Web servers and the total download time of a given resource. Such forecast information is required e.g. when one has to download information resource which is available on many Internet nodes at various geographical locations.

2. SEQUENTIAL GAUSSIAN SIMULATION METHOD

The first step in geostatistical simulation is modeling of the variable processes, and next the simulation of these variables using an elementary grid. The foundation of sequential conditional simulation is Bayes' theorem and Monte Carlo (stochastic) simulation [13]. The simulation is said to be conditional if the realizations honor the data at their *n* sample locations:

$$z^{\ell}(u_\alpha) = z(u_\alpha), \forall \ell, \alpha = 1, \ldots, n. \tag{1}$$

A random function is a set of random variables over the study area. Let $\{Z(u'_j), j = 1, \ldots, N\}$ be a set of random variables at the *N* grid cells with location u'_j. The objective of a sequential conditional simulation is to generate several joint realizations of the *N* random variables: $\{z^{\ell}(u'_j), j = 1, \ldots N\}, \ell = 1, \ldots, L$, conditional to the data set $\{z^{\ell}(u_\alpha), \alpha = 1, \ldots, n\}$. This gives an *N*-point conditional cumulative distribution function that models the joint uncertainty at *N* locations u'_j:

$$F(u'_1, \ldots, u'_N; z_1, \ldots, z_N \mid (n)) = P\{Z(u'_1) \le z_1, \ldots, Z(u'_{\mathbf{N}}) \le z_N \mid (n)\}. \tag{2}$$

By recursively applying Bayes' theorem, an *N*-point conditional cumulative distribution function is formulated as a product of *N* one-point conditional cumulative distribution functions:

$$\begin{aligned} &F(u'_1, \ldots, u'_N; z_1, \ldots, z_N \mid (n)) = \\ &F(u'_N; z_N \mid (n + N - 1)) \times F(u'_{N-1}; z_{N-1} \mid (n + N - 2)) \times \cdots \times F(u'_1; z_1 \mid (n)). \end{aligned} \tag{3}$$

This decomposition allows a realization to be made in N successive or sequential steps. At the first location $u_1^{'}$, the cumulative distribution function is modeled conditionally to the n original sample data:

$$F\left(u_1^{'}; z \mid (n)\right) = P\left\{Z\left(u_1^{'}\right) \leq z \mid (n)\right\}. \tag{4}$$

Draw from the conditional cumulative distribution function an estimate, $z^1(u_1)$, which becomes part of the conditioning data for all subsequent drawings for the first realization. This process is repeated until all of the N nodes have a simulated value. This is one realization. A second realization would start with the original conditioning data and visit the N nodes in a different sequence.

The sequential simulation algorithm requires the determination of a conditional cumulative distribution function at each simulated node. In practice, the conditional cumulative distribution function is determined by the mean and variance from either simple kriging or ordinary kriging using either a multi-Gaussian or an indicator random function.

Algorithm of the Sequential Gaussian Simulation [12]:

1. Define a random path through all of the nodes to be simulated that is its each node exactly once.
2. Use simple kriging or ordinary kriging to determine the mean and variance of the Gaussian conditional cumulative distribution function at a node. Retain a specified number of neighboring data to be used as conditioning data. Both previously simulated node values and original data are included.
3. Draw randomly from the conditional cumulative distribution function and assign that value to the node being simulated.
4. Repeat steps 2 and 3 for all simulation locations.
5. Back transform the simulated normal values into the values for the original attribute.
6. Repeat steps 1–5 for multiple realizations.

Across the L realizations, the means, variances, frequencies, etc., can be spatially assessed. The appropriateness of the Gaussian distribution must be tested before simulation, often calling for a prior transformation of original data into a new data set with a standard normal cumulative distribution function. The simulated normal score values need to be transformed back to the simulated values for the original variable. SGS methods and their applications are described in details in [1, 6, 7, 8, 14].

3. PRELIMINARY DATA ANALYSIS

The database was created on the basis of active measurements made by MWING system which is Internet measurement infrastructure developed in our Institute [2, 3]. MWING has been developed to design and perform both kinds of experiments in a controlled way. Agents were installed on local hosts of networks belonging to academic campuses in four geographical locations: in Poland in Wrocław, Gliwice, Gdańsk, and in USA in Las Vegas. To create the discussed database, measurement data obtained from the agent located in Gliwice were used. The measured parameters referred to downloading a copy of a text document from many www servers located among others in Europe.

The measurements encompassed the period between 1st and 30th June 2008 and they were taken every day at the same time, at 06:00 a.m., 12:00 a.m., and 6:00 p.m. The input database necessary for calculations contains the information about server (node) geographical location with which the Gliwice agent were connected, the total downloading time and the time of taking the measurement.

Table 1. Elementary Statistical Parameters of Server Performance on the Internet within period 1–30.06.2008, at 6:00 a.m., 12:00 a.m., and 6:00 p.m.

Statistical parameters	Minimum value X_{min} (s)	Maximum value X_{max} (s)	Average value X (s)	Standard deviation S (s)	Variability coefficient V (%)	Skewness coefficient G	Kurtosis coefficient K
6:00 a.m.	0.08	7.20	1.00	1.19	119.00	2.39	7.85
12:00 a.m.	0.07	4.93	1.10	1.21	110.00	2.04	5.75
6:00 p.m.	0.07	9.06	1.12	1.35	120.54	2.44	9.35

Elementary statistics of Web performance for the considered servers are presented in table 1. Taking into account the minimum and maximum values, a rather large data range is observed. Only for data measured at 12:00 a.m. this difference is smaller. Moreover the high value of standard deviation and the coefficient of variation, which is under 100% for each considered hours confirms the process variation. However, the skewness and kurtosis coefficients values indicate that the distribution of the considered Web performances should show similarity to a symmetrical distribution but with only small right side asymmetry.

Location of www servers in analyzed area (Europe) is presented in figure 1. It is a base map where the size of the crosses corresponds to the download time from a given Web server. As could be seen in figure 1, there are two servers with very high time of download, one in Sweden and second in France. Additionally this map is characterized the large span of servers in analyzed area.

A histogram of loads distribution on the nodes of a computer network, presented in figure 2, is slightly asymmetric, single-wing and positively skewed. Modal classes 0.10 s–0.60 s and 0.60 s–1.05 s clearly dominate.

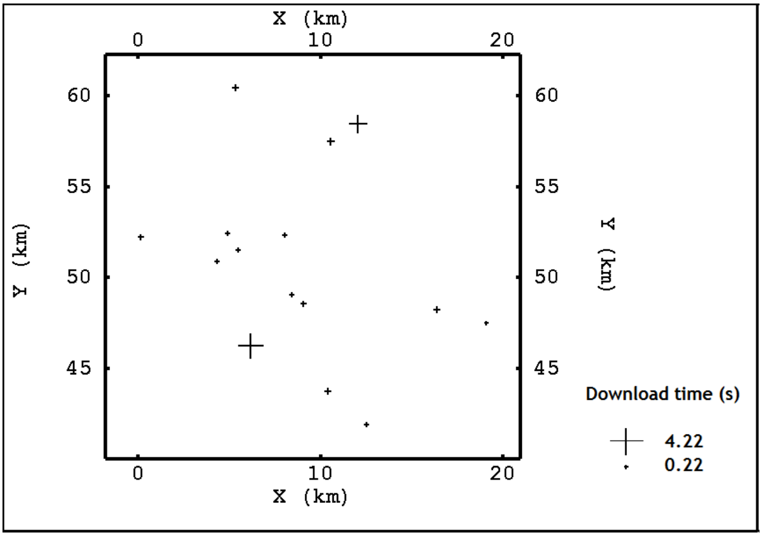


Fig. 1. Base map of download time values from the www servers

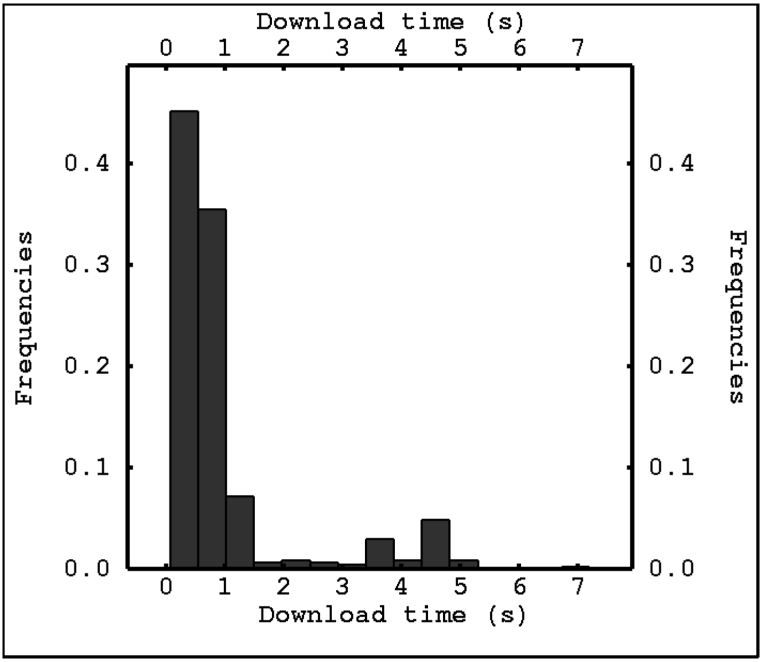


Fig. 2. Histogram of load distribution on Internet nodes in June at 6:00 a.m.

3. STRUCTURAL ANALYSIS OF DATA

3.1. GAUSSIAN ANAMORPHOSIS

During the calculations of Gaussian transformation frequency inversion model was used. The number of adopted Hermite polynomials in anamorphosis was equal to 100. Figure 3 presents the course of Gaussian anamorphosis. Due to limited number of pages exemplary plot calculating for 6:00 p.m. is presented.

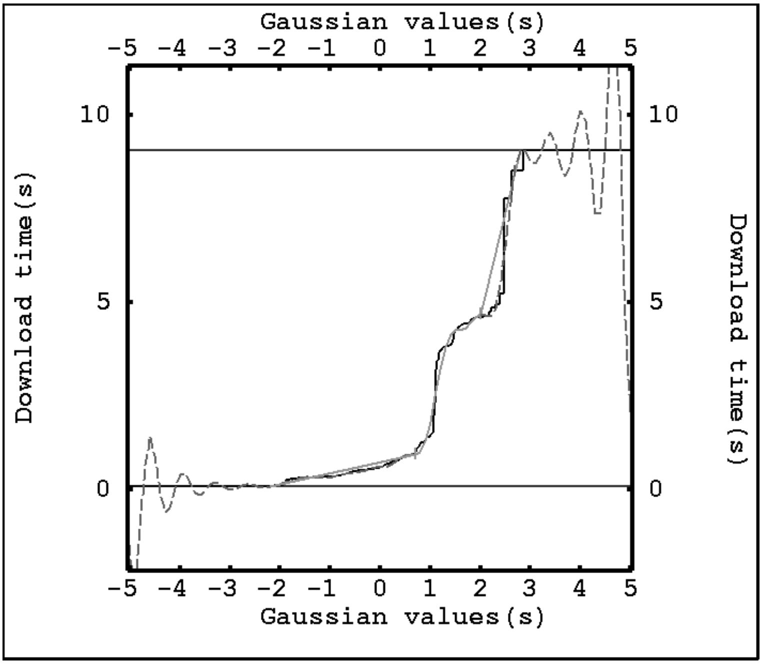


Fig. 3. Calculations of anamorphosis function used in the forecast of loads on the nodes measured at 6:00 p.m. using SGS method: thickened line – experimental curve, thin line – polynomial expansion, intermittent line – final anamorphosis function

The observed gentle interval in range [0.70; 2.00] which was obtained for Gaussian values proves good match between the adopted theoretical model and the empirical anamorphosis function of loads distribution (fig. 3).

3.1. VARIOGRAM MODEL

The next step in the preliminary data analysis and Gaussian anamorphosis calculations in preparation for forecasting, is modeling a theoretical variogram function. During the variogram model approximation, the nuggets effect function was used to consider Web performance at 12:00 a.m. A directional variogram was calculated along the time axis (for 90º direction). The distance class for this variogram was 1.00 km. Figure 4 presents a directional variogram approximated by the theoretical model of the nuggets effect and J Bessel. The variogram function indicates a gentle rising trend.

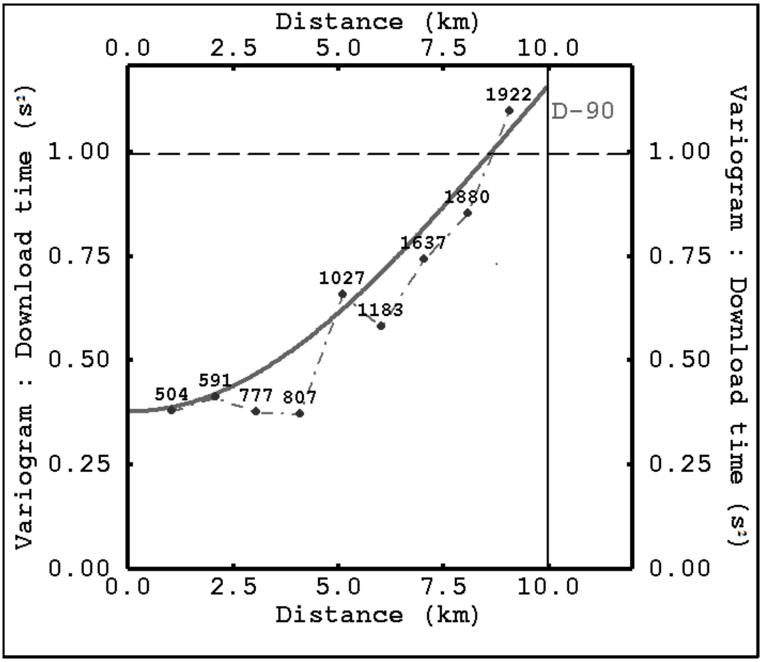


Fig. 4. Directional variogram along the time axis for Web performance on Internet nodes at 12:00 a.m. in June 2008, approximated by the theoretical model of the nuggets effect and J Bessel

The next two directional variograms of Web performance measured at 6:00 a.m. and 6:00 p.m. were approximated by the theoretical model of the nuggets effect and J Bessel too.

4. SPATIAL WEB PERFORMANCE FORECASTING WITH SEQUENTIAL GAUSSIAN SIMULATION METHOD

The forecasting models used to predict the total time of resource download from the Internet was the above discussed variogram models and Gaussian anamorphosis depending on the forecasted hour in June. In the simulation, the moving neighbourhood type was adopted where the search ellipsoid was 10 km for X and Y directions and 18 km for Z direction in the case of Web performance at 6:00 a.m., 12:00 a.m., and 6:00 p.m. The forecast of the download time was determined on the basis of 100 simulation realizations. In the simulation the punctual type was used. 3D forecast was calculated with a one-week time advance, i.e. it encompassed the period between 1st and 7th July 2008. The tables below present global statistics of the average forecasted values of node performance in a computer network for a period of 1 week, where data for 6:00 a.m., 12:00 a.m. and 6:00 p.m., respectively are presented.

Table 2. Global statistics of the forecasted values of Web server performance on the Internet with a one-week time advance, calculated using the Sequential Gaussian Simulation method, for 6:00 a.m., 12:00 a.m., and 6:00 p.m.

Geostatistical parameter	Mean value X (s)	Maximum value X_{max} (s)	Minimum value X_{min} (s)	Variance S^2 (s)^2	Standard deviation S (s)	Coefficient of variation V (%)
Mean forecasted value Z_s for 6:00 a.m.	1.09	5.75	0.10	0.69	0.83	76.15
Mean forecasted value Z_s for 12:00 a.m.	1.38	4.78	0.18	0.59	0.77	55.80
Mean forecasted value Z_s for 6:00 p.m.	1.40	6.45	0.13	0.69	0.83	59.29

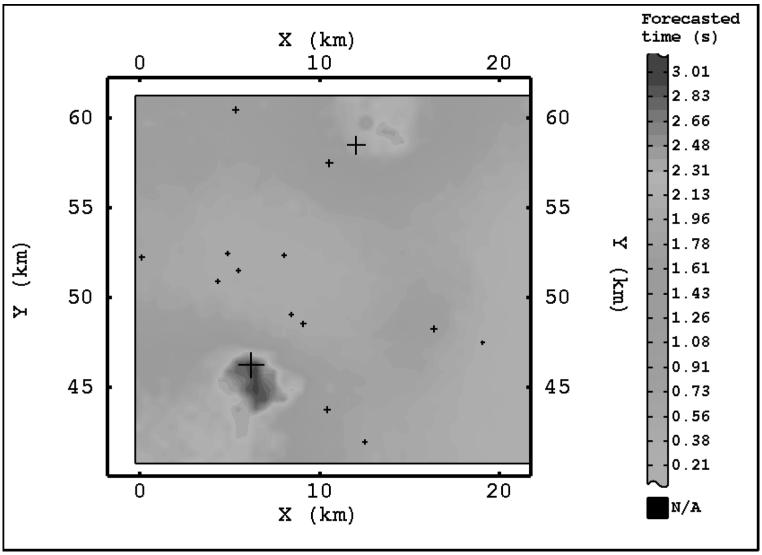


Fig. 5. Exemplary raster map of download time values from the Internet on 01.07.2008 at 6:00 a.m.

On the basis of the forecasts of Web server performance presented in table 2, very interesting phenomena can be observed. Namely, the forecasts for 6.00 a.m. have the biggest data range taking into account the disparity between the minimum value and the maximum one, and the standard deviation which is the highest for this time in comparison with the mean value. Moreover, for 12:00 a.m. coefficient of variation for all Web servers was the smallest.

The final effect of the forecast calculations is presented in figure 5. It is a raster map for the 1st day of prognosis (1–7.07.2008), it presents the download time from the Internet. The size of the cross corresponds to the download time from a given Web server (the download time is given in seconds). Geostatistics methods could give information about performance not only in considered nodes, but for whole considered area. In this map, two servers with large time of download (in Sweden and France) are characteristic.

5. CONCLUSIONS

The novel approach Sequential Gaussian Simulation geostatistics method in spatial forecasting of download time, thanks to which we can find out where there are Internet overloads, seems to be justified. A realistic possibility of using geostatic methods in yet another, new discipline which is computer science is outlined.

Additionally there is a need to work on the improvement of forecast accuracy. Internet loads is analyzed also in other areas, using various measurement data and prediction lengths [4, 5]. However, the next step should be an attempt to use other geostatistics methods, which have already been successfully used by the author to forecast loads in power transmission and distribution networks [9, 10, 11].

REFERENCES

[1] BOHLING G., *Stochastic simulation and reservoir modeling workflow*, C&PE 940, 21.10. 2005.

[2] BORZEMSKI L., CICHOCKI L., KLIBER M., FRAS M., NOWAK Z., *MWING: A Multiagent System for Web Site Measurements,* In: Nguyen N.T., Grzech A., Howlett R.J., Jain L.C. (eds.) KES-AMSTA 2007. LNCS (LNAI), Vol. 4496, Springer, Heidelberg 2007, pp. 278–287.

[3] BORZEMSKI L., CICHOCKI L., KLIBER M., *Architecture of Multiagent Internet Measurement System MWING Release 2,* In: Hakansson, A., Nguyen N.T., Hartung R.L., Howlett R.J., Jain L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, Springer, Heidelberg 2009, pp. 410–419.

[4] BORZEMSKI L., KAMIŃSKA-CHUCHMAŁA A., *3D Web Performance Forecasting Using Turning Bands Method*, Communications in Computer and Information Science, Volume 160, Springer-Verlag Berlin, 2011, pp. 102–113.

[5] BORZEMSKI L., KAMIŃSKA-CHUCHMAŁA A., *Knowledge Discovery about Web Performance with Geostatistical Turning Bands Method*, A. König et al. (Eds.): KES 2011, Part II, LNAI 6882, Springer, Heidelberg 2011, pp. 581–590.

[6] GERTNER G., *Comparison of Computationally Intensive Spatial Statistical Methods for Generating Inputs for Spatially Explicit Error Budgets*, IUFRO 4.11 Conference, Greenwich 2001.

[7] HANSEN T.M., JOURNEL A.G., TARANTOLA A., MOSEGAARD K., *Linear Inverse Gaussian theory and geostatistics*, Geophysics, Vol. 71, No. 6, November–December 2006.
[8] HICKS P.J., *Unconditional sequential Gaussian simulation for 3-D flow in a heterogeneous core*, Journal of Petroleum Science and Engineering 16, 1996, pp. 209–219.
[9] KAMIŃSKA-CHUCHMAŁA A., WILCZYŃSKI A., *Przestrzenne prognozowanie obciążeń elektrycznych w sieci NN metodą Sekwencyjnej Symulacji Gaussa*, Rynek Energii, nr 1(92), (2011), pp. 35–40.
[10] KAMIŃSKA-CHUCHMAŁA A., WILCZYŃSKI A., *Prognozowanie 3D obciążeń elektrycznych z wykorzystaniem symulacyjnej metody geostatystycznej Turning Bands*, Prace Wrocławskiego Towarzystwa Naukowego, tom XVI, seria B nr 215, Wrocław 2009, pp. 41–48.
[11] KAMIŃSKA-CHUCHMAŁA A., WILCZYŃSKI A., *Prognozowanie 3D obciążeń elektrycznych z wykorzystaniem symulacyjnej metody geostatystycznej Turning Bands*, Prace Wrocławskiego Towarzystwa Naukowego, tom XVI, seria B nr 215, Wrocław 2009, pp. 41–48.
[12] KING S.L., *Sequential Gaussian simulation vs. simulated annealing for locating pockets of high-value commercial trees in Pennsylvania*, Annals of Operations Research 95, 2000, pp. 177–203.
[13] TARANTOLA A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, SCIAM Society for Industrial and Applied Mathematics, Philadelphia, 2005.
[14] WANG G., GERTNER G.Z., PARYSOW P., ANDERSON A.B., *Spatial prediction and uncertainty analysis of topographical factors for the Revised Universal Soil Loss Equation (RUSLE)*, Journal of Soil and Water Conservation. 3rd Quarter, 2000.

Autonomous Systems,
computer network monitoring

Grzegorz KOTOWSKI*, Dariusz GĄSIOR*

POPULARITY FACTOR BASED AUTONOMOUS SYSTEMS MONITORING INTENSITY

In present days access to information located in Web sites hosted on Web servers, located all over the world is very crucial. Web resources, of course, may vary in sizes, what may lead the obtaining time to vary and the differences may be slightly significant. Therefore, the prediction algorithm [5] should be applied in order to appropriate choice of a source. Unfortunately, even a constant observation of the Internet or its simplified, aggregated version [4] may not bring the optimal solution, as the evenly monitored Autonomous Systems may lead to more erroneous prediction of obtaining time, when the requests for the certain resources are more frequent. The solution to this problem is presented in this chapter.

1. INTRODUCTION

1.1. AUTONOMOUS SYSTEMS

As the Internet consists of billions of hosts, to simplify the management, it has been divided into Autonomous Systems[1], every one of which having a registered number (Autonomous System Number[2]), and containing a number of hosts. The first guidelines for ASes were presented in [1] and introduced 16-bit numbering of ASes (providing a range from 0 to 65,535). In 2007, a new (32-bit) numbering system emerged due to an incoming lack of unassigned ASNs [6]. Nowadays, there are more

---

* Institute of Informatics, Wroclaw University of Technology, Wybrzeże Wyspianskiego 27, 50-370 Wroclaw, e-mail {grzegorz.kotowski, dariusz.gasior}@pwr.wroc.pl

[1] Further in the text also: *AS (plural: ASes).*

[2] Further in the text also: *ASN (plural: ASNs).*

than 70,000 registered ASes, and about 50,000 are active. The rest is usually reserved by IANA (Internet Assigned Numbers Authority) and Regional Internet Registries (RIR) such as RIPE, AfriNIC, APNIC, ARIN, LACNIC for other purposes. The official division map is presented in Fig. 1.

Fig. 1. RIR coverage world map[3]

Even though the number of 50,000 is only the small percent of full 32-bit numbering range[4], the AS interconnection structure is complex enough (see Fig. 2) for monitoring, not mentioning that every AS carries even up to hundreds or thousands of Internet hosts.

1.2. AUTONOMOUS SYSTEMS MONITORING PURPOSE

Nowadays, the access to information is very crucial. As the Internet is considered as the omnipresent, public accessible source of any kind of information located in billions of Web pages, the access time is the most important perceived quality index of acquiring it from the regular user's point of view. To put it briefly: *the quicker the Web page loads, the better*. In order to decrease that time, copies of the Web resources are placed all over the Internet in so-called mirror servers, and a user may decide from which location to download a resource from. *So how should one make their choice?*

Naturally many source choice approaches exist. On one hand a user has a manual choice based on their individual intuition and/or experience; on the other there are

[3] Map based on a blank political world map from *wikipedia.org*.
[4] Up to 4,294,967,296.

automated ones that can be divided into simple and complex approaches. The methods of the first (automated) group may be based on: geographical distance, world region (e.g. continent, country, state, etc.) closeness, domain similarity, or IP route path distance, and the second on e.g. download time prediction [5].

Putting the manual and simple automated methods aside, let us take a closer look to the automated ones. The common part of all of the aforementioned approaches is the knowledge about the network. However, the simple methods are based on rather static knowledge that can be once gathered and then periodically updated, whereas the complex ones are based on knowledge about a dynamically changing structure, and have to be regularly updated. Such regular updates need constant monitoring and observation of its (the structure) components characteristics. In case of an AS, it may be: RTT, TCP connect time, HTTP response time or even their various combinations.

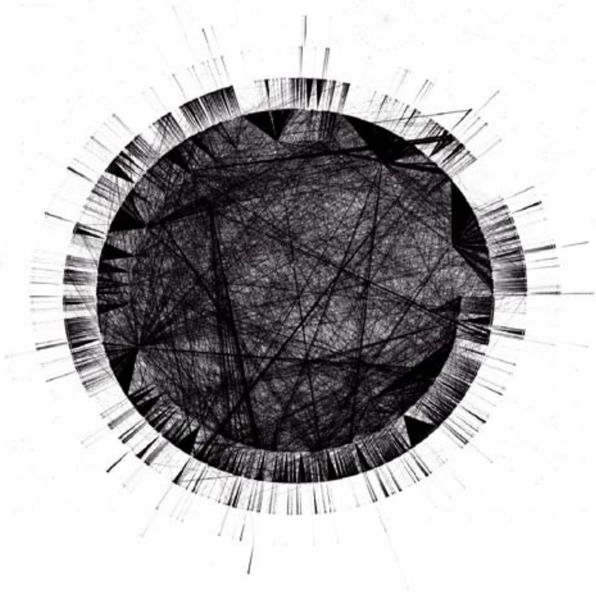


Fig. 2 Autonomous System interconnection map

Unfortunately, dealing with such a complicated structure (see Fig. 2) prevents one from real time Internet monitoring. What is more, it forces simplification of the structure so that the operation of data acquisition and analysis is possible.

Obvious is that to obtain the big picture of the Internet and real-time full characteristics of every AS, one needs to monitor all the hosts in every single AS. Unfortunately that is impossible because of computational power and network bandwidth limits. Therefore, the number of observed hosts has to be reduced. To achieve the reduction, one needs to perform at least two main operations. First, one has to choose single hosts belonging to the AS as its representatives. This will help in calculating the characteristics of an AS and its *behavior*. Nevertheless, even choosing a few representatives for an AS makes a number of few hundred thousands of hosts to observe in global. Second, one needs to reduce the number of monitored ASes by aggregation, based on representatives' characteristics similarity [4]. Not only does applying such a reduction slightly decrease the global[5] number of monitored hosts, but it also gives a compromising Internet structure, seen from a certain point of view.

2. AUTONOMOUS SYSTEMS MONITORING ISSUES

During the usage of the system based on the knowledge from evenly monitored ASes, one may notice some user trends, showing their interests in some Web resources, leading to more often downloads from certain ASes. As the system may only provide the source choice basing on predicted obtaining time, such a situation may take effect in more erroneous predictions. The solution to that is the application of the algorithm concerning the end-user preferences, which imply the popularity changes of certain Web resources and as a result of this the ASes which the hosts containing the resources are located in.

3. WORST CASE AUTONOMOUS SYSTEMS MONITORING INTENSITY ADJUSTING ALGORITHM

3.1. NOTATION

Let:

t_{ij} – download time (real) of an i^{th} resource from the j^{th} location,

$\overline{t_{ij}}$ – estimated download time of an i^{th} resource from the j^{th} location, and $\overline{t_{ij}} = \varphi_{ij}\left(\overline{\tau}_j\right)$,

d_{ij} – uncertainty of download time $\overline{t_{ij}}$,

φ_{ij} – predictor of the download time of an i^{th} resource from the j^{th} location (prediction function),

[5] In terms of the discussed system

$\overline{\tau}_j$ – estimated value of the τ_j characteristics of j^{th} location (e.g. RTT) based on measurements, where τ_{jn} is n^{th} measurement of j^{th} location,

δ_j – uncertainty of measuring τ_j characteristics (the more measurements, the lower),

$j \in \{1,\ldots,J\}$ – location index (assumed to be AS or Meta-AS),

ω_i – popularity (intensity of requests) of an i^{th} resource,

f_j – intensity of observation of an j^{th} AS (j^{th} location),

$\overline{F}$ – maximal number of observations (measurements) that is possible to be performed in a time unit concerning limited resources (network and computing) dedicated for measurement,

c_j – "cost" of an observation of an j^{th} location,

$i \in \{1,\ldots,I\}$ – resource index.

3.2. PROBLEM STATEMENT

As both $\overline{t_{ij}} = \varphi_{ij}(\overline{\tau}_j)$ and the uncertainty δ_j depend on the number of measurements (equivalent to measurement intensity), the d_{ij} uncertainty also depends on the measurement intensity (f_j), what will denoted $d_{ij}(f_j)$.

Let $j^*(i) = \arg\min_j \overline{t_{ij}}$ be the host, which is the i^{th} resource downloaded from (has the shortest predicted obtaining time). Let $\hat{j}(i) = \arg\min_j t_{ij}$ be the host, which should be chosen (has the shortest real obtaining time).

Quality index is given by the following formula:

$$Q(f) = \sum_i \omega_i (t_{ij^*(i)} - t_{i\hat{j}(i)}) + \sum_j c_j f_j \,, \tag{1}$$

where: $f = [f_1,\ldots,f_j,\ldots,f_J]^T$.

It is worth to notice that values of t_{ij} are not known a priori, but their predicted values $\overline{t_{ij}}$ provide a prediction, the lower the uncertainty is, the better. To put it clearly: *the more intensive observation, the better prediction to be achieved*. Therefore the higher values of f_j are selected, the lower the first element of the quality index, i.e. $\sum_i \omega_i (t_{ij^*(i)} - t_{i\hat{j}(i)})$, is. Assuming that:

$$t_{ij} \in [\overline{t_{ij}} - d_{ij}(f_j), \overline{t_{ij}} + d_{ij}(f_j)] \tag{2}$$

the index value may be limited by $\overline{Q}(f) \geq Q(f)$, where:

$$\overline{Q}(f) = \sum_i \omega_i (g_{ij}(f_j)) + \sum_j c_j f_j \,, \tag{3}$$

where:

$$g_{ij}(f_j) = t_{ij^*(i)} + d_{ij^*(i)}(f_j) - t_{i\bar{j}(i)} + d_{i\bar{j}(i)}(f_j)\,, \tag{4}$$

and

$$\bar{j}(i) = \arg\min_j \overline{t_{ij}} - d_{ij}(f_j) \tag{5}$$

1) Of course it is worth to notice that $\overline{t_{ij}}$ values are not known, until the observations are conducted, and the f_j value should be known before.

2) Nevertheless, noticeable is that the difference

$$t_{ij^*(i)} + d_{ij^*(i)}(f_j) - t_{i\bar{j}(i)} + d_{i\bar{j}(i)}(f_j)$$

is non-increasing with f_j increase (if (2) is true).

The problem may be formulated as follows:
Find:

$$f^* = \arg\min Q(f)\,,$$

such that:

$$\sum_j f_j \leq F$$

Unfortunately, according to the aforementioned reasons, the problem is unsolvable. Therefore, in this chapter, we propose to find such a selection of the measurement intensity that will minimize the upper limit on $Q(f)$, namely:

Find:

$$\hat{f} = \arg\min \overline{Q}(f),$$

such that:

$$\sum_j f_j \le F$$

In this chapter, we consider the case, where F is unlimited but increasing f implies greater cost (i.e. c_j multipliers are nonzero).

3.3. SOLUTION ALGORITHM

In order to find the minimum of function $\overline{Q}(f)$, we will use the gradient descent method:

$$f_j[n] = f_j[n-1] + k_j \left.\frac{\partial \overline{Q}(f)}{\partial f_i}\right|_{f_j = f_j[n-1]} \tag{6}$$

Unfortunately, because of the nondifferentiability of function $g_{ij}(f_j)$, the procedure (6) cannot be applied. Instead of that we will use the approximation:

$$\left.\frac{\partial \overline{Q}(f)}{\partial f_j}\right|_{f_j = f_j[n-1]} \cong \begin{cases} \overline{Q_n}(f) & \mathit{for} \quad f_j[n-1] \neq f_j[n-2] \\ 0 & \mathit{for} \quad f_j[n-1] = f_j[n-2] \end{cases} \tag{7}$$

where:

$$\overline{Q_n}(f) = \frac{\overline{Q}([f_1[n-2], \dots, f_j[n-1], \dots, f_J[n-2]]) - \overline{Q}(f[n-2])}{f_j[n-1] - f_j[n-2]} \tag{8}$$

which finally brings:

$$f_j[n] = f_j[n-1] - k_j\left[\overline{Q_n}(f)\right] \tag{9}$$

and the value of $g_{ij}(f_j)$ – being the element of $\overline{Q}(f_j)$ – is calculated basing on observation.

4. SIMULATION RESULTS

The simple simulation was performed in which two Web resources with different popularity were chosen. For each of these resources two independent localizations were found. As it turned out these sources were located in three separate ASes (both resources were present in one of the selected ASes). In the following case:

1. The resource R_1 is located in locations (ASes) 1 and 2.
2. The resource R_2 is located in locations (ASes) 2 and 3.

The observed data are presented in Table 1.

Table 1 Simulation results

n	f_1 [1/min]	f_2 [1/min]	f_3 [1/min]	Resource R_1 [min]		Resource R_2 [min]	
1	15	10	5	$\overline{t_{11}} = 2$	$d_{11} = 1$	$\overline{t_{22}} = 1{,}5$	$d_{22} = 1$
				$\overline{t_{12}} = 3$	$d_{12} = 2$	$\overline{t_{23}} = 2$	$d_{23} = 1{,}5$
2	10	5	15	$\overline{t_{11}} = 2{,}5$	$d_{11} = 1{,}5$	$\overline{t_{22}} = 1{,}5$	$d_{22} = 1{,}25$
				$\overline{t_{12}} = 3$	$d_{12} = 2{,}5$	$\overline{t_{23}} = 1{,}5$	$d_{23} = 1$

Additionally the popularity (intensity of requests) is $\omega_1 = 10\,{}^{req}\!/\!_{h}$ [6] and $\omega_2 = 1\,{}^{req}\!/\!_{h}$.

In this case we assume that the download time for a single resource is a multiple of average time of downloading the test chunk of data (the predicted obtaining time is proportionally greater than test chunk obtaining time). We also assume that the

[6] *req* – request.

uncertainty of the predicted download time is also a multiple of average test chunk downloading time, and the measurement uncertainty is a type A uncertainty given by the formula [2, 3]:

$$u(X) = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n(n-1)}} \tag{10}$$

Where $X_1, X_2, \ldots, X_i, \ldots, X_n$ are measurement results, $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, and n is the number of measurements.

Assuming coefficients $k_j = 0{,}5$ (for each j) and $c_1 = c_2 = c_3 = 1$, the intensity of observations should change in the following way:

$f_1[3] = f_1[2] + 3{,}5 = 13{,}5$

$f_2[3] = f_2[2] + 3 = 8$

$f_3[3] = f_3[2] - 5 = 10$

5. CONCLUSIONS AND FURTHER WORKS

The chapter included the introduction and description of the Internet monitoring purposes and the proposal of solution algorithm to the network[7] monitoring and observation problem in case of limited networking and computing power resources. The presented algorithm implies the way monitoring intensity modification according to the popularity of a single AS, and the Web resource located in it itself. The system is especially taking into account the case of the mirrored resources, relieving the end-user of making the choice of the appropriate source on their own.

Further works will concentrate on application of the designed mechanism to the automated system enabling the choice of the best source of the certain resource and also source monitoring and observation planning, concerning the resource popularity based on end-user preferences.

---

[7] In this case: the Internet.

REFERENCES

[1] HAWKINSON J., BATES T., *Guidelines for creation, selection, and registration of an Autonomous System (AS)*, March 1996, Request for Comments, 1930.

[2] Joint Committee for Guides in Metrology, *Guide to the Expression of Uncertainty in Measurement* Geneva, 1995.

[3] Joint Committee for Guides in Metrology, *Evaluation of measurement data – An introduction to the 'Guide to the expression of uncertainty in measurement' and related documents*. Geneva, 2009.

[4] KOTOWSKI G., GĄSIOR D., *Application of non-probabilistic descriptions of uncertainty for evaluation of approximate performance parameters in autonomous systems aggregation*, In: Information systems architecture and technology: advances in Web-Age Information Systems. eds Leszek Borzemski [et al.]. Wrocław: Oficyna Wydaw. PWr., 2009, 59–68.

[5] KOTOWSKI G., GĄSIOR D., *Adaptive fuzzy predictor for Web server resource obtaining time anticipation,* In: Information systems architecture and technology: New developments Web-Age Information Systems. eds Leszek Borzemski [et al.]. Wrocław: Oficyna Wydaw. PWr., 2010, 77–88.

[6] VOHRA Q., CHEN E., *BGP Support for Four-octet AS Number Space*, May 2007, Request for Comments: 4893.

performance algorithms
efficient resource acquisition

Wojciech SYPNIEWSKI*, Ziemowit NOWAK*

SEARCH ENGINE'S PROTOTYPE WITH "BEST PERFORMANCE HIT" MECHANISM

The publication describes the issue related to the efficient acquisition of resources on the Web basing mainly on the Internet model used in the algorithms best performance hit. The experimental system has been built that uses Web search engine performance algorithms to estimate the quality of resulting resources. Moreover, time of obtaining resources was carried out with comparative analysis of hybrid algorithms using the collected and compiled actual data. Implemented search engine allows searching for specific terms making the download time estimation of the corresponding resources on the basis of geographical location data and distance at the level of autonomous systems and also based on performance data gathered from group of servers deployed in random locations on the surface of the Earth.

1. INTRODUCTION

Service access to the global Internet WWW (*World Wide Web*) develops from the beginning in a very dynamic way, resulting in increasingly higher requirements of the newly formed and existing resources. One of the key factors shaping Internet traffic is the perception of resources by the end user (*user-perceived performance*). The components of perception in this case are defined as: high-performance, availability and security. Along with the ever faster links and servers with very high bandwidth and power, perceived waiting time by the user on the resource has become a major aspect of quality assessment. The performance and reliability of specific Web services is affected by various factors, e.g., used networking technologies, software solutions on the server side as well as infrastructure and thus use the cache option, CDN, the URL-IP translation, the performance and availability of DNS service.

---

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

Perceptions of the quality of the web resource are mainly due to the delay and bandwidth. The main cause of delays, seen by the end user level access to the web server via web browser, is the time required for the connection, i.e. the time between sending a request to receive the first bit of data. The lower the value of the delay, the faster it will download the appropriate resource. Another key element is the bandwidth between the agent and the server hosting the resource. Bandwidth is defined as a numeric value determining the number of bytes sent to the route network for a specified amount of time. Thus, network bandwidth determines the speed seen by the end user.

2. SOURCES OF PERFORMANCE LOSS AND EFFICIENT RESOURCE ACQUISITION IN WORLD WIDE WEB SERVICE

On the total time from dispatch of a request by the client to obtain the resource consists several different factors:

– data rate links between the parties,
– waiting for a response from the server,
– the answer (resource).

Intentionally omitted were the sizes of requests, which are negligible and have no effect on performance. The sources of performance degradation parameters are thus directly responsible for the transmission speed links, the time of expectations and stock size. On this basis, it can be concluded that the loss of productivity, or increase waiting time for a resource by the end user, is mainly caused by the quality of links, the type and quantity of intermediate routers, in combination, and the transmission control protocol packets. In short, the total time of your request for a resource to its receipt depends primarily on the size of the resource and the effective transmission speed of packets.

Acquisition of resources in an efficient manner is the main theme of work and decided on his subject. The basic parameters determining the quality of the resources were described above, so now we can move to the description of the algorithm used to create a prototype search engine applications. The algorithm is named *Best performance hit*, and was first described in [1]. The main assumption of the algorithm is to make possible the download time estimation of any resource without having to make additional measurements at the time. The work presents two different ways of estimation of time: on the basis of AS-level and geographic distance. Below are described two hybrid algorithms and I want to focus on them because the observation of the authors and creator of the algorithms shows that they are much more efficient than their two non-hybrid predecessors.

2.1. HYBRID BEST PERFORMANCE HIT ALGORITHMS

The first described algorithm will be the one which is determining transmission time based on the ratio λ, which was defined in this publication as the AVG algorithm. Another described is the second hybrid algorithm based on the ratio κ, which was defined in the work briefly as an algorithm MAX.

To understand the description of the algorithm is also necessary to determine the present indications:

$z^{(2)}$ – the size of the found resource,

$z^{(3)}$ – number (address) AS, where the non-monitored server is found, providing a resource,

$z^{(4)}$ – coordinates AS, in which the non-monitored server is found, providing a resource,

$\bar{y}$ – transmission time resource.

The general form of the AVG algorithm is presented in Figure 1. The principle of the Internet model is as follows: for each resource discovered by the search engine, the model passes the size of discovered resource $z^{(2)}$, the AS number, in which the non-monitored server is having the resource $z^{(3)}$, and $z^{(4)}$ which stores the coordinates of the AS. By convention, in order to simplify the writing of dependence the index n has been omitted. On the basis of given in $z^{(3)}$ AS number, and data from the knowledge base is determined *M*-element vector indices Γ. In turn, on the basis of the $z^{(4)}$ AS geographic coordinates and data from the knowledge base, determined *M*-element vector indices Η. As a result of aggregation of vectors Γ and Η the vector indices Κ is created. The next step is the transfer of the module is to give from module MAX to multiplexer, the index j of λ_j indicator such that $\lambda_j = \lambda^*$, where:

$$\lambda^* = \max_m\{\lambda_m\} \tag{1}$$

Then the multiplexer selects the vector of the element with index j, ie the bandwidth of the j-th server, and passes its value to the next module. This module from given relation $\bar{y} = \frac{z^{(2)}}{w_j}$ determines the transmission time and transmit it to present draft steering system.

The general form of the algorithm MAX is presented in Figure 2. The principle of the Internet model is as follows: for each resource discovered by the search engine, the model passes the size of discovered resource $z^{(2)}$, the AS number, in which the non-monitored server is having the resource $z^{(3)}$, and $z^{(4)}$ which stores the coordinates of the AS.. By convention, in order to simplify the writing of dependence the index n has

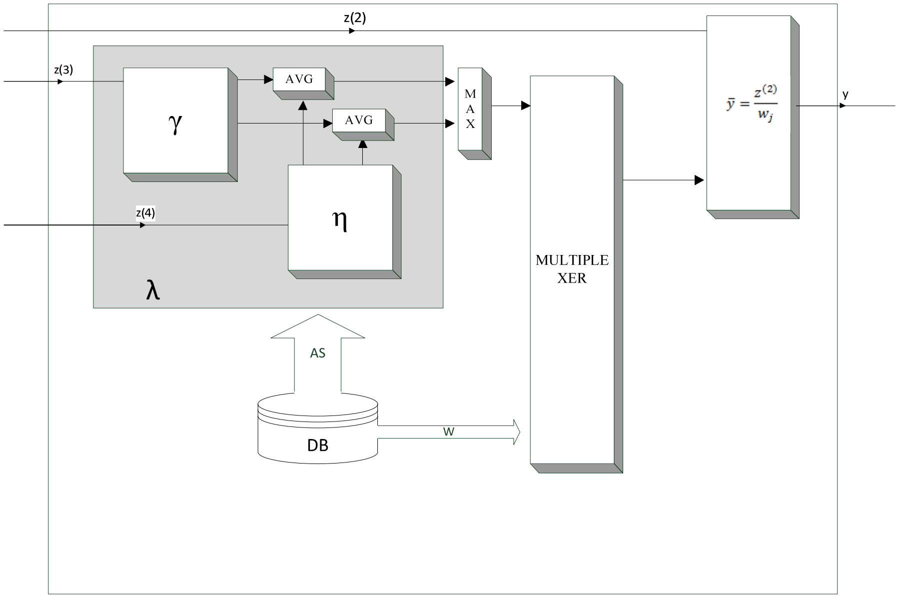


Fig. 1. Internet model build using AVG algorithm

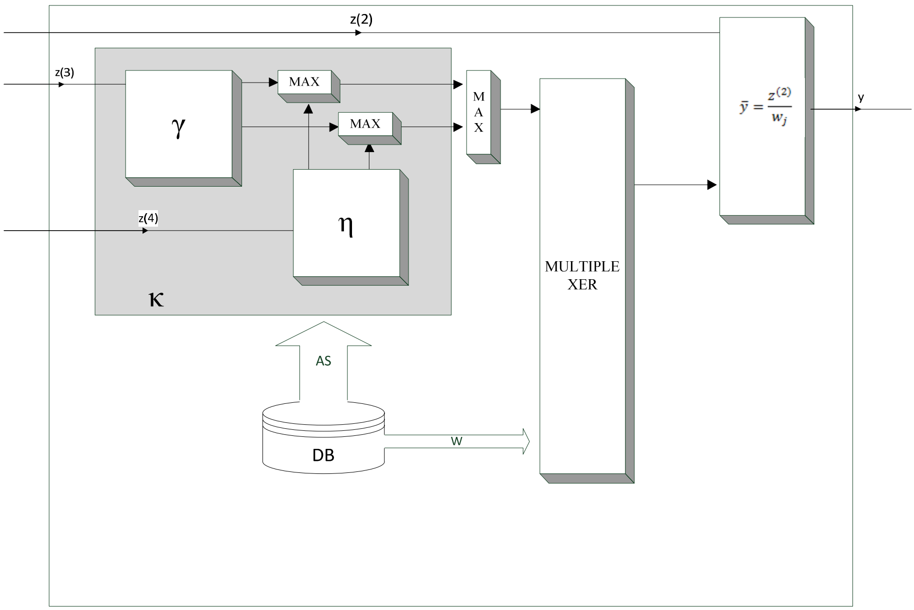


Fig. 2. Internet model build using MAX algorithm

been omitted. On the basis of given in $z^{(3)}$ AS number, and data from the knowledge base the M-element vector indices Γ is determined. In turn, on the basis of the $z^{(4)}$ AS geographic coordinates and data from the knowledge base, the M-element vector indices H is determined. As a result of aggregation of vectors Γ and H the K vector indices is created. The next step is the transfer from the module MAX to multiplexer the index j if indicator κ_j such that $\kappa_j = \kappa^*$, where:

$$\kappa^* = \max_m\{\kappa_m\} \tag{2}$$

Then the multiplexer selects the vector of the element with index j, ie the bandwidth of the j-th server, and passes its value to the next module. This module from given relation $\bar{y} = \frac{z^{(2)}}{w_j}$ determines the transmission time and transmit it to present draft steering system.

3. WEB SEARCH ENGINE PROJECT AND IMPLEMENTATION

Description of implementation and project implementation is made in accordance with accepted standards for requirements management. On this basis, the authors developed the stages of implementation. Creating of an application has been started by gathering business requirements and technical and accurately describing them (resolution stage). For this purpose the authors of the study used a web application for project management [3]. This enabled the efficient description of the requirements as specified and in a universal way. The next step was to examine the feasibility of the project, which de facto in this case was taken before exceptional set out requirements for the project. Such an exception was possible because of previous studies and early trials, which were presented in [2]. The next step was to accurately define the project, its structure, business model and its implementation, and testing activities. The final step was to develop a complete system capable of installation on any machine.

The mechanism of action of this search engine is similar to that of normal search engines, with the difference that will be based on the results that are returned by Google, that is, every query sent to our sophisticated data reflects a search engine google.com and adapted to the requirements of the project. Since this is a prototype version does not have complete functionality offered by standard search engines, so were used the following functionalities:

– The search will be limited to the keywords associated with the subject "Programming Web-based systems". These words are taken from the index [4], which resulted in approximately 6000 queries.

– Operation of web crawlers (web crawler) to index the data will be based on sophisticated data collection by Google and their corresponding parsed, it is possible to perform further operations.

– Results can search, because of the nature of the algorithm Best performance hit, have been reduced further to the results of which are text files with extension PDF. This limited the level of measurement noise, which in the case of HTML documents is because today's Internet model, very complex, not least because of the content delivery network (CDN) or AJAX requests (*Asynchronous JavaScript and XML*), which prevents the one hand, the distance measurement server, on the other the size of the resource.

In accordance with the requirements of the algorithm best performance hit, it needs additional information about each discovered resource from the search engine. Those are:

– size of the resource,

– AS number,

– geolocation data.

Conceptually, the algorithm has been found that these data are sent by the search engine, but none of the major search engines (Google, Bing, Yahoo) provide the information through their API, and therefore was required to create crawler, which in addition to saving the data into the database, processes each and adds the result of the above information and additionally adds an entry associated with a given keyword or keywords. Index Knowledge Base contains a sufficiently large amount of data, so it’s possible to test the performance capabilities of crawler and simultaneously have a large number of test data and test the performance of the algorithm to choose the best resource in terms of extreme performance.

3.1. SOFTWARE ENVIRONMENT

After reviewing possible programming solutions, the Django Web framework has been chosen in version 1.4. This decision was made because of the great freedom that we have to modify the code, a large availability of libraries necessary to implement the algorithm Best Performance Hit and capabilities to relatively quickly and easily create Web applications. In the case of released projects which would hardly be changed within the application, and not in any event license permit interference in the code without making modifications to their authors and the general public, which would require additional expenditure of time and there would also be eventually accepted. In the case of implementation in Python all the libraries that are necessary to comply with search engines are based on the license GPL / LGPL, and MIT, which allows for great flexibility in adapting to the needs of the algorithm.

Python version and the interpreter used in implementing the code is ver. 2.6 – unfortunately ver. 3.1, despite many improvements, did not support the libraries needed for the construction project. The biggest advantage of this solution is the use of high-level language developed primarily for rapid implementation of technology applications in the DRY (*Do not Repeat Yourself*) and mainly web application development and the use of Django, which in its structure contains most of the required elements of a complete web application. This allowed for the fulfillment of all the business requirements and technical specifications referred to in the earlier demands on programming environment, namely:

– Easy mobility made possible by the compact structure of files, high availability packages Python and Django on different OS platforms. To move applications from one server to another, we need just to have the installed environment and copy the file system without the specific hardware configuration and software. Additionally, Django was created to implement using the modified MVC (*Model-View-Controller*) design pattern, defined as MTV (*Model-Template-View*) and therefore any modification of code by people not associated with the project is feasible in a fast manner and also allows for easy tracking errors (*bug tracking*) and measure the performance of individual modules, as well as specific methods.

– Scalability, Django uses the architecture of the "shared-nothing", which allows adding hardware to any hardware level – from the database servers, cache servers to the web servers.

– Modular construction system, one of the fundamental strengths of the chosen framework is the ability to easily and quickly adapt to change. Each subsystem of the system, i.e. in the case of the present application will be a web robot, search engine and monitor the servers is a separate module and created a project to add new modules, removing or moving existing single modules for other projects require us to only make changes to the settings file in which the specific names of these modules.

3.2. CHOSEN SOLUTIONS

When creating a system, authors benefited from additional external solutions, available as ready-made modules or individual labor saving methods, allowing to obtain concrete results.

The first problem was to download data from Google, so that it's easier to make data transformations. To this end, copyright would be needed to create the parser with a controlled way to connect and retrieve data, which would be a laborious task. Therefore the authors benefited from the finished module *xgoogle*. It has ready-made methods to easily retrieve desired data and then make modifications on them. It was necessary to make some minor changes inside the module against the requirements of the project, after which all the requirements for receiving the data have been met.

To be able to determine the geographic location by coordinates indicating the server holding the resource, it was necessary to find a database with associated ranges of IP addresses to their position on the map. This was done in order to select from among several existing solutions described in more detail in [5] and [6] and compared in many aspects. On this basis it was decided to free MaxMind GeoIP Lite solution, which results in comparative tests were comparable to the pay arrangements and are sufficient for the purpose of the system. Because the application GeoIP was created in C, it would be necessary to use the API developed in Python made officially by MaxMind. However, the version was created directly in Python without using the API.

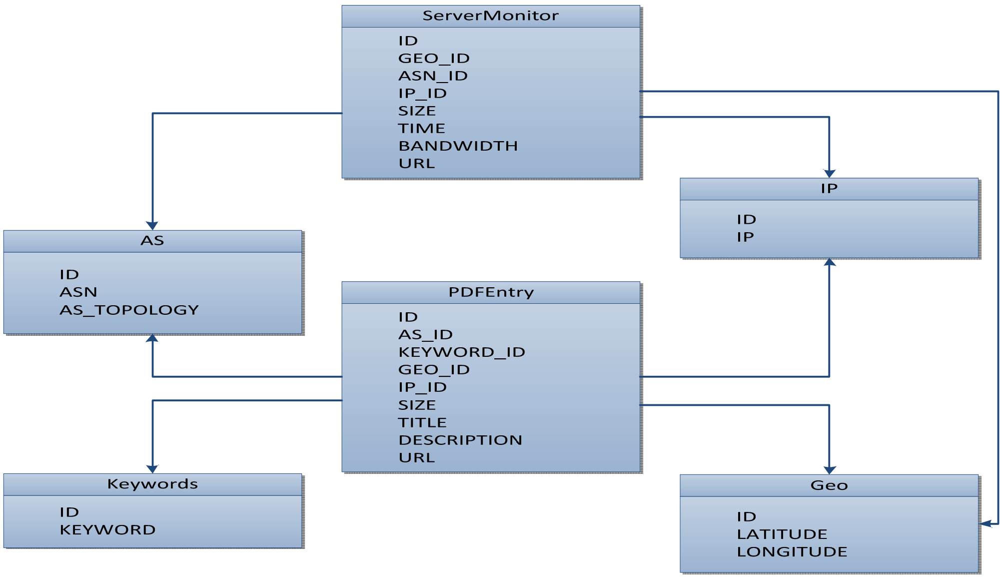


Fig. 3. Database structure of a system

To gain further investigated AS numbers of servers that have the desired resources, it is necessary to use the data available in the whois servers. Such servers are available, inter alia, by a group Cymru. In this case it was necessary to use the APIs available on the official site of the Python language.

The database is based on the RDBMS engine SQLite3. In the case of the prototype solution is sufficiently efficient, access times and query execution times are comparable with those obtained by the database based on MySQL. In addition, the aspect of the rationale for using this technology is to avoid having to use an additional server to run SQL, because SQLite does not require any additional processes to operate and perform operations. Construction of the database is based on a double star schema. The root element of the first scheme is a table containing foreign keys PDFEntry tables AS, IP, Keywords and Geo. The second scheme is based on the table ServerMonitor related foreign keys of tables AS, IP, and Geo.

4. RESULTS

Without interference measurement results for the stocks with the greatest chance of access requests as you can see are definitely better for the results using the performance algorithm. Despite appearing exacerbations, they can be regarded as a negligible value within the error limits of the algorithm. In conclusion, it is difficult to determine the algorithms work in a visible and streamline the process of obtaining resource material for the search phrase. Moreover, two algorithms have shown very similar, very high values of quality and low error rate calculation, but according to the author's assessment prove the algorithm MAX to be a better one.

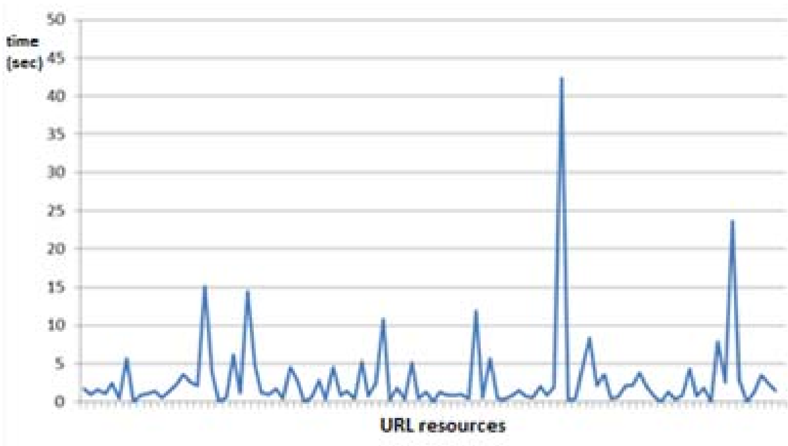


Fig. 4. Best resource obtaining times for each phrases sorted by relevance

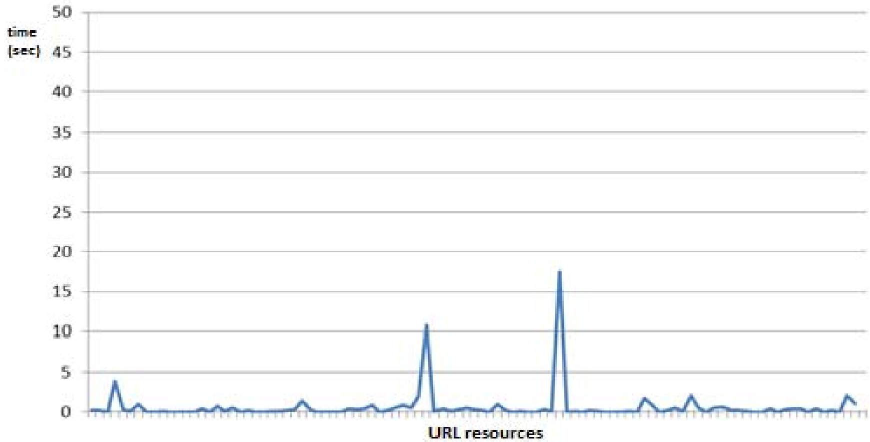


Fig. 5. Best resource obtaining times for each phrases sorted by Best Performance Hit algorithm

4. CONCLUSION

The complete application with the implemented performance search algorithms become a success. The bases for such statements are especially research, where the results surpassed expectations. According to the first attempts [1], the average efficiency of algorithms ranged from 70–80% and the actual results of work carried out on a large research group showed a value of at least 10% better. The project created from the environment and from the algorithms for best performance hit was created in accordance with the intentions of the authors and set at the start of business and technical requirements. Through careful analysis of requirements and determine how to resolve them in a transparent manner, it was possible to produce software that can be included in all of the elements necessary to include performance algorithms.

The problematic aspect was the creation of a knowledge base of primary resources search engine. However, in the aspect of creating a prototype and not the final product, after constructing a few limitations, it was possible to show the possible creation of a system to a large extent a new search quality data. Despite the created limitations the system did not manage to avoid all the pitfalls. The first trials and tests have been performed on natural results, so resulting searches were all kinds of documents. The results, however, proved to be disturbing and strongly deviate from the expected. The reasons for this state of affairs seems to be many, most have been addressed and described in the work. One of the unknown values are the content delivery network (CDN), which cannot be determined and identified at the moment in a meaningful way.

It can be said that such a solution, after implementing more of the possible extensions to search and create a sufficiently large knowledge base, could occur on a large scale and be appreciated by users demanding better, faster and more efficient solutions, especially in terms of acquiring resources.

REFERENCES

[1] BORZEMSKI L., NOWAK Z., *Best performance hit – nowa metoda pozyskiwania zasobów WWW za pośrednictwem wyszukiwarek internetowych*, In: Wysokowydajne sieci komputerowe. Zastosowanie i bezpieczeństwo, A. Kwiecień, A. Grzywak (Eds.), Warszawa, WKŁ, 2005.

[2] NOWAK Z., *Metodyka i algorytmy efektywnego pozyskiwania zasobów WWW*, PhD thesis, Wrocław, PWr, 2004.

[3] GatherSpace portal, www.GatherSpace.com, 2011.

[4] DEITEL H. M., DEITEL P. J., *Internet & World Wide Web: How to Program, 4/e*, Upper Saddle River, Prentice Hall, 2008.

[5] SHAVITT Y., ZILBERMAN N., *A Study of Geolocation Databases*, Computing Research Repository – CORR, vol. abs/1005.5, 2010.

[6] HOLZHAUER F., *IP Geolocation*, Internet Measurement seminar, CAIDA, 2007.

Content Aware Networking, Content Centric Networks,
Network of information, Future Internet

Remigiusz SAMBORSKI*

EVALUATION OF AVAILABLE CONTENT AWARE NETWORKING PROTOTYPES: CCNX AND NETINF

This document provides an evaluation of two currently available open source solutions for Content Aware Networking: Content Centric Network and Network of Information.

Main objectives of the document are: to evaluate current status of work; to show strengths and weak points of both implementations; to show possibilities for improvement in download speeds and content localization.

Available applications and scenarios in both implementations were tested and summarized. Research results include download speed measurements in CCN implementation in 3 different topologies. Best performance achieved in CCN is 3.6 times worse than a reference test made with SSH protocol. Nevertheless the article acknowledges CCN as current pioneer in Content Aware Networking solutions.

1. INTRODUCTION

1.1. CONTENT AWARE NETWORKING

The phrase Content Aware Networking (CAN) was introduced in 2001 [4]. After observing the evolution of the Internet it became obvious that currently used protocols and network approach are not efficient enough. Many flows could be avoided by changing the point of view – from connection oriented to content oriented.

Van Jacobson is calling CAN the third generation network. The first generation, the telephone networks, is about interconnecting wires, enabling users to have conver-

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

sations with each other. Second generation networking is about interconnecting devices, enabling services on those devices to communicate. The (new) third generation is about disseminating information, making that information available to applications and users efficiently on a large scale [1].

Protocols used currently in the Internet were developed in early 70s. At that point of time the main objective for their development was robustness, self-reparation and achieving no central point of failure. TCP/IP family of protocols suits well to these goals. Since that time many things have changed. Internet grew to enormous size by interconnecting a huge amount of devices all over the world. The infrastructure is getting better every day and the protocols used to build the Network are getting obsolete.. Now we are able to observe that one of the main usage areas of the Internet is content delivery and that it is done inefficiently by currently used protocols. Main drawbacks of using TCP/IP are:

- mostly host to host connections, multicast and broadcast are not well supported on large scale
- security relies on connections, the network doesn't secure the data (content)
- multihoming can't be used effectively, for instance session information can’t be naturally shared between two interfaces
- most services require permanent Internet access, there are only two states: online or offline, there is no middle state
- routing tables in the core routers are growing to huge sizes, discussion is emerging on the hypothetical problem of exploding routing tables
- congestion problems could be avoided on architectural level
- number of IP addresses is limited
- overlay networks, which allow data dissemination (i.e. BitTorrent and other Peer to Peer), may be used only for big files – time needed to find peers is too long for small requests like http

Content Aware Network is a new way to look at networking. The main difference is to identify content (what) instead of network location (where). CAN approach requires content to be signed and versioned. The content is decoupled from it’s location and way of transportation – it may be delivered to end user via Ethernet, IPv4, IPv6, Pen-drive or any other mean of data transmission. This separation has many benefits for users and developers. Other assumptions for the new generation of network are: built-in data dissemination to allow better bandwidth utilization and efficiently use devices potential; support for mobility and augmented Internet.

At the time of writing there are two implementations of CAN available with open-source licences, there are: Content Centric Network and Network of Information. Next two chapters describe shortly these implementations and give references for further reading.

1.2. CONTENT CENTRIC NETWORK

Content Centric Network is a project created and supervised by Van Jacobson at Palo Alto Research Center, USA. Figure 1 shows differences between IP and CCN protocol stacks. CCN introduces two new layers: strategy and security. The strategy layer is responsible for making choices to optimize usage of multiple conductivities in layer 2 under changing conditions.

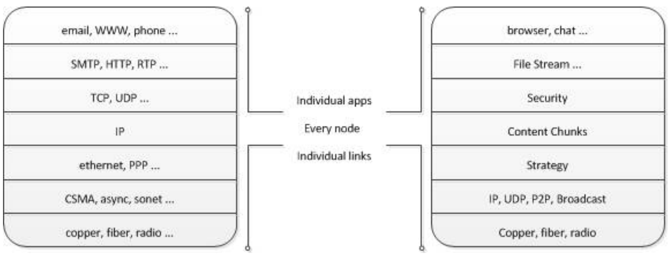


Fig. 1. IP and CCN protocol stacks [1]

Content in CCN is divided into content chunks, which are shared between hosts and are considered an universal agreement in this architecture – similar to an IP packet in IP stack. Communication is based on two types of packets: Interest and Data (figure 2). Interest packets can be called requests for data. These packets are broadcasted by the user, who demands content. An Interest packet is routed by name until it reaches a node, which has the demanded data. A Data packet returns using the same path used by an Interest packet to the user. CCN's architecture allows Interest packets to be send by multiple interfaces (called faces for distinction between IP protocols stack) simultaneously.

CCN namespace is human readable. This architecture uses URI like scheme for naming data. Routing is based on names and longest-match look-up. Data names may represent local context, for instance: /ThisRoom/projector, /Local/Friends etc.

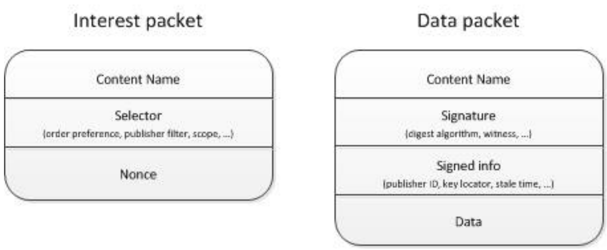


Fig. 2. Interest and Data packets in CCN [1]

Names create a hierarchical tree, which has these advantages among other: routing protocols from IP architecture may be adapted to work with CCN; trust system (key distribution) may be based on this hierarchy; content doesn't need to be registered. To keep content names human-readable they cannot be self-signing. In CCN the data and name signature is located in Data packet (figure 2).

This is only a brief introduction to CCN. More information may be found in [1, 5, 6].

1.3. NETWORK OF INFORMATION

Network of Information (ab. NetInf) was developed as a part of "4WARD Architecture and Design for the Future Internet" project. It implements the same information-centric paradigm as CCN, but there are several differences between both architectures.

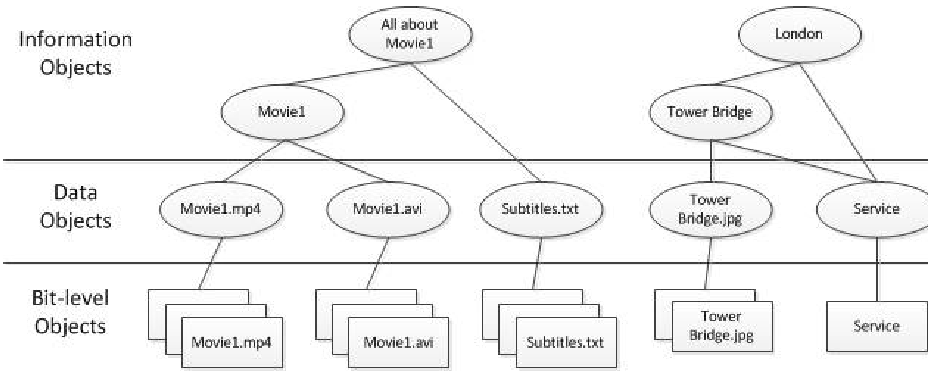


Fig. 3. NetInf objects hierarchy [3]

NetInf groups content into Objects. There are three categories of objects: Information Objects (IOs), Data Objects (DOs), Bit-level Objects (BOs). A sample objects hierarchy is presented on figure 3. Information objects represent the highest level of semantics in NetInf. IOs link to DOs, which contain the specific content. One IO may aggregate other IOs to represent content connections and hierarchy. Bit-level objects contain the data. The objects structure allows this architecture to divide data in smaller chunks.

In NetInf implementation the only difference between an IO and DO is that the first one contains meta-tags that are helpful in search processes.

Naming scheme in NetInf is different from the one used in CCN. The names are self-signed and self-certifying. Thus identifiers are not easy to remember and there is a need for a system similar to currently used DNS to resolve names. A sample content name in NetInf may look like this:

ni:HASH_OF_PK=8c4e559d464e38c68ac6a9760f4aad371470ccf9~HASH_OF_PK_IDENT=SHA1~VERSION_KIND=UNVERSIONED

The basic structure of NetInf identifiers is shown in figure 4. Each identifier consists of three parts: a mandatory, well defined type tag that defines the identifier type, an optional part P that is the cryptographic hash (e.g. SHA-1) of the owner's public key, and a mandatory label L that uniquely identifies the entity. The particular format of the label depends on the type of the identifier specified by the type tag, i.e., the type tag defines how the identifier and the specific parts of the identifier have to be interpreted [3].

Tag	P=Hash (PublicKey$_{Owner}$)	L=Label

Fig. 4. NetInf name structure [3]

Due to a non hierarchical namespace NetInf requires a Name Resolution (ab. NR) system, which will be responsible for locating content by it's id. Distributed Hash Table (DHT) based systems are a suitable approach for the implementation of the NR mechanism.

More information on NetInf may be found in [2, 3, 7].

1.4. ARTICLE'S OBJECTIVES AND STRUCTURE

Main objectives of the document are: to evaluate current status of work; to show strengths and weak points of both implementations; to show possibilities for improvement in download speeds and content localization.

This work is a summary of work done during period of March to June 2011 by students in a computer science project held at Wroclaw University of Technology.

Research was based on code analysis and practical tests of available scenarios and applications created by CCN and NetInf developers. For better understanding of API, applications were built in both prototypes. A detailed description of developed applications is not included in this work and may be discussed elsewhere. This document focuses on usefulness of each prototype as a next generation network.

Download speeds in CCN prototype were measured in 3 different topologies. All measurements were repeated at least 10 times and the best and worst values were discarded. The calculated averages are shown in following chapters.

First we introduced the main principles of CAN and described shortly each of the implementations with an emphasis on their differences. Next a presentation of collected data and other findings will follow. At the end some guidelines for future work will be given.

2. EVALUATION

2.1. CCNx – OVERVIEW

In tests CCNx version 0.4.0 downloaded from projects website [6] was used. Notebook computers with Ubuntu 10.10 operating system and mobile phones with Android 2.2 installed were used.

CCNx prototype has two code trees, one is written in C/Posix and the other is written in JAVA. In the analysed version the JAVA tree is more complex and has more functionalities available. Nevertheless we presume further C/Posix development to be done due to performance issues of JAVA and a possibility to integrate CCNx in Linux kernel.

Core of the system is a daemon called ccnd, which is responsible for CCNx communication. It normally runs in background and listens on a specific UDP port (default is 9695) for data from other nodes.

This implementation includes two simple applications, which can be used to test and present it's basic concepts in practice: ccnChat – a simple chat application; ccnFileProxy – an application allowing file transfers in CCN network.

There is also a number of helpful utilities, for example: ccnexplore – used to explore repositories; ccndc – allows modifications to forwarding table (FIB); ccngetfile – retrieve a file published as CCN content and save it to a local file; ccnputfile - publish a file as CCN content.

More information on the prototype may be found in [9].

2.2. CCNCHAT AND CCNFILEPROXY

ccnChat application is available on two platforms: Linux and Android. It requires a content object repository to be created. Unfortunately there is only a Linux implementation of the content object repository available. This means that there needs to be a Linux (PC) running a repository in every usage scenario. We were able to communicate using CCN protocol between two and more Linuxes and also to communicate between a PC with Linux and a mobile device with Android system. Chat can't be run to connect two Android devices due to repository boundaries described above.

ccnFileProxy application is available only on Linux platform. The core of this application is a content object repository created in a chosen folder. This repository is analogous to the one used by ccnChat application. After creating a content repository nodes may use ccnputfile and ccngetfile utilities to operate on it's content. Our tests have proved that files transferred using the above utilities were fulfilling CAN network paradigm.

Both applications presented above require CCNx to have proper forwarding table (FIB) entries. These entries consist of two values: content name, target IP address. FIB is used by CCN daemon (ccnd) to determine, where Interest packets should be sent. In our tests we specified static target IPs for content names being tested.

2.3. CCNX FILE TRANSFER MEASUREMENTS

We choose ccnFileProxy application for performance tests in CCN prototype due to it's maturity. Our first set of tests consisted of 3 samples presented in table 1.

Table 1. CCN test 1 – samples

Sample number	Number of files in sample	Size of a single file in sample [b]	Total sample size [b]
1	1000	100000	100000000
2	20	5000000	
3	2	50000000	

We have tested two different topologies presented on figure 5 and 6.

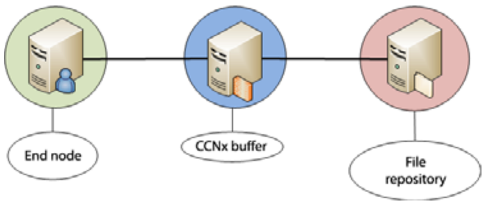


Fig. 5. CCN test 1 topology 1

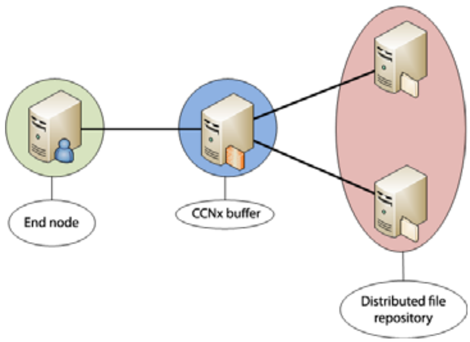


Fig. 6. CCN test 1 topology 2

In topology 1 the whole sample was located in one content repository. In topology 2 every sample was split in half between two repositories. First measurement was made with CCNx buffer node empty and the second one, when it contained buffered data. Every measurement was made 10 times, the best and worst value were left behind and an average from 8 results was calculated. We simulated 3 different bandwidths: 5Mbit/s, 10 Mbit/s, 20Mbit/s. Tables 2 and 3 contains results of this experiment. Time is represented in the following format – minutes: seconds, miliseconds.

Table 2: CCN test 1 topology 1 results

Sample number		5 Mbit/s	10 Mbit/s	20 Mbit/s
I	Empty buffer	34:48,70	33:37,40	33:28,30
	Full buffer	33:32,00	32:15,00	32:05,00
II	Empty buffer	03:56,10	02:23,60	01:51,20
	Full buffer	03:48,00	02:12,10	01:40,70
III	Empty buffer	03:21,00	02:14,20	01:08,70
	Full buffer	03:09,30	01:43,20	01:02,70

Table 3. CCN test 1 topology 2 results

I	Empty buffer	44:05,10	34:14,70	32:54,80
	Full buffer	34:18,20	33:01,00	32:39,90
II	Empty buffer	04:18,40	03:43,00	03:21,00
	Full buffer	04:07,20	03:24,60	02:54,00
III	Empty buffer	04:05,30	02:48,80	02:13,40
	Full buffer	03:49,30	02:28,80	02:03,20

Data is divided into two groups: empty buffer and full buffer. First group is a situation, when no data was recently downloaded through the CCNx buffer node and it must be downloaded directly from file repository. Second group represents a situation, in which CCNx buffer is serving the buffered data to end node.

The buffering at CCNx buffer node resulted in shorter download times with every sample, bandwidth and topology. The best result was observed at sample III topology 1 and bandwidth of 10Mbit/s with a 23,13% of time reduction. The worst situation was at sample I and bandwidth of 20Mbit/s with only a 0,76% time reduction. We are planning to repeat these tests with different topologies, bottlenecks and many concurrent end-nodes involved.

There is a huge time difference between sample I and II, III. The total size of all samples is the same, but sample number I consists of many files (1000). The overtime might be connected to content segmentation, which is CPU consuming and must be done before the content is available in CCNx. The segmentation is described with more details in the second test.

After splitting the content between two repositories the download times got worse, even after buffering the data at the CCNx buffer. This behaviour requires further studies.

The second test was prepared in topology showed on figure 7. Three sizes of samples were used: 14.4MB, 52.8MB and 211.2MB. The link speed was set to 5Mbit/s.

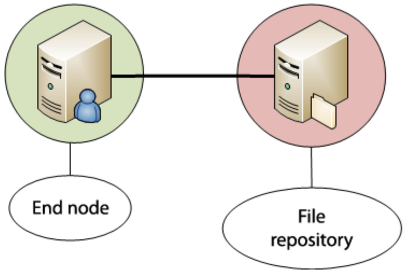


Fig. 7. CCN test 2 topology

Method of tests was similar to the one used in the first test. Ten measurements were made and two extreme (minimum and maximum) values were discarded. Average was calculated from the remaining 8 values. Results are presented in tables 4 and 5. Figure 8 shows results on a diagram.

Table 4. CCN test 2 results – download times and speeds

Sample		Download time [min:sec]	Download speed [KB/s]
Name	Size [MB]		
CCNX – not segmented	14,4	02:05	117,67
CCNX – segmented	14,4	00:39	375,46
CCNX – not segmented	52,8	07:32	119,61
CCNX – segmented	52,8	01:58	455,25
SSH – reference	52,8	00:32	1641,38
CCNX – not segmented	211,2	29:23	122,62
CCNX – segmented	211,2	09:46	368,84

Table 5. CCN test 2 results - average download speed

	Average download speed [KB/s]
CCNX – not segmented	119,97
CCNX – segmented	399,85

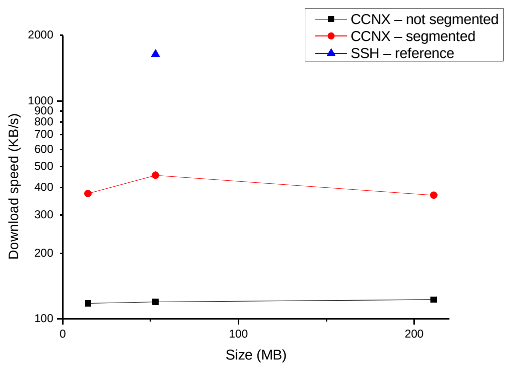


Fig. 8. CCN test 2 results – diagram

The second test shows results in two categories: not segmented and segmented. All content sent via CCN protocol is divided into content chunks (segments). For reference an SSH file transfer was measured for 52.8MB sample.

Content segmentation is very CPU consuming. We did some reference tests (not showed in this document) on a slower machine and got very bad results in this process. When content isn't segmented the measured time is a sum of time used to create content chunks and the time of data sending. In the second category the time is shorter, because no segmentation has to be done.

We were able to achieve a best speed of 455,25 KB/s with a 52,8 MB file. The speed of SSH transfer for the same file was 3.6 times better. One of the explanations is that CCNx is in a very early state of development and further optimization must be made before it can be used broadly.

2.4. CCNx – SUMMARY

The CCNx prototype is still undergoing heavy development. The documentation is not as extensive as in NetInf, but the implementation is more robust and carefully thought out.

We were able to run series of performance tests and develop applications using this network architecture. The android implementation could be used to create software that communicated via CCN with a PC. There is still no support for communication between two mobile devices, but this should change with new versions of this prototype.

At the time of writing, CCNx implements only static FIB entries, which are responsible for content locating. Multicast or broadcast address may be used in the forwarding table, but there is no implementation of dynamic algorithms available yet. Authors suggest that protocols used currently in IP networks (i.e. IS-IS or OSPF) may be adapted to CCN [1].

2.5. NETINF – OVERVIEW

In our tests we used NetInf Virtual Machine version beta 3 downloaded from the project's site [7]. The VM was run in Oracle Virtual Box [11].

The Virtual Machine is running a Linux Ubuntu operating system with preinstalled NetInf, Eclipse development IDE and inFox (NetInf firefox plugin).

NetInf prototype is written in Java and is well documented. There are two scenarios, which can be used to test it's capabilities. Main goal of scenario 1 is to show data integrity and data availability features of NetInf prototype. The second scenario is about update awareness using a built-in Event Service. More information about scenarios may be found in NetInf documentation [10].

The prototype includes two software plugins: Firefox (called inFox) and Thunderbird. Next chapters describe tests done on NetInf prototype.

2.6. NetInf – SCENARIO AND PLUGIN TESTS

After many tests, changes in configuration and changes in source files we were unable to fully deploy scenario 1. We tried every version of NetInf virtual machine (from beta 1 to 3) and also our own installations of NetInf from source codes available at project's site [7]. The code is very complicated and hard to debug. We were able to run a single node with management tool to create and edit Information/Data objects, but communication with other nodes was not possible to achieve.

We contacted NetInf team and were able to get responses with fixes to some discovered bugs. These bugfixes allowed us to fully run scenario 2.

Scenario 2 in VM version beta 3 is working properly. It is possible to communicate between two machines using the same network connection (WIFI or Ethernet). We were able to observe Event Service in action. We haven't tried connecting nodes from different networks (routing involved), but due to usage of TCP as the link layer there should be no problems with such configurations.

The inFox plugin was tested and a chrome plugin was written for test purposes. Both plugins use NetInf testbed available at project's site [7] to resolve and download Information and Data objects for websites identified by a NetInf object name (addresses starting with 'ni:HASH_OF_PK='). The information object doesn't contain the website itself – it only contains a special attribute called ‘Locator’, which is a website URL. After reading new URL the browser is redirected to this address. A different behavior occurs with a test Data Object, which contains image data. The image data is downloaded in a DO and presented in the browser. After downloading IO or DO the browser plugin makes integrity checks to verify the data was not spoofed or destroyed.

2.7. NetInf SUMMARY

NetInf doesn't provide tools for file transfers yet. Scenario 1, which could be potentially used to make Data Object transfers for time and speed measurements was impossible to deploy. Therefore we couldn't make similar speed tests to the ones presented in CCNX.

When starting with the evaluation NetInf project seemed to be in a more advanced and stable state than the CCNX project. This assumption was wrong. NetInf has a spacious documentation, but the source code and beta VM doesn't stand for it solidly.

One of the big issues was tied to node identity. Every time a node was created it required a new key identity file. This restraint led to loosing all data from previous session. By data we understand IOs and DOs. There was no way to access them once the node was restarted.

We were able to develop some applications on this platform based on scenario 2 for demonstration purposes, but they suffer from problems in the prototype itself. Nevertheless some of the ideas in NetInf are worth considering in future networking – for instance the build-in Event Service and DHT like name look-ups.

3. SUMMARY/FUTURE WORK

Table 6 gathers differences between both evaluated prototypes. The CCN prototype and protocol has many similarities to TCP/IP protocols stack. There are no central nodes required for content registration and look-up. The decisions are made on node level. We expect the CCN project to be growing faster and to have a greater potential to become a next generation network than NetInf.

Table 6. Comparison table: CCNx vs NetInf

	CCNx	**NetInf**
Naming scheme	URI like names, human-readable	Self-certifing and self-signing names, not easy to remember, require DNS like tools to be human-readable
Content registration	Not required	Required
Implementation – api availability	C, JAVA, Android	JAVA
Implementation – general	Code easy to build, application can be run out of the box	Many problems encountered; only one of twa scenarios was possible to be fully tested; testbed available at projects site suffered from frequent outages
Content locating	Every node keeps a forwarding table; only static entries available	Special Name Resolution nodes required using DHT like protocols to locate content
Events notifications	No built-in support	Event Service nodes available
Documentation	Basic	Very extensive, not always in accordance with implementation

CCN currently supports only static entries in forwarding tables (FIBs), this creates a place for further studies and development. Available routing protocols may be used and accommodated for this CAN network. The IP routing protocols will need to be changed to achieve good results with a new paradigm. These protocols were optimized for routing to one location only whereas in CAN the routing will be taking place to many locations.

During future studies we will be trying choose the best from available protocols and to optimize it to new conditions.

REFERENCES

[1] Van JACOBSON,. SMETTERS D.K, THORNTON J.D., PLASS M.F., BRIGGS N.H.,. BRAYNARD R.L, *Networking Named Content*, CoNEXT 2009 Conference.

[2] 4WARD Team, *Introduction to a 4WARD Architectural Approach*, 2009.
[3] 4WARD Team, *D-6.1 First NetInf architecture description*, 2009.
[4] SUBBIAH B., UZMI Z.A., *Content Aware Networking in the Internet: Issues and Challenges,* ICC 06.2001.
[5] Van JACOBSON, *A New Way to look at Networking*, Google Conference, 30.09.2006.
[6] Project CCNx™. http://www.ccnx.org, Sep. 2009.
[7] Project NetInf, http://www.netinf.org, 2011.
[8] Van JaCobson, Diana K. Smetters, *Securing Network Content,* 2009.
[9] CCNx™ project group, *CCNx Documentation,* 2011.
[10] 4WARD Team, *NetInf Documentation,* 2011.
[11] Oracle VirtualBox Project, http://www.virtualbox.org, 2011.

GLOSSARY

BO	– Bit Object used in NetInf architecture
CAN	– Content Aware Network
CCN/CCNx	– Content Centric Network – implementation of CAN developed at PARC
DO	– Data Object used in NetInf architecture
FIB	– Forwarding Information Base – in CCN a table similar to a routing table in IP, it's responsible for Interest packets destinations
IO	– Information Object used in NetInf architecture
Multihoming	– a situation, in which one device has multiple parallel Internet connections, for instance a mobile phone using 3G and Wifi connection at the same time
NetInf	– Network of Information – implementation of CAN developed by 4WARD Team
VM	– Virtual Machine

data storage, virtualization, data security

Tomasz BILSKI*

TOWARDS SECURE NETWORK STORAGE

The chapter presents new, general concept of data storage system based on network environment with a special emphasis on data security. A purpose of the concept is to provide tools for synthesis as well as analysis of storage systems. The concept is based on set of user requirements (level of security, performance, cost, …) and set of security measurements and technology features (quantitative as well as qualitative). The question is if it is possible to meet the user requirements with a use of available security measurements, technologies and acceptable cost. Security is usually defined with a use of 3 requirements: confidentiality, integrity and availability. The requirements are well known but new features, factors and architectures of storage system encourage to look for new, pragmatic explanations of them. It may be simply said that the requirements should be always met. In network storage systems this means fulfilling the requirements in every data location (local, remote), in every data state (storage, transmission) and in every operation performed on data. Here we propose security analysis with a use of security matrices.

1. INTRODUCTION

1.1. NEW CHALLENGES, NEW TRADEOFFS

Contemporary information systems are rapidly changing. New challenges are emerging almost every day. New aspects should be taken into account in order to design competitive IT system. One of general trends visible in data storage evolution is virtualization of storage systems based on networking technologies. In data storage design process one has to take into account many new available and emerging technologies as well as factors and features that were not considered in the past. Here we may enumerate only some of them: increased security awareness, cloud computing, backups of virtual machines, exponential growth of volume of data and number of data units, more stringent requirements for recovery time objectives and recovery

* Poznań University of Technology, Pl. Sklodowskiej-Curie 5, 60-965 Poznan, Poland.

point objectives, storage tiering, flash memory, increased user mobility, ecology awareness and so on. New tradeoffs between different factors and features are observable. It is questionable if the old storage models and systems are suitable for such modern technologies, factors and features.

1.2. SECURITY MODELS

Security is defined with a use of three general requirements: confidentiality, integrity and availability. In the past several security models were proposed that deal with the requirements. Some of the existing models were dedicated to operating system analysis. Classical examples are: Bell-LaPadula model [1] used for years as security evaluation standard (e.g. in TCSEC) and matrix model proposed by Harrison, Ruzzo and Ullman in 1976 [9]. Another set of models was developed for database system security evaluation. Database security is evaluated in the following exemplary models: Jajodia-Sandhu [10], Smith-Winslett [12], Sea View [7].

Complex, contemporary information systems working in virtual, network environment couldn't be evaluated only with a use of security models from the past. At the moment we have to deal with a set of new circumstances, requirements and new tradeoffs which were not incorporated into security models 20 or 30 years ago.

1.3. CONTEMPORARY DATA STORAGE SYSTEMS – SURVEY OF REQUIREMENTS

IT systems are changing. Some of the changes are induced by changes in user's requirements. There is a set of general requirements that are common and fixed, e.g. high performance, ergonomics. Nevertheless, some requirements are becoming more important and new requirements are cropping up. A set of requirements associated with storage system is large and diversified – requirements are associated with technology, economy, law. The set consists of requirements related to:

- ecology awareness,
- power consumption,
- increased user mobility,
- scalability,
- regulatory compliance,
- RPO (Recovery Point Objective), RTO (Recovery Time Objective),
- ROI (Return on Investment).

In the process of system construction, configuration and normal operation we have to fulfil a set of such prerequisites. At the same time data security level must not be compromised.

1.4. CONTEMPORARY DATA STORAGE SYSTEMS – SURVEY OF TECHNIQUES

In contemporary data storage systems we may encounter storage techniques/methods that may be divided into three categories:

- well-known and utilized for years without essential modifications,
- utilized for years but adjusted or enhanced for modern systems,
- emerging, unknown in the past.

Among the techniques/methods that are well-known and that are used for long time without major changes we have, for example: file systems, some storage hardware (magnetic tapes, disks, RAIDs).

The second category consists of techniques that evolved from the ones used in the past. For example HSM (Hierarchical Storage Management) systems used in the past together with mainframes evolved into multitiered storage systems – in a typical scenario there are three tiers: flash, hard disk, tape and pieces of data migrate between the tiers in accordance with usage frequency. SCSI bus designed for local connections more than 25 years ago extended its deployment to Internet – iSCSI[1] protocol that can be run over existing IP network infrastructure is today a popular method for access to remote storage system. The same happened to FC (Fibre Channel) protocol which evolved into several branches: FCoE – Ethernet version as well as iFCP and FCIP – two versions dedicated to IP networks [2].

From the security point of view the last category is especially important. Let us focus on such techniques and methods that are relatively new and fundamentally different from the ones used in the past. All of them have an impact on data security aspects and should be included in security evaluation processes. Among the emerging, data storage techniques we have:

- deduplication,
- thin provisioning,
- cloud storage and storage virtualization (based on networks),
- flash memory hardware.

Deduplication

Deduplication is removal of all non-unique data units from backup system. Units of data are compared and only unique units are physically written to backup storage. In the units comparison process hashes (e.g. MD5, SHA) of data units are used. The hashes are also used as indices to lookup table. Deduplication systems may be classified according to unit size and deduplication phase time/space location. From unit size point of view we have two types of deduplication system. One is based on files, the other – on blocks of data. In the first case identical files are removed (one copy is preserved while the other is

[1] It must be noted here that iSCSI inherits from SCSI lack of security measures and operates in Internet as a cleartext protocol without cryptographic protection for transferred data.

replaced with an index) from the backup system. Redundant copy of a file is removed even if files have different names. In the case of block-based approach data comparison is performed in a global scale – identical blocks in different files are removed from the system. Deduplication may be performed before data are stored on disk (in-band mode of operation) or after they are stored on disk (out-of-band mode of operation).

There are several issues related to the security of storage system with deduplication function. It is obvious that deduplication is in opposition to redundancy used widely to protect data availability. So each storage system with deduplication function is more vulnerable to availability threats. Furthermore, since deduplication uses hash functions to compare units of data very small risk[2] related to collisions must not be excluded – two different units of data with the same hash values (particularly in the case of MD5) may be identified as two identical units and one of them will be mistakenly removed from the system [6]. And finally it must be noted that deduplication is useless if backuped data are encrypted. General requirement for encryption systems says that two encrypted chunks of the same data should be different – so deduplication of such chunks after encryption could not be performed.

Probability of collision in deduplication system is extremely low. It can be reduced by using hash function that produces longer digest. However the reduction is at the expense of performance. For example two times longer hash output value means approximately four times longer hash function processing [11].

Storage virtualization

Storage virtualization is one of many emerging IT trends. Shortly speaking, storage virtualization is the logical abstraction of physical storage systems and media. The main purpose is to hide the complexity of this physical storage in order to: simplify storage management, reduce management costs and provide economical, scalable solution to user demands. Virtualization goals should be achieved without loss of data integrity, reliability and performance. There are many techniques of virtualization. Among them: virtualization at the host, virtualization at the storage system, fabric-based virtualization. Storage virtualization is based on networks and supported by many innovative technologies: FC, iSCSI, SAN, VLAN, VRAID, multitiered systems.

Thin provisioning

It was mentioned that virtualization should provide economical, scalable solution to user demands. One of such techniques is thin provisioning. In general thin provisioning means storage allocation to servers and applications on a just-enough and just-in-time basis[3]. Servers and applications share common pool of storage. Storage space

[2] Much smaller than the risk of hard disk bit error, which is at the level of about 10^{-14}.

[3] This is in contrast with traditional way of storage management in which one has to estimate the amount of storage particular applications will need over time and provide that disk space in advance. So, more hardware than needed should be bought and poorly utilized disks spin ceaselessly.

for a given user is allocated only as data are written. User purchases only the storage he needs at the moment.

From security point of view here we have common situation. A resource is shared among some subjects. One greedy subject requesting vast amount of storage may potentially starve other subjects. Thin provisioning system which is incorrectly configured may lead to problems with data availability. An important factor is proper setting of threshold alerts. If there are no threshold alerts or they are set at very high value the available storage space may be drained and applications may shutdown or users may lose their data. It is important to have sufficient time for adding disk space (real, not virtual) when used storage space is reaching threshold.

Flash memory hardware

Flash memory modules are becoming cheaper and more popular. More and more often they are used as a main storage: on line as well as a tier in backup system. Some features of this type of storage medium are unique. Any secure system utilizing flash memory as storage hardware should be aware of the features. The exemplary unique features of flash are limited number of read/write cycles (in low cost Multi Level Cell (MLC) technology) and extremely high bit error rate (tab. 1).

Table 1. Reliability features of contemporary storage media

Storage medium	Bit error rate (BER)	Number of read/write cycles
Flash SLC	10^{-9}–10^{-11}	100 000–1000 000
Flash MLC (2 bits/cell)	10^{-5}–10^{-7}	10 000
Flash MLC (4 bits/cell)	$<10^{-5}$	1000
Hard disk	10^{-14}–10^{-15}	1000 000
Tape (DLT)	10^{-11}–10^{-15}	10 000 (load cycles)

2. NETWORK STORAGE SECURITY ISSUES

It may be simply said that the security requirements should be always met. In network storage systems this means fulfilling the requirements in every data location, in every data state and in every operation performed on data files or data units.

Security of network storage system may be analyzed from many points of view. First of all in such a system we have 3 general locations in which data security may be violated:

- original, local system,
- network (private or public),
- remote storage system.

Another, more detailed approach to the security problem may be based on data status and type of operation performed on data. We have such statuses and operations as:

- normal processing at local system,
- storage at local system,
- transmission from local to remote storage,
- storage at remote system,
- migration/demigration between different storage tiers,
- transmission from remote to local system,
- deduplication,
- end of data lifetime (permanent erasure).

Seemingly, the tasks of storage and transmission are different. The tasks are performed with a use of diverse software and hardware technologies. But, the basic properties of these services are the same. Data storage may be described as data transmission through time. In this case, "the sender" and "the receiver" are usually not separated. On the other hand data transmission may be described as brief data storage in the communication medium. The "write" and "read" processes are performed by two distant elements. In both cases the process is based on the changes of the medium states. The changes of the states are stimulated by the write/send element and sensed by the read/receive element. So it is possible to evaluate jointly both tasks [4].

3. SECURITY MATRICES

3.1. MATRIX TEMPLATE

Knowing security requirements and operations and statuses of data we may define a matrix template for security analysis in which rows are related to different operations and states, while columns are related to different security requirements (tab. 2).

Table 2. Security matrix template

Operation/status	Requirements		
	Confidentiality	Integrity	Availability
…			
…			
…			

In this way we may define several types of security matrix: threat matrix, risk level matrix, protection matrix. The matrices may be used to analyze security issues related to data storage, particularly:

- threat classification,
- comprehensiveness of security measures,
- relations between different security measures and requirements.

The relations between security measures and requirements may be positive or negative. Positive relation occurs if a security measure used to fulfill particular requirement in a particular data operation/status fulfills or makes simpler to fulfill another requirement in the same operation/status or the same requirement in another operation/status. For example, some policies of data encryption protect data at the time of storage as well as transmission [3, 5]. Negative relation occurs if a security measure used to fulfill particular requirement in a particular data operation/status makes harder to fulfill another requirement in the same operation/status or the same requirement in another operation/status. For example, data encryption may decrease availability level since decryption key loss leads to data unavailability. It is very important to discover all such relations in complex security system.

3.2. THREAT MATRIX

Threat matrix consists of threats and vulnerabilities associated with each operation/status/security requirement tuple. It must be said that some threats are related to particular network/storage technology while other threats are independent of a given technology. Particular set of threats is dependent on a given system features – table 3 presents exemplary threat matrix.

Table 3. Threat matrix example

Operation/status	Requirements		
	Confidentiality	Integrity	Availability
Normal processing at local system	Trojans, spyware, insider abuse	User error	
Storage at local system	Hardware theft, Trojans	Trojans	Hardware theft, different forms of disasters
Transmission from local to remote	Wire tapping	Network transmission errors	
Storage at remote			Hardware theft, different forms of disasters
Tier 1 (flash)	Hardware theft	Read/write cycles exhaustion	
Tier 2 (hard disk)			
Tier 3 (tape)			
Deduplication		Hash collision risk	
Migration/demigration		Software errors Bus transmission errors	
Transmission from remote to local	Wire tapping	Network transmission errors	
End of data lifetime	Hardware theft	–	–

3.3. RISK LEVEL MATRIX

Approximate data on security risk levels related to particular requirement/operation/status tuple may be provided and presented in risk level matrix. The matrix may be used to categorize security problems and select a set of the problems which should be treated with high priority. Risk levels are dependent on a given system features – table 4 presents some data on risks related to different operations/statuses. Such data may be collected from product specifications, standards and security surveys (e.g. annual CSI/FBI Computer Crime and Security Survey[4]).

Table 4. Risk level matrix example

Operation/status	Requirements		
	Confidentiality	Integrity	Availability
Normal processing at local system	Insider abuse risk ~50 %		
Storage at local system		Hard disk BER 10^{-14}	
Transmission from local to remote		IEEE 802.11 BER 10^{-5}	
Storage at remote			
Tier 1 (flash)		BER as low as 10^{-5} (MLC)	
Tier 2 (hard disk)		Hard disk BER 10^{-14}	
Tier 3 (tape)		Exemplary tape BER 10^{-15}	
Deduplication		MD5 collision probability (40 PB, chunks 4 KB) 10^{-13}	
Migration/demigration		FC BER 10^{-12}	
Transmission from remote to local		IEEE 802.11 BER 10^{-5}	
End of data lifetime			

3.4. PROTECTION MATRIX

A set of necessary methods and tools for system protection may be presented in the form of protection matrix. It must be noted that such attempt to associate security component with each threat/vulnerability is discussed for years (e.g. [8]) and widely adopted (e.g. in security policies). Method proposed here is especially dedicated to network storage systems. Particular set of security measures is dependent on two security matrices mentioned previously – table 5 presents exemplary protection matrix. It must be noted that security measures may be incorporated into hardware, integrated with operating system or application.

[4] http://gocsi.com

Table 5. Protection matrix example

Operation/status	Requirements		
	Confidentiality	Integrity	Availability
Normal processing at local system	Antivirus		
Storage at local system	Data encryption	RAID, suitable storage environment conditions	
Transmission from local to remote	SSL, IPSec	CRC, ECC	
Storage at remote		Suitable storage environment conditions, backup	
Tier 1 (flash)	File encryption	CRC	
Tier 2 (hard disk)	Disk encryption	CRC, SMART, RAID (and other forms of redundancy)	
Tier 3 (tape)	Tape encryption	CRC	
Deduplication		Hash function with long output. Additional check for chunks with identical hashes.	
Migration/demigration	SAN with redundant channels		
Transmission from remote to local	SSL, IPSec	CRC, ECC	
End of data lifetime	Degaussing, overwriting	-	-

5. CONCLUSION

Data protection and security analysis become more and more complex and tricky. Numerous security models were proposed in the past, but contemporary information systems working in virtual, network environment couldn't be evaluated with a use of security models from the past. We have to deal with a set of circumstances, requirements and tradeoffs that were not incorporated into security models 20 or 30 years ago. In the chapter we have presented yet another general method to facilitate and aid the processes of system protection. The template and the matrices may be used to split the complete problem/task into many simpler ones and to discover relations between different security requirements, different threats and different security measures.

ACKNOWLEDGMENTS

The research project is scheduled for years 2010–2013 and supported partly by scientific grant from the Polish Ministry of Education and Science.

REFERENCES

[1] BELL D. E., LAPADULA L. J., *Secure computer systems: Mathematical foundations and model*, M74-244, MITRE Corp., Bedford, MA, 1974.

[2] BILSKI T., *A Model for Storage Processes in Network Environment and its Implementation*, Proceedings Fifth IEEE International Workshop on Storage Network Architecture and Parallel I/Os SNAPI 2008, IEEE, 2008, pp. 47–53.

[3] BILSKI T., *Hard disk data organization and encryption granularity*, [In:] Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, (Eds. A. Grzech, L. Borzemski, J. Świątek, Z. Wilimowska), Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2007, pp. 107–116.

[4] BILSKI T., *A Formal Model for Data Storage Security Evaluation*, [In:] Selected Papers of the Fifth International Conference on Computational Science and Applications, (Eds. M. Gavrilova, O. Gervasi), Kuala Lumpur, IEEE, 2007, pp. 253–257.

[5] BILSKI T., *Network Backup Confidentiality Evaluation*, [In:] ACS 2006 13th International Multi-Conference on Advanced Computer Systems (Eds. Pejaś J., El Fray I., Saeed K.), Międzyzdroje, 18–20 October 2006, Vol. I, pp. 321–330.

[6] CANNON D., *Data Deduplication and Tivoli Storage Manager*, IBM Corporation 2009, http://www.ibm.com/developerworks/wikis/download/attachments/106987789/TSMDataDeduplication.pdf?version=1

[7] CASTANO S., FUGINI M., MARTELLA G., SAMARATI P., *Database Security*, Addison-Wesley, Wokingham, 1995.

[8] CLEMENTS D., *Fuzzy models for computer security metrics*, Ph.D. Thesis, University of California Department of Electrical Engineering and Computer Science, Berkeley, CA, 1977.

[9] HARRISON M. A., RUZZO W. L., ULLMAN J.D., *Protection in operating systems*, Communications of the ACM, Vol. 19, No. 8, 1976, p. 461-471.

[10] JAJODIA S., SANDHU R., *Toward a multilevel relational data model*, Proceeding of the ACM-SIGMOD Conference on Management of Data, Denver, CO, 1991, pp. 50–59.

[11] NAKAJIMA J., MATSUI M., *Performance Analysis and Parallel Implementation of Dedicated Hash Functions*, Proceedings EUROCRYPT '02 Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology, Springer-Verlag London, 2002.

[12] SMITH K., WINSLETT M., *Multilevel secure rules: Integrating the multilevel secure and active data models*, Thuraisingham B. M., Landwehr C.E. (ed.), Database Security IV. Status and Prospects, Elsevier, North-Holland 1993, pp. 35–53.

PART II

KNOWLEDGE DISCOVERY AND ENGINEERING

Decisional DNA, Set of Experience Knowledge Structure, Interactive TV, Digital TV, XML

Haoxi ZHANG*, Cesar SANIN*, Edward SZCZERBICKI*

GAINING KNOWLEDGE THROUGH EXPERIENCE IN DIGITAL TV

Interactive television is an evolutionary integration of the Internet and digital TV. It allows viewers to interact with television content and services. Thanks to the booming of digital TV, viewers' TV watch experience could be extremely valuable. By running customized applications either inside the digital TV or at viewers' set-top boxes, we can capture viewers' TV watch experience. In this work, we introduce a new approach called Decisional DNA Digital TV in order to capture, reuse, and share the viewers' TV watch experience and preference; and we present the features, architecture and initial experimental results of this approach. Decisional DNA is a domain-independent, flexible, smart knowledge representation structure which allows its domains to acquire, reuse, evolve and share knowledge in an easy and standard way.

1. INTRODUCTION

Thanks to digitalization and computerization of our daily life, the TV set is becoming intelligent and interactive, or even becoming a computer. Many companies and organizations have involved into the implementation of such intelligent and interactive TV, like Oracle, the Digital Video Broadcasting Project (DVB) [7], Apple, and Google. Also, a few solutions have been offered by these companies, such as Java TV [11], Multimedia Home Platform [13], and Google TV [9].

By using these existing solutions, developers can add interactive functions into digital TV sets. In this work, a domain-independent and standard approach, called the Decisional DNA Digital TV (DDNA DTV) is presented. It is based on the Java TV platform

* The University of Newcastle, University Drive, Callaghan, 2308, NSW, Australia.
e-mail: Haoxi.Zhang@uon.edu.au; Cesar.Sanin@Newcastle.edu.au;
Edward.Szczerbicki@Newcastle.edu.au

[11], and using a novel knowledge representation structure – Decisional DNA [2] to capture, reuse, and share viewers' TV watch experiences.

This paper is organized as follows: section two describes an academic background on basic concepts related to our work; section three shows the features and architecture for the DDNA DTV Systems. Section four presents the simulation and experiments of our work. Finally, in section five, concluding remarks are drawn.

2. BACKGROUND

2.1. DIGITAL TV AND INTERACTIVE TV

Digital television (DTV) is the television broadcasting system that uses the digital signals to transmit program contents [15]. DTV not only delivers distortion-free video and audio signals, but also it offers much higher radio spectrum efficiency than analog television does. Furthermore, DTV can seamlessly integrate with other digital media, communication systems, and computer networks , enabling interactive multimedia services and data transmission [15].

There are a number of various picture formats defined by the combination of size, aspect ratio (width to height ratio), and interlacing supported by digital television. With digital terrestrial television broadcasting in the world, these various picture formats can be generally divided into two types: Standard Definition TV (SDTV) and High-definition television (HDTV) [16].

In supporting the reception of digital television, there are various ways: using an antenna (known as an aerial in some countries) is one of the traditional solutions of receiving DTV (and TV in general). This way is known as Digital Terrestrial Television (DTT) [17]. With DTT, the signal quality and channels to the viewers are limited by the antenna. Apart from DTT, there are other ways have been introduced to receive digital television. Among them, digital satellite and digital cable are the most familiar ways to people. In some countries, digital Multichannel Multipoint Distribution Service (MMDS) [18] is used in supporting transmissions of TV signals over microwaves. Other standards, such as Internet Protocol TV (IPTV) [19] is relying on Digital Subscriber Line (DSL) or optical cable line, and support receiving TV via Internet Protocol; Digital Multimedia Broadcasting (DMB) [14] and Digital Video Broadcasting - Handheld (DVB-H) [12] are introduced to enable handheld devices such as mobile phones to receive TV signals.

Interactive television (also known as iTV) is an evolutionary integration of the Internet and DTV [8]. It contains a bunch of techniques, and enables viewers to interact with television services and content.

The most exciting thing of an interactive TV is the ability to run applications that have been downloaded as part of the broadcast stream: this is really what makes the difference between a basic digital TV box and an interactive TV system. In order to support and enable interactive applications, the receiver is required to support not only the implementation of APIs needed to run the applications, but also the infrastructure needed to inform the receiver what applications are available and how to run them.

Interactive TV has drawn attention from researchers, organizations, and companies, and there have been a few efforts and solutions offered by them. Java TV and Multimedia Home Platform are the two most popular and vibrant techniques in this field [11], [13].

2.1.1 JAVA TV

Java TV is a Java-based software framework designed for supporting digital TV platforms from Sun Microsystems. A range of the common elements that are needed in a digital TV platform are combined into Java TV, such as access to service information, the core application lifecycle and model, and access to broadcast services (either through the Java Media Framework or through Java TV itself) [11].

Most importantly, Java TV is not bound to a certain set of digital TV standards. Java TV is explicit, pure, and independent. As a result, many solutions for digital TV can work equally well with Java TV, such as ATSC (the Advanced Television Systems Committee) solutions [1], or DVB-based systems [12]. This gives Java TV a very strong point that Java TV applications will not be tied to any specific broadcast system [11].

2.1.2. MULTIMEDIA HOME PLATFORM

Multimedia Home Platform (MHP) is an open standard middleware system introduced by the DVB Project for interactive and enhanced DTV [13].

Through MHP, the execution and reception of Java-based, interactive applications become possible on a DTV. Interactive TV applications can be transferred with video and audio streams over the broadcast channel. These applications can be, for instance, e-mail, games, information services, interactive voting, shopping or SMS.

A common interface among interactive applications and the terminals on which those applications run is defined by MHP. This interface helps different applications of a provider to be decoupled from particular software and hardware implementations of different MHP terminals. Moreover, the MHP extends the existing DVB open standards for interactive and broadcast services in several different broadcasting networks, such as terrestrial networks, satellite or cable networks. And MHP can be applied to various terminals, such as set-top boxes, multimedia PCs, and integrated DTV [13].

In summary, Java TV describes a set of digital TV concepts such as selecting a new service, accessing service information, and loading files from a carrousel rather than a normal file system. Most importantly, it does so in a way that is not tied to any specific digital TV standard. It is possible to perform Java TV on a DVB system or on an ATSC system. In fact, this is what happening currently with the spread of open standards such as Automated Content Access Protocol (ACAP), MHP, and OpenCable Application Platform (OCAP). While the MHP, on the other hand, is particular to the DVB series of standards[11].

2.2. SET OF EXPERIENCE KNOWLEDGE STRUCTURE (SOEKS) AND DECISIONAL DNA

The Set of Experience Knowledge Structure (SOEKS or shortly SOE) is a domain-independent, flexible and standard knowledge representation structure [3]. It has been developed to acquire and store formal decision events in an explicit way [4]. It is a model based upon available and existing knowledge, which must adapt to the decision event it is built from (i.e. it is a dynamic structure that depends on the information provided by a formal decision event) [6]; besides, it can be represented in XML or OWL as an ontology in order to make it transportable and shareable [3] [4].

SOEKS is composed of variables, functions, constraints and rules associated in a DNA shape permitting the integration of the Decisional DNA of an organization [6]. Variables normally implicate representing knowledge using an attribute-value language (i.e. by a vector of variables and values) [5], and they are the centre root of the structure and the starting point for the SOEKS. Functions represent relationships between a set of input variables and a dependent variable; moreover, functions can be applied for reasoning optimal states. Constraints are another way of associations among the variables. They are restrictions of the feasible solutions, limitations of possibilities in a decision event, and factors that restrict the performance of a system. Finally, rules are relationships between a consequence and a condition linked by the statements IF-THEN-ELSE. They are conditional relationships that control the universe of variables [6].

Additionally, SOEKS is designed similarly to DNA at some important features. First, the combination of the four components of the SOE gives uniqueness, just as the combination of four nucleotides of DNA does. Secondly, the elements of SOEKS are connected with each other in order to imitate a gene, and each SOE can be classified, and acts like a gene in DNA [6]. As the gene produces phenotypes, the SOE brings values of decisions according to the combined elements. Then a decisional chromosome storing decisional "strategies" for a category is formed by a group of SOE of the same category. Finally, a diverse group of SOE chromosomes comprise what is called the Decisional DNA [4].

In short, as a domain-independent, flexible and standard knowledge representation structure, SOEKS and Decisional DNA provide an ideal approach which can not only be very easily applied to various embedded systems (domain-independent), but also enable standard knowledge communication and sharing among these embedded systems.

3. DECISIONAL DNA DIGITAL TV

In order to capture, reuse, and share viewers' TV watching experiences, we applied the novel knowledge representation structure – Decisional DNA, to digital TV, called The Decisional DNA Digital TV (DDNA DTV).

DDNA DTV consists of the User Interface, the System I/O, the Integrator, the Prognoser, the Convertor and the Decisional DNA Repository (see Fig. 1).

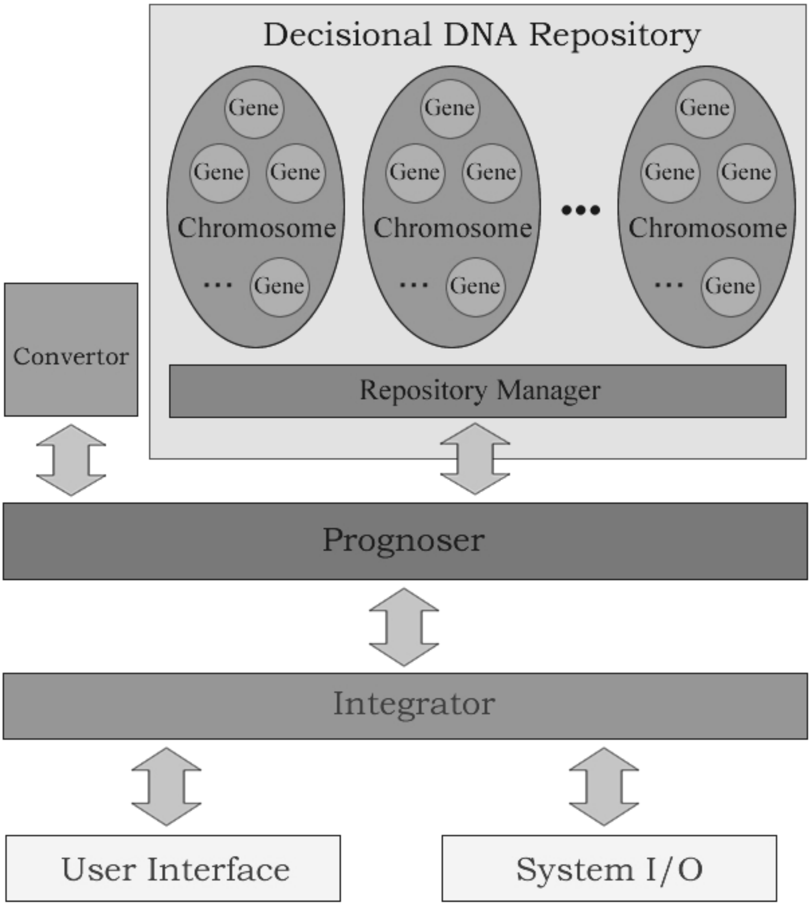


Fig. 1. System architecture for Decisional DNA DTV

• *User Interface:* The User Interface is developed to interact with the user/viewer. In particular, user can control, set and configure the system by using the user interface. Like the user can use remote control to select services, give feedback to a movie and interact with the service provider through the User Interface.

• *System I/O:* The System I/O allows our Decisional DNA approach to communicate with its domain. The System I/O tells the DTV which service is selected, for example, what movie should play, what feedback was given. Also, it reads the media stream, feedback, system time, service information from its domain.

• *Integrator:* The Integrator is the place where the scenario data is gathered and organized. In our case, we link each experience with a certain scenario describing the circumstance under which experience is acquired, such as the system time, the name of a selected service, user input and other service information. The Integrator organizes the scenario data and send them to the Prognoser for further processing.

• *Prognoser*: The Prognoser is in charge of sorting, analyzing, organizing, creating and retrieving experience. It sorts data received from the Integrator, and then, it analyzes and organizes the data according to the system configuration. Finally, it interacts with the Decisional DNA Repository and the XML Parser in order to store and reuse experience depending on the purpose of different tasks.

• *Convertor*: The Convertor translates knowledge statements generated by the Prognoser into the Decisional DNA experience structure; and interprets the retrieved Decisional DNA experiences for reusing. In this case, the Decisional DNA experiences are represented in XML format.

• *Decisional DNA Repository*: The Decisional DNA Repository is the core software component for our approach. It is the place where experiences are stored and managed. It is composed of the Repository Manager and Chromosomes:

a) *Repository Manager*: The Repository Manager is the interface of the Decisional DNA Repository. It answers operation commands sent by the Prognoser and manages the Chromosomes.

b)*Chromosomes*: The Chromosome is the place where a same category of Decisional DNA genes are gathered and stored. It is used to capture the decisional "strategies" for a category. A Decisional DNA gene carries a single decisional event (i.e. a SOE) [2]; and it can be represented by a set of XML tags described in [10] to store Decisional DNA in XML files.

4. EXPERIMENTAL TESTING OF THE CONCEPT

For testing the Decisional DNA applied to DTV, we used the Java TV SDK with NetBeans 6.8 on a DELL Latitude ES400 laptop. At this stage, the main purpose of our experiments is to prove that the Decisional DNA can work with Java TV, and our approach can provide its domain with the ability of experience capturing and reusing. Thus, we assume that there are only five types of movies, namely action, adventure, animation, comedy, and crime. And each movie is represented by its type plus an ID number, like Action1, Comedy2; there are 20 movies for each type.

We simulated a viewer watching movies on the DDNA DTV. Fig. 2 shows a screenshot of the viewer's TV. As we can see, the viewer's screen is composed by five components: Service Name which shows “Movies” here, Service Information which displays introduction of a selected movie, Ranking, Movie Showcase which shows movies recommended by the system, and “Show More…” button, by which viewer can access additional movies. At the beginning, DDNA DTV recommends two movies from each movie type. Once the system gets enough experiences, it will recommend movies according to those experiences.

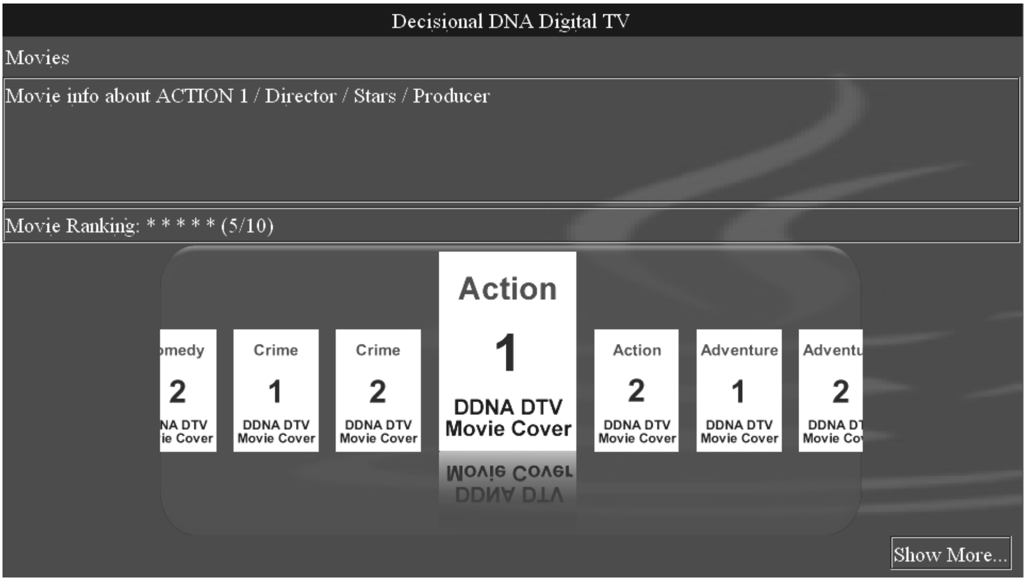


Fig. 2. Screenshot of DDNA DTV

We capture viewer's watching experience by recording seven variables: Movie Name, Director, Watch Date, Watch Time, Ranking, Type, and Viewer. Movie Name and Director are used to indicate which movie the viewer watched. Watch Day and Watch Time store date and time when the movie is watched. Ranking shows how the viewer likes the movie. Type illustrates what kind of movie it is. Viewer saves the name of user. Those variables are gathered and organized by the Integrator and then send to the Prognoser; finally, they are stored as a SOEKS in XML format [10]. Once the system have more than ten SOEKS (this number can be set in the system), it begins to analyze viewer's watching preference, and gives the viewer better recommendations according to analyzed settings.

For example, we assume that there is a viewer, Tom, who likes to watch action movies on every Saturday night as shown in Table 1.

Table 1. Tom's movie watching records

Movie Name	Watch Date	Watch Time	Ranking	Type
Action1	8/01/2011	19:35	7	Action
Action2	22/01/2011	20:02	9	Action
Action3	29/01/2011	20:13	8.5	Action
Action4	12/02/2011	19:42	8.7	Action
Action5	19/02/2011	21:07	8.6	Action

When the Prognoser recommends new movies to the user, it retrieves those stored watching experiences from the Decisional DNA Repository, and analyzes those experiences according to the user's settings. In this experiment, we analyze the movie types the user watched, and what day in a week the user usually watches them.

Equation (1) demonstrates how the system calculates the amount of a movie type should take in the newly recommended movie list.

$$N = (T \times 100 / D + 5) / \tag{1}$$

N represents how many movies should be recommended from a specific movie type; T represents how many movies of a specific movie type have been watched on a specific week day; D represents the total amount of watched movies on that week day. For example, Tom watched 11 movies in total on Saturdays, and 5 of those movies are action movies. Therefore, there should be five action movies in the next recommendation list : $(5 \times 100 \ / 11 + 5) / 10 = 5$.

As we assumed, during a few weeks of capturing experience, the system learns and knows that Tom watched 5 action movies, 2 adventure movies, 1 animation movie, 2 comedy movies, and 1 crime movie on Saturdays so far. As a result, the system will recommend 5 action movies, 2 adventure movies, 1 animation movie, 2 comedy movies, and 1 crime movie for him on next Saturday. Fig. 3 shows a screenshot of a newly recommended movie list for Tom.

Fig. 3. Newly recommended movie list for Tom

5. CONCLUSIONS AND FUTURE WORK

In this work, we introduce the concept, features, and architecture of the Decisional DNA DTV. Also, initial experiments we did on a DELL laptop with Java TV SDK are presented. As the result shows, the DDNA DTV can work under the Java TV environment, and it enables its domain to capture and store viewers' TV watching experiences; finally, those captured experiences are used in its future tasks to serve viewers better.

Since the DDNA DTV research is at its early stage, there are further research and refinement to be done, some of them are:

– Enhancement of the efficiency of Decisional DNA Repository storage and query.

– Further development of the user login system.

– Refinement and further development of algorithm using in the Prognoser.

– Implement better ways to interpret the user experience such as fuzzy logic.

REFERENCES

[1] ATSC, *the Advanced Television Systems Committee,* Available from: http://www.atsc.org/cms/.

[2] SANIN C., SZCZERBICKI E., *Experience-based Knowledge Representation SOEKS*, Cybernetics and Systems, Vol. 40, No. 2, 2009, 99–122.

[3] SANIN C., SZCZERBICKI E., *Extending Set of Experience Knowledge Structure into a Transportable Language Extensible Markup Language*, International Journal of Cybernetics and Systems, Vol. 37, No. 2-3, 2006, 97–117.

[4] SANIN C., SZCZERBICKI E., *An OWL ontology of Set of Experience Knowledge Structure,* Journal of Universal Computer Science, Vol. 13, No. 2, 2007, 209–223.

[5] LLOYD, J.W., *Logic for Learning: Learning Comprehensible Theories from Structure Data*, Springer, Berlin, 2003.

[6] SANIN C., MANCILLA-AMAYA L., SZCZERBICKI E., and CAYFORD Howell P., *Application of a Multi-domain Knowledge Structure: The Decisional DNA*, Intel. Sys. for Know. Management, SCI 252, 2009, 65–86.

[7] DVB, *The Digital Video Broadcasting Project*, http://www.dvb.org/

[8] SCHWALB E., *iTV Handbook: Technologies & Standards*, Prentice Hall PTR, ACM Computers in Entertainment, Vol. 2, Number 2, Article 7, 2004.

[9] Google, *Google TV*, Available from: http://www.google.com/tv/

[10] MALDONADO SANIN C.A., *Smart Knowledge Management System*, PhD Thesis, Faculty of Engineering and Built Environment – School of Mechanical Engineering, University of Newcastle, E. SZCZERBICKI, Doctor of Philosophy Degree, Newcastle, 2007.

[11] TV Without Borders, *Java TV Tutorial*, Available from: http://www.interactivetvweb.org/tutorials/javatv/

[12] REIMERS U. H., *DVB-The Family of International standards for Digital Video broadcasting*, Proceedings of the IEEE, Vol. 94, Issue 1, 2006, 173–182.

[13] VRBA V., CVRK L., and SYKORA M., *Framework for Digital TV Applications*, Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning, 2006, 184–-188.

[14] World DMB, *Digital Multimedia Broadcasting*, Available from: http://www.worlddab.org/

[15] WU Y., HIRAKAWA S., REIMERS U., and WHITAKER J., *Overview of Digital Television Development Worldwide*, Proceedings of IEEE, Vol. 94, No. 1, 2006, 8–21.

[16] Wikipedia, *Digital Television*, Available from: http://en.wikipedia.org/wiki/Digital_television/

[17] Wikipedia, *Digital Terrestrial Television*, Available from: http://en.wikipedia.org/wiki/Digital_terrestrial_television/

[18] IEEE, *Multichannel Multipoint Distribution Service*, Available from: http://grouper.ieee.org/groups/802/16/

[19] YARALI A., CHERRY A., *Internet Protocol Television (IPTV)*, TENCON IEEE Region 10, 2005, 1–6.

Knowledge management,
Business Process Management ,
Knowledge Integration, Decision support

Bartosz KUCHARSKI*, Edward SZCZERBICKI*

ADDING KNOWLEDGE MANAGEMENT PERSPECTIVE TO BUSINESS PROCESS MANAGEMENT

Business Process Management (BPM) as a holistic management is one of the most promising approaches to management in general, but there is still a question of how can we address knowledge management issues in this approach. This work is trying to add knowledge management perspective to all five main BPM activities: identifying, mapping, measuring, analyzing and redesigning the processes in an organization. Knowledge Management and Business Process Management have both similarities and differences, but in our opinion they should be used together to help with the conversion into self-learning organizations in knowledge based society.

1. BUSINESS PROCESS MANAGEMENT

1.1. INTRODUCTION

Business Process Management (BPM) has emerged as a comprehensive consolidation of disciplines sharing the belief that a process-centered approach leads to substantial improvements in both performance and compliance of a system. Apart from productivity gains, BPM has the power to innovate and continuously transform businesses and entire cross-organizational value chains. The paradigm of “process thinking” is by no means an invention of the last two decades but had already been postulated by early economists such as Adam Smith or engineers such as Frederick Taylor [1].

* Gdansk University of Technology, Gdansk, Poland.

The idea of "process thinking" is an old one and has been evaluated over time from work simplification paradigms such as Quality Control, Six Sigma, and Lean methodologies. Part of its origins is also related to Business Management and Information Technology (Figure 1).

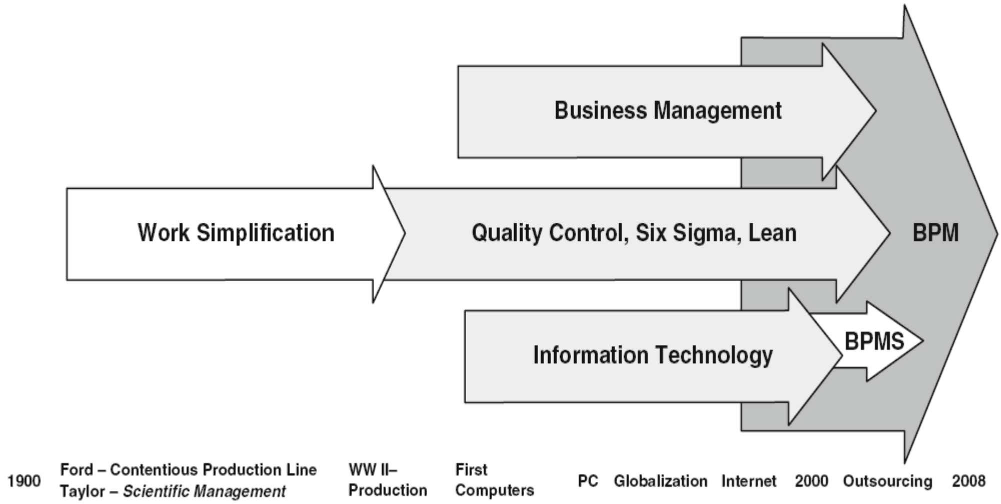


Fig. 1. The Evolution of "process thinking"[2]

The "process thinking" approach became very important in the following three main areas. First, in project management [3]. Second, in Total Quality Management (TQM), SixSigm and Lean approaches. And third, in Information Technology (IT) related areas such as Service Oriented Architecture, Enterprise Service Bus, and the Process Engine.

1.2. MAIN CONCEPTS OF BPM

The most important concept in BPM is a process. The main part of all process definitions in literature is that there is a chain of activities or transformations and it is about the dynamics of how things are done or happen. There is no commitment for that the process must be somehow defined. A business process corresponds to business domain, but depending on the approach the definitions vary.

Some popular process definitions can be provided as follows:

1. The business process or business method is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers [4]
2. The business process is a series of steps designed to produce a product or service. Most processes (...) are cross-functional, spanning the 'white space' between the boxes on the organization chart. Some processes result in a product or

service that is received by an organization's external customer. We call these primary processes. Other processes produce products that are invisible to the external customer but essential to the effective management of the business. We call these support processes [5].

3. The business process is a series of activities occurring within a company that lead to a specific end. Most often, the business process focuses on meeting the needs of the customer and delivering a good or service that will fulfill that need. In many cases, the business process is actually a collection of interrelated processes that function in a logical sequence to achieve the ultimate goal [6].

In other words, business processes refers to all processes in organization directly or indirectly, it crosses an organization in many ways, but still has some sequence of activities and interations with other processes.

Business Process Modeling is a way of describing business processes. This approach assumes a formal notation for describing processes in such way that processes may be analyzed and improved. The history of modelling techniques is quite long starting from Gantt charts around 1899 through flow charts in the 1920s, Functional Flow Block Diagram and Program Evaluation and Review Technique (PERT) in 1957, Data Flow Diagrams and Integration Definition (IDEF) in the 1970s [7]. The modern methods are Unified Modeling Language(UML) and Business Process Modeling Notation(BPMN). The UML comes from software development domain, and BPMN is a notation for business analysts. The important differentiator is that UML can be used for process approach and that BPMN is dedicated to such approach.

Business process management is a field of management focused on aligning organizations with the needs of clients. It is a holistic management approach that promotes business effectiveness and efficiency while striving for innovation, flexibility and integration with technology [8]. In other words it is a business management approach that focuses on business process with information technologies matters paradigm. Integrated with IT tools and technologies, BPM is the enabler of implementation, measure, execution and simulation the business processes and could be considered as main differentiator from SixSigma and Lean approaches. The other perspectives are incorporated from former “process thinking” approaches.

2. KNOWLEDGE MANAGEMENT

2.1. INTRODUCTION

Knowledge Management (KM) comprises a range of strategies and practices used in an organization to identify, create, represent, distribute, and enable adoption of insights and experiences. Such insights and experiences comprise knowledge, either embodied in individuals or embedded in organizational processes or practice [9]. The range of KM strategies and practices is very wide and interdisciplinary.

2.2. KNOWLEDGE MANAGEMENT VS COMPETENCE MANAGEMENT

Knowledge is regarded as a key asset in modern economy, so it is a one of the core competences that gives sustainable competitive advantage to an organization. The difference between KM and competence management is that the core competences could also include material or financial assets. The relations between those two is illustrated in Figure 2.

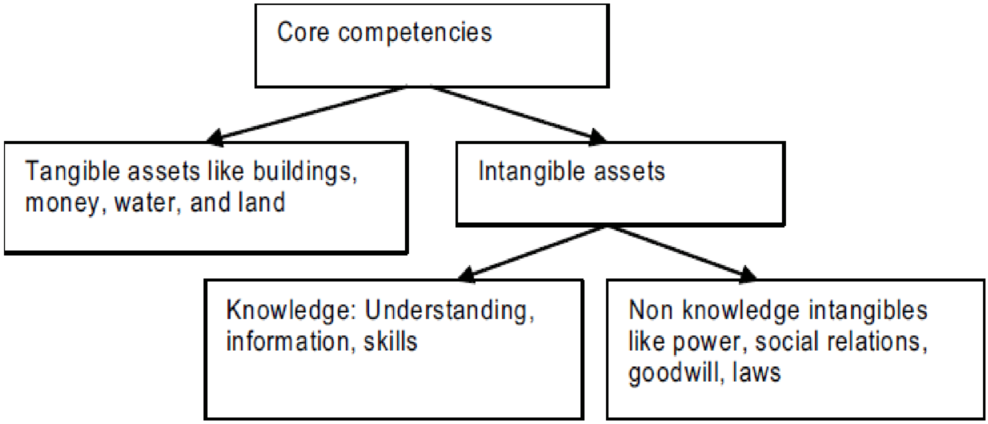


Fig. 2. Relations between core competencies and knowledge [10]

2.3. TACIT, LATENT AND EXPLICITE KNOWLEDGE

The pioneers of KM like Nonaka and Takeuchi divided the knowledge into two categories depending on whether it can, or can not be expressed. Tacit knowledge cannot be codified or learned without long term experience because of its personal nature. Explicit knowledge can be written down or spoken. The term latent knowledge is related to knowledge that could be externalized, but not without attenuation. The difficulties to express this knowledge without attenuation usually stem from the fact that this knowledge resides in subconsciousness [10].

2.4. KNOWLEDGE INTEGRATION

Processes for identification, acquisition and internal utilization of external knowledge are called Knowledge Integration (KI). This term describes how an organization learns from the environment, especially how enterprise interacts with knowledge providers.

2.5. BUSINESS MANAGEMENT IN KNOWLEDGE MANAGEMENT

KM as an approach to generic management in all businesses has to address all management activates like planning, organizing staffing, leading and controlling. The goal is remains the same to accomplish a goal in a efficient way by using available resources. The efficient way means an optimum and proper choices between available options. The choice usually has its consequences, because consumes resources and is in conflict with other options. An answer to this problem from KM approach is called tension control that enables finding the balance between opposite forces.

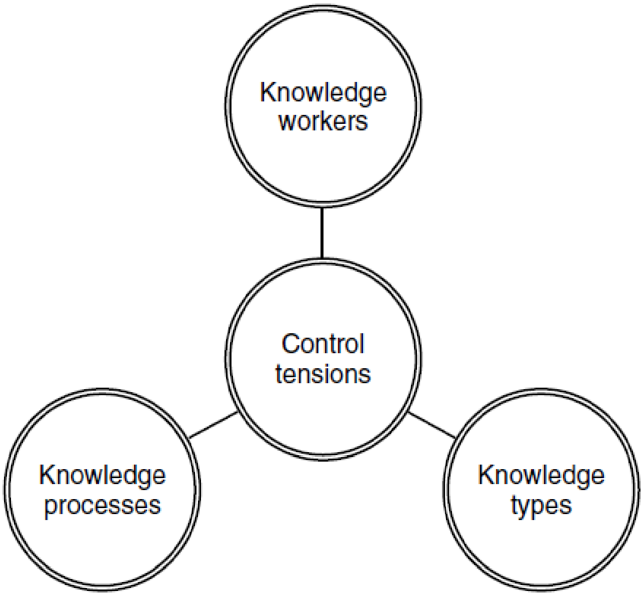


Fig. 3. Knowledge management control tensions [11]

3. COMPARISON OF BPM AND KM

Both BPM and KM are hot management approaches and can be seen as comprehensive solutions for gaining sustainable competitive advantage. KM is aiming at innovative approaches to be one step before the competitors by building an organization based on knowledge that will enable to adopt to changing environment and through innovations introduce a fast track for new product or service development. BPM is about optimization of existing processes and enable for higher quality services and products then competitors can at the same effort. This is the very first obstacle on high principle level, in other words we have new product or service development versus business continuity dilemma. At presence the change is unavoidable and must be addressed not only from performance perspective, but also from knowledge perspective. The combination of the two approaches may be the answer the dilemma.

3.1. SIMILARITES BETWEEN BPM AND KM

The first observation is that those two approaches overlap in process area both ways. Managing processes and their continuous improvement is a primary principle of BPM and processes are at the same one of key knowledge assets because include a great deal of explicit knowledge how things are done in an enterprise. The process approach is one of the ways of describing business and not only from static organizational perspective, but also the dynamics. Changing an existing process in an organization in BPM’s terminology is a process improvement it KM terms it is a “combination” the transition for explicit knowledge to explicit knowledge. From processes inputs and output perspective there are dedicated processes for KM especially in KI area like cooperation with consultancy agencies, courses provides or research centers. Those processes must be measured and the results valued from business perspective. The will of continues improvement itself is a common thing for KM and BPM. Both assumes employee development including trainings and education to achieve required level of competence. Both use valuating employees and intellectual capital to make aware decisions. The empowerment and involvement of all personnel is also seen as necessary element for successful introduction any of those approaches. The another common thing is teambuilding and collaboration to achieve best results. Both recognizes the need of gathering knowledge about the market, especially in BPM about customers and competitors. Both try to optimize decision making processes and decisions to achieve better quality and efficiency. Finally both take hope in information technology as an enabler to achieve their goals, some techniques as exploring the data are the same. The information how things are done is necessary for both in knowledge sharing or gathering information about processes.

3.2. DIFFERENCES BETWEEN BPM AND KM

Despite of many similarities there are significant differences in focus and strategy between KM and BPM. KM aim to embed knowledge in personnel, customer, processes, products and services whilst BPM tries to achieve better effectiveness and efficiency through better use of available resources. KM regards knowledge as source of competitive advantage whilst BPM is striving for excellence through benchmarking, etc. KM finds in knowledge a way of improving productivity whilst BPM sets high standards of performance in all areas within an organization. BPM is strictly focused on measurable results and on customer. KM focus on knowledge including those which cannot be even properly externalized like tacit and latent knowledge. KM aims to create or disseminate new knowledge and embedding it in new technologies and products. BPM finds effective leadership through commitment for focus on results and customer measured by qualitative and quantitative measurable data, the management basis on facts and processes, mutual respect, trust and benefits of all stakeholders. KM will always search for new sources of knowledge and information and try to adopt knowledge to the market. BPM will stick to explicate knowledge gathered from customers and process execution data providers.

4. COMBINING BPM AND KM

Both approaches seems to be adequate, but how we can use them both on the same time. The answer is to find a compromise in places where they differ and introduce new elements which were not addressed. The most important is compromise on focus, the knowledge versus the results. Of course we can still say, that knowledge should be valued and treated as an output and still manage by fact basing purely on BPM paradigms, but that will be far from true knowledge management. There are many reasons for such strong statement. The first is that apart explicit knowledge there is tacit and latent knowledge that cannot be even properly expressed thus methods for valuing such knowledge are far from meeting BPM standards for measurement. The second is that BPM assumes processes every ware and again that is only partially true for KM. Some processes for KM can be defined on level required by BPM to control their execution, we can formalize cooperation with consultancy agencies, research centre, some processes for access control to knowledge assets, even processes for publishing, registering, altering knowledge assets, etc. The problem is some crucial element within knowledge management cannot be put in frame of defined process i.e. the discovery of knowledge, the combination of existing knowledge, applying existing knowledge on new field, etc. Some KM techniques like brainstorming are by definition free in form at certain stages. Even if we put those activities in frames of defined process the clue will remain undefined there will be no instruction how to came up with a new idea. The third reason is that KM will search for knowledge and preserve it even there is no strict relation to currently executed processes, from BPM perspective it is a waste of resources and reduced as activity not adding value in the chain.

The only option for BPM is to verify the approach and focus on knowledge and results. The knowledge in meaning much wider then customer or market, but also interdisciplinary and not directly related to existing processes within an organization. The second thing is to leave some space for non-processes managed aspects of knowledge management.

4.1. APPLYING BPM TO KM

BPM goes with quite excessive toolset for managing processes, especially software that could be easily applied to KM processes. The processes must be defined in some formal notation like BPMN and executed on some platform that allows human interaction. Such software is called workflow or Business Process Management System or Suit (BPMS) with human interactions. Most of them enable designing graphical user interface and assign task to specific groups of users usually called roles. The roles are mapped to swim-lines which are containers for activities in process definition. The platform allows for long time user sessions and particular flow of actions (the instance of process) is usually called a case.

The benefits for such approach will be:

- Common workplace platform with the to-do list for each person and user group.
- Support for teamwork which includes locking the work item, redirecting the task in case of leaves, etc.
- Adding numbers to KM processes, some common measurements from BPM should be applied i.e. response time, execution time, personal task queue length.
- Enabling task prioritization, that will support the work organization.
- Monitoring the cases in statistical terms.
- Support for escalation root for prolonging cases.

4.2. APPLYING KM TO BPM

Adding knowledge aspect to BPM, will force some deeper changes. First there must be agreement on putting additional effort into processes execution. The balance between knowledge orientation and performance on operational level must be found. The reasonable option is that, we agree on effort that will not add any human required actions on execution level, but only increase requirements for infrastructure capacity and require additional supportive processes for knowledge management. Such approach will enable to apply knowledge management do decisions and experience preservation during process execution. There are even possible improvements in processes due to applying a decision support that is based on experience. Such approach may use Set of Experience Knowledge Structure(SOEKS)[12] for storing decision events and using this in case based reasoning decision support[13]. The implementation for knowledge management support will only require adding a not human actions that will preserve the data available at decision point and scoring the decision in a supportive process. When we got data and the final score we can measure the distance in a new case and suggest the answer. The another way is to discover rules by KM processes and apply those rules using a rule engine software. It is a way of putting rule based knowledge repository to work in operational processes. This approach will able to externalize the embedded in process logic to a common repository, that will be good not only for KM purposes but also for enabling more complex and flexible management.

KM may also support BPM with managing the knowledge artifact like process definitions, by putting them into wider context that could include organizational aspects and adding knowledge references between processes. In this case process definition will become more then a map of interactions for passing the work, but also contains known knowledge effects like gaining certain kind of knowledge including tacit and latent knowledge of performers. Other example of adding references to process definitions is mapping of possessed intellectual capital on processes for better development and protection.

KM treats knowledge as a source of sustainable competitive advantage. The knowledge leads to products and services that allow to be step ahead the competitors, to have good relations with customers. This is the most significant benefit of introducing KM, in this case BPM will take the execution part. Of course BPM must be focused on the results and knowledge.

5. SUMMARY

Knowledge Management and Business Process Management can go together, but some adjustments are needed for both. Knowledge management as less formalized approach will have to put some of their activities into defined processes, and agree on higher level of measurement, but in return receive a toolset and gain from BPMS access to process execution. The BPM must leave the orthodox LEAN approach and make some room for knowledge activities as it must obey the law and compliance regulations. The combination of both can be beneficial in many ways, including performance, business continuality, and intellectual capital.

REFERENCES

[1] BROCKE J. ROSEMANN M., *Foreword*, In: Handbook on Business Process Management 1, BROCKE J. ROSEMANN M. (Eds), Berlin, Springer-Verlag, 2010, 4.

[2] HARMON P., *The Scope and Evolution of Business Process Management* In: Handbook on Business Process Management 1, BROCKE J. ROSEMANN M. (Eds), Berlin, Springer-Verlag, 2010, 38.

[3] *A Guide to the project management body of knowledge (PMBOK Guide) – fourth Edition*, Newtown Square, Project Management Institute, 2008, 37–69.

[4] *Business process* – Wikipedia, the free encyclopedia, *http://en.wikipedia.org/wiki/Business_process,* 10.07.2011.

[5] RUMMLER G. A., BRACHE A. P., *Improving Performance: How to Manage the White Space in the Organization Chart, 2nd Edition*, San Francisco, Jossey–Bass, 1995.

[6] What is a Business Process ?, *http://www.wisegeek.com/what-is-a-business-process.htm*, 07-04-2011.

[7] Business process modeling, *http://en.wikipedia.org/wiki/Business_process_modeling*, 10-04-2011.

[8] Business process management, *http://en.wikipedia.org/wiki/Business_process_management,* 22-07-2011.

[9] *Knowledge Management*, *http://en.wikipedia.org/wiki/Knowledge_management*, 10-04-2011.

[10] WIJNHOVEN F., *Knowledge Management: More than a Buzzword,* Knowledge Integration The practice of Knowledge Management in small and medium enterprises, Berlin, Springer-Verlag, 2006, 4–8.

[11] DESOUZA K.C., AWAZU Y., *Engaged Knowledge Management*, Basingstroke, Palgrave Macmillan, 2005, 10.

[12] SANIN C., SZCZERBICKI E., *Experience-Based Knowledge Representation SOEKS,* Cybernetics and Systems: An International Journal, Taylor and Francis, Vol. 40(2), 2009, 99–122.

[13] KUCHARSKI B., SZCZERBICKI E., *Knowledge Management – Decision support in a workflow system*, In: Information systems architecture and technology: IT Models in Management Processes, Wilimowska Z.(Eds), Wroclaw, Oficyna Wydawnicza Politechniki Wroclawskiej, 2010, 117–128.

integrating models, information management,
information integration strategies

Dariusz PUT*

STRATEGIES AND SYSTEMS OF INFORMATION RESOURCES INTEGRATION

In this chapter the issue of information resources integration is discussed. Firstly, the problem of organisational integration as well as various solutions are described. Four levels of integration are defined, existing strategies are discussed and some examples of models and systems integrating various information resources are mentioned. Next, the proposed Heterogeneous Information Resources Integrating Model (HIRIM) is characterised: its architecture, possible versions and properties. We prove that designing such a model or system it is necessary to compromise which leads to the solution never meeting all expected needs. Finally, some conclusions are discussed.

1. INTRODUCTION

A considerable number of various factors contribute to the complexity of the problem of information resources integration. They are: variety of data and information structures, a great many systems and query languages, diversity of data models, the necessity to integrate information collected by one department but also the whole organisation, a few cooperating organisations or the whole sector. There is a variety of users' information needs including foreseeable (identified during the process of the system designing) as well as unforeseeable (formulated *ad hoc* during the system exploitation). There are three main tasks executed by integrating systems: the communication with users, storing and making accessible information resources and the accomplishment of integration tasks. There have been a considerable number of solutions elaborated as well as systems and models dedicated to fulfilling the task of integration. It is necessary to search for solutions that make possible to integrate various information resources and enable access to heterogeneous, dispersed data and

* Cracow University of Economics, 27 Rakowicka Str., 31-510 Krakow, Poland.

information. The scope of this chapter is to describe the architecture of proposed information integration model dedicated to integrating various information resources stored in independent systems and to discuss its core properties.

2. INFORMATION RESOURCES INTEGRATION STRATEGIES AND SYSTEMS

Organisational integration covers a considerable number of components: database systems, information systems, analysis and design of enterprises, agent systems, processes co-ordination, organisation theories and modelling methods. Such multidimensional perception of the problem constitutes a substantial obstacle in preparing unified solution which might be applied under any conditions and in every organisation. So, a term *integration*, partly because of the extensiveness of the issue, partly because the problem is relatively new, is not unambiguous and commonly identically understood. There are researchers that focus mainly on technological aspects of integration, paying attention to databases and data schemas linking (see e.g. [1, 6, 7, 16]). However, there are a considerable number of approaches, where integration is perceived as a multidimensional issue, covering three or four levels (see tab. 1): processes, applications, data and systems [3, 5, 8, 9, 10, 11, 14, 15, 17].

Table 1. The levels of integration

Processes	Applications	Data	Systems	
			technological view	multidimensional view
co-ordination and automation of processes taking place in organisations and fulfilled together with business partners	communication between various applications	exchange of data, information and plain files	communication between systems, connections and exchange of data with the use of computer networks	integration of organisation on all levels: processes, applications and information resources

The possibility of effective using of information resources depends on efficient management of the resources which is a challenge for contemporary organisations. Information resources management is the process that does not finishes when a solution is elaborated and employed. The process covers the necessity of standardisation of possessed resources, because the aim of integration is to design a system in which data and information are identically understood by all users. During preparation of integration solutions various approaches are used, including conceptual schemas designing, exchangeable information dictionaries creation and definition of attributes elaboration. In this chapter information resources integration is understood as all activities leading to the elaboration of a solution that enables efficient management of information resources which means storing, identical understanding by users (employees, clients, applications) and the possibility of effective searching of necessary information and its processing. This definition covers four aspects of integration. Two of them (storing and identical understanding) constitute means to achieve aims which are the possibility of obtaining necessary information and its processing.

Taking under consideration the method of storing information resources criterion, two main information integration strategies have to be mentioned (fig. 1):

- design and implementation of one uniform shared database and migration of all data to newly created system,
- creation of some kind of interface above existing, heterogeneous information resources.

Considering the strategy based on dispersed resources it is assumed that there are various heterogeneous information systems in organisation, therefore, existing information management methods do not change and newly created systems are, as until now, employed locally and implemented by or for individual departments or employees. Such a solution is based on federation of independent databases or, widely, information resources repositories. In this case integration may be achieved by the mediation of dedicated schema which contains meta-information about dispersed resources being exchanged. Another such a solution involves linking co-operating systems peer-to-peer or by co-operating applications, and in more advanced solutions, creation of commonly accessible services (fig. 1). Users and applications searching necessary information access resources using predefined interfaces which may be of various forms: catalogues, registers, applications or services. Metadata catalogues enable access to information about shared data (databases, the Internet resources, documents, services). Catalogues are created in accordance with a schema designed for one or several co-operating organisations. One main catalogue is dedicated to combine partial catalogues and constitutes "catalogue of catalogues". Metadata registers contain metadata consistent with structures, models, dictionaries and schemas. Registers delivers to designers and architects methods for creating and decoding data and for co-sharing the environment inside organisations.

While designing the system for applications integration or SOA, the problem of information integration is often not taken into consideration. It is assumed that applications and services, fulfilling their tasks, use local or dispersed repositories and problems connected with employing information resources and their integration are considered by local systems administrators. It does not mean that the problem of integration is neglected – it is worked out on the level of local systems. Solutions more technologically advanced are also designed. They join together advantages of EAI (*Enterprise Application Integration*) and systems based on shared schema. Such integrating systems may be applied in one organisation as well as together by business partners.

An example of a system joining two strategies, one shared database and solution with unchanged local systems, is data warehouse. Central repository is designed for storing shared, mainly historic resources and local systems remain unchanged and they are dispersed. Resources collected in such a database are used for various analyses and local systems for current facts recording. The main purpose of designing data warehouse is to separate transactional resources from analytical ones, which may have an significant impact on the effectiveness of systems operation.

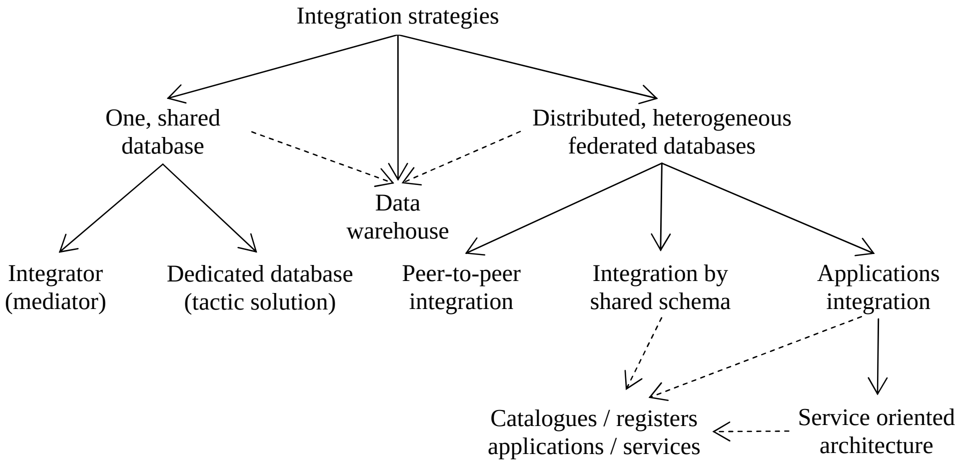


Fig. 1. Information resources integration strategies

There is a considerable number of models and systems proposed to integrate various data and information resources, e.g.: iMeMex [4], Lore [13], SEAL [12], TSIMMIS [2]. The features of an integration model vary according to the element that is perceived as the core. In some solutions the most important is expressiveness defined as a possibility to precise representation of designed area, in other solutions user friendliness, where users' information needs and their knowledge and skills in the field of query formulation are matters of prime importance. Existing solutions are dedicated to integrate various kinds of data and information. The iMeMex is the example of PIM (*Personal Information Management*) systems used by users for management of their personal information resources stored on their computers. It consists of three layers. The third one (*work performance and management*) is responsible for designing and implementation of customer solutions, creation of work environments as personalized and context-sensitive user interfaces available through portals. The TSIMMIS and SEAL projects are dedicated to integrating all possible kinds of information resources: plain files, structured and semistructured data. In the TSIMMIS project there is no special language designed for end users. However, there is a graphic user interface, but it only supports users in the process of queries formulation. The Lore is an example of a system designed especially to manage semistructured data.

The question is, how to manage data and information in organisation in order to facilitate data accessibility, store and deliver only correct and proper data, establish efficient (even automatic) management of an integrating system, create flexible system, which facilitates enclosing new as well as excluding or modifying existing information resources. There are so many options that even making a decision concerning only the choice of integration strategy is a complex task.

3. THE MODEL FOR INTEGRATION OF HETEROGENEOUS INFORMATION RESOURCES

The definition of information integrating model should comprise the description of its architecture, formal framework for the structure of its components and detailed solutions concerning methods of designing them and modifying during exploitation. The proposed HIRIM model consists of four layers (fig. 2):

- local information resources, which consists of information resources systems being integrated,
- local communication, which consists of translators, queries converters and, optionally, local schemas,
- global (shared), which consists of four modules: metadata, integrator (mediator), localizer and metainformation updater,
- user interface with dedicated query language.

As far as the architecture of HIRIM model is concerned the structure and content of information resources constituting the lowest layer are negligible. They may be freely changed during the system exploitation. Communication wrappers are defined separately for every sub-system constituting the integrating system. They consist of three modules: translators, queries converters and local schemas. The task of translators is to negotiate names and formats of data after receiving queries formulated by users as well as after execution of queries and before sending results to the mediators. The task of queries converters is to transform queries to the format understandable in individual local systems. The local schemas comprise metainformation about this subset of resources that are accessible via the integrating system. The global layer consists of four modules: metadata (global schema), localizer, mediator and metainformation updater. The metadata comprises information about all resources accessible via the system. The localizer stores information enabling identification of those sub-systems, to which queries formulated by users should be sent. The mediator is responsible for joining data and information obtained from local communication layer and sending them to users in response to their queries. Metainformation updater is an application that constantly monitors information resources and updates metadata if the structure of local schema is changed. The user interface layer assists users during the process of queries formulation. This component may be standardised (which is recommended) or created individually for or by separate users. It also contains a module being used for the presentation of information obtained in response to formulated queries.

As far as the quality of the system is concerned the most important part of the model is the global schema. With a view to the fact that shared information comes from a variety of systems, where the structure of data is heterogeneous und unpredictable, while designing this layer it has to be taken under consideration the variety of forms of data sources. In addition, the system has to be ready to response to diversity of actors needs, including services, predefined and undefined users and communication agents. What is more, this layer must be possibly flexible and possibly automatically modified. The metadata has to comprise actual information, so it has to be modified on line. Its structure has an enormous impact to the effectiveness of the whole system. Due to these varieties, the proposed HIRIM model may be designed in a range of versions, accordingly to its features concerning:

- the complexity of a query language and, as a consequence, the complexity of queries formulation,
- accuracy of data sent to users in response to queries formulated by them,
- flexibility (possibility of enclosing new or modifying existing information resources),
- expressiveness (ability to precisely represent facts and connections between them).

On the basis of accepted strategy, in the system based on the HIRIM the main emphasis may be put on flexibility of a solution, non-complicated structure of query language and possibly non-complicated structure of shared elements at the cost of the system expressiveness and the increase of the complexity of communication wrappers structures. Let us call such systems user oriented. If the main emphasis is put on designing a system as expressive as possible, which results in receiving precise answers for detailed queries (including the possibility to formulate aggregates, use aggregate functions or create calculations), the user query language has to be more complex and, as a consequence, the system becomes less flexible, because shared elements, used e.g. for queries formulations, have to have complex structure. Let us call such systems information oriented. In information oriented systems the structure of global schema is also complex. The fundamental issue is to establish rules that make possible designing of maximally expressive systems so as it will be possible to contain there all necessary metadata about shared resources together with connections between individual concepts. As far as the resources searching process is concerned, such systems are characterised by high effectiveness. Users receive precise information, there is no need to additional information processing. However, such a solution is possible only if users are familiar with a query language (which in this situation is complex) as well as they have access to metainformation about shared resources and are able to use it. An increase of the complexity of the query language leads to extending expectations as far as users knowledge, skills and abilities are concerned. As a consequence, the query language becomes complicated and complex queries may be formulated only by specialists. In user oriented systems only wrapper communication modules, being components of the local communication layer, have complex structures. However, such a solution is more user-friendly – the users do not have to possess abilities and knowledge concerning queries formulation.

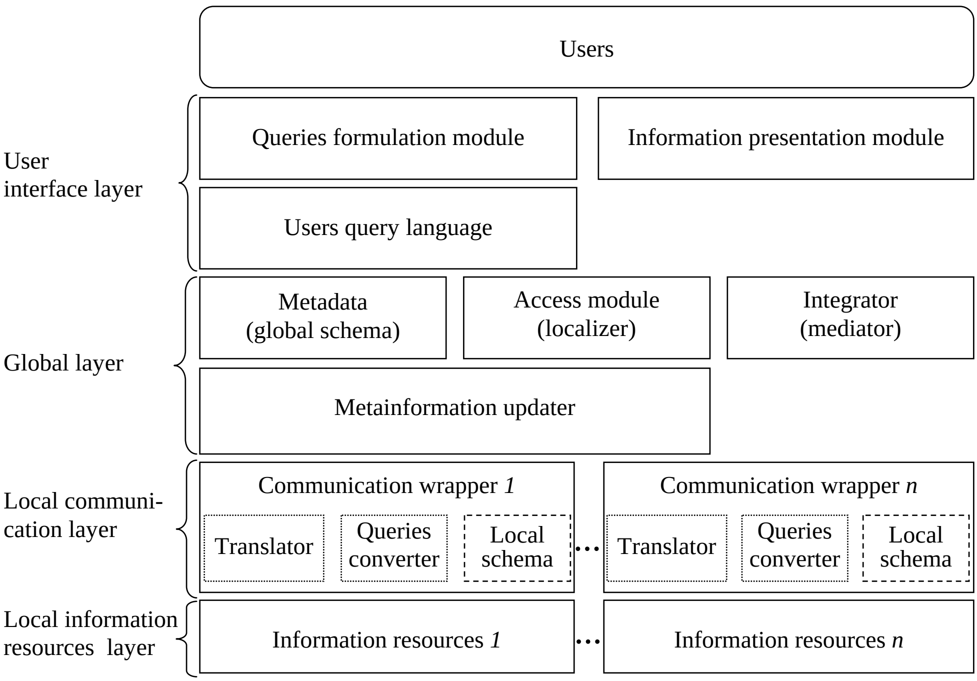


Fig. 2. The architecture of the HIRIM model

Information oriented systems have complex global schema (thick global schema module). In such models a key issue is the expressiveness of the global schema, so as all shared information as well as connections between individual pieces of information might be included. Such systems are characterised by high searching effectiveness. Users receive precisely the information they expect. But they have to have some knowledge including query language (which in this situation is complex) as well as shared information structure. Increasing complexity of query language leads to the increase of demands expected from users. As a consequence, a query language becomes so complicated that detailed queries may be formulated only by specialists. In user oriented systems a wrappers module is complex (thick wrappers module). Such systems are user-friendly, users do not have to have special knowledge or skills to formulate queries.

Table 2. Characteristics of information integrating systems

Minimum requirements as far as query formulation is concerned	
Positive characteristics	Negative characteristics
– non-complicated query language – simple structure of model shared elements – satisfactory flexibility	– received information not quite precise – small expressiveness – complex structure of communication wrappers

There are three components of the user layer: the interface enabling queries formulation, the established user query language and the application responsible for presentation of information resources received in response to users queries. From among three modules comprising the interface layer only one – the user query language – has to be uniform, globally defined. Two remaining modules may be standardised or designed locally for or by individual users.

Taking under consideration desirable characteristics of integrating models identified on the basis of existing solutions analysis, the proposed HIRIM model has following properties:

- the structure consists of four layers,
- every layer consists of modules fulfilling individual tasks that enable users to share information resources,
- sub-systems being integrated remain unchanged,
- the model may be used for designing of systems integrating only one category of information resources as well as all categories: hermetic, open (schema based and semistructural) and services,
- enables integration of any database management systems and other repositories,
- every piece of information is an instance described by attributes,
- instances represent concepts – information about instances is accessible for users and applications through concepts,
- the structure of concept depends on the category the concept belongs to: all concepts are described by attributes and information about their contents or, in the case of services, their tasks. In addition, services are characterised by information about input (necessary for proper operation of applications) and output data,
- metainformation about shared information resources is designed as ontology,
- the structure of the ontology is dynamically modified during a system based on the model exploitation, so it is not necessary to periodically actualise of sharing modules or even to re-design them,

- the ontology has a directed graph structure,
- a syntax of user query language is as simple as possible, so queries may be formulated even by inexperienced users and designing of translators reformulating queries created by users to query languages used in individual sub-systems is non-complicated, it is also easy to design graphic user interface assisting users in the process of queries formulation,
- requirement set to users concentrate more on the necessity to process data than on having knowledge about the structure of query language and the ability to formulate queries,
- queries execution is two-stage – the query language enables to choose information resources only party processed, they must be additionally transformed by users – this feature of the model is a consequence of the assumption that the user query language structure is as simple as possible,
- queries may be formulated in text or graphic forms,
- during the process of queries formulation it is necessary to have an access to the content of the ontology, which may be fulfilled by graphic interface, so the process of information resources choice is intuitive and user friendly,
- some of tasks fulfilled during the integration process are processed in the local layer which means that in sub-systems constituting the integrating system modules performing these tasks have to be designed,
- design, implementation and exploitation of the system are evolutionary,
- shared modules structures (global and local schemas, localizer) are designed on the basis of XML technology (e.g. RDF, OWL), so it will be possible to link a system designed on the basis of the model with other similar solutions,
- communication protocols have to be established for all modules, so as modules structures may be freely modified, because co-operation with other components will take place on the basis of exchanging announcements, metadata or information resources in established forms.

4. CONCLUSIONS AND FUTURE WORK

In this chapter, the problem of information resources integration was discussed. We tried to show variety of concepts, strategies and systems designed to integrate various kind of resources. Diversity and heterogeneity of information resources and proposed solutions constitute serious obstacle in elaborating solution that may be employed by any organisation. In the chapter the architecture of the HIRIM is described and tasks fulfilled by its components are discussed. The presented model may constitute a basis for designing systems integrating heterogeneous information resources stored in a variety of independent systems. These systems may be user oriented or information oriented. We tried to justify that there are tight connections between desirable characteristics of modules comprising any integration system, so while designing such a system one has to make some compromises. In the future we are planning to design framework for all modules constituting the HIRIM model.

REFERENCES

[1] ALSÈNE E., *The Computer Integration of the Enterprise*, IEEE Transactions on Engineering Management, 46 (1), 1999, 26–35.

[2] CHAWATHE S., GARCIA-MOLINA H., HAMMER J., IRELAND K., PAPAKONSTANTINOU Y., ULLMAN J., WIDOM J., *The TSIMMIS Project: Integration of Heterogeneous Information Sources*, Proceedings of IPSJ Conference, Tokyo, Japan, 1994, 7–18.

[3] CHERRINGTON J.O., DENNA E.L., ANDROS D. P., *Developing an Event-based Business Solution: The Case of IBM's National Employee Disbursement System*, Journal of Information Systems, 10 (1), 1996, 51–69.

[4] DITTRICH J.-P., VAZ SALLES M.A., *iDM: A Unified and Versatile Data Model for Personal Dataspace Management*, 32nd VLDB Conference, Seul, Korea, 2006, 367–378.

[5] GIACHETTI R.E., *A Framework to Review the Information Integration of the Enterprise*, International Journal of Production Research, Taylor & Francis Ltd., 42 (6), 2004, 1147–1166.

[6] HAAS L., *Beauty and the Neast: The Theory and Practice of Information Integration*, Database Theory – ICDT 2007, Lecture Notes in Computer Science, Springer, 4353, 2007, 28–43.

[7] HALEVY A.Y., RAJARAMAN A., ORDILLE. J.J., *Data Integration: The Teenage Years*, VLDB, 2006, 9–16.

[8] KALINICHENKO L.A., *Canonical Model Development Techniques Aimed at Semantic Interoperability in the Heterogeneous World of Information Modeling*, Proceedings of the CAiSE INTEROP Workshop on "Knowledge and Model Driven Information Systems Engineering for Networked Organizations", Riga Technical University, 2004, 101–116.

[9] KLAUS K., ROSEMANN M., GABLE G., *What is ERP?*, Information Systems Frontiers, 2 (2), 2000, 141–162.

[10] KOSANKE K., VERNADAT F., ZELM M., *Cimosa: Enterprise Engineering and Integration*, Computers and Industry, 40 (2–3), 1999, 83–97.

[11] MARKUS M.L., TANIS C., *The Enterprise Systems Experience – From Adoption to Success. Framing the Domains of IT Management: Projecting the Future Through the Past*, Pinnaflex Educational Resources, Cincinnati, 2000, 173–207.

[12] MAEDCHE A., STAAB S., STUDER R., SURE Y., VOLZ R., *SEAL – Tying up Information Integration and Web Site Management by Ontologies*, Technical Report, Institute AIFB, University of Karlsruhe, Germany, 2002.

[13] McHUGH J., ABITEBOUL S., GOLDMAN R., QUASS D., WIDOM J., *Lore: A Database Management System for Semistructured Data*, SIGMOD Record, 26 (3), 1997, 54–66.

[14] MEJABI O.O., SINGH N., *A Framework for Enterprise-wide Integration*, Computer Integrated Manufacturing, 10 (1–4), 1997, 212–220.

[15] SMITH M., *Time to Govern Your Data*. Intelligent Enterprise, 9 (2), 2006.

[16] TRUMAN G.E., *A Discrepancy-based Measurement Approach for Data Integration*, Journal of Organizational Computing and Electronic Commerce, 8 (3), 1998, 169–193.

[17] VOLKOFF O., STRONG D. M., ELMES M. B., *Understanding Enterprise System-enabled Integration*, European Journal of Information Systems, 14, 2005, 110–120.

natural language processing, text analysis, knowledge extraction, machine learning

Dariusz BANASIAK*, Jarosław MIERZWA*, Antoni STERNA*

EXTRACTION OF ELEMENTARY FACTS FROM NATURAL LANGUAGE TEXTS

The chapter presents the problem of extraction of elementary facts from complex sentences in natural language. In learning stage the system is processing complex sentence and corresponding set of simple sentences, reflecting elementary facts expressed in complex sentence. Link Grammar is used for syntactic analysis. Obtained syntactic structures are supplemented with semantic features and thus constitute syntactic-semantic images. During process of image matching, patterns of elementary facts are identified within the image of complex sentence and stored in knowledge base. In working stage, similar image of the new, analyzed sentence is created. This image is matched with patterns of already known elementary facts taken from knowledge base. Successful matching means that new elementary fact was discovered. Some aspects of syntactic and semantic analysis are discussed, e.g. the problem of selection from many alternative syntactic structures generated by Link Grammar.

1. INTRODUCTION

At present significant parts of human knowledge are digitalized and easily accessible. However, human capabilities to process the flood of information are limited. Therefore, automated extraction of the core of information, contained in real-word texts, is crucial, even at the cost of necessary simplifications. First, it is necessary to find documents which may contain response to the query for information. Then, relevant texts should be further processed in order to extract needed information, preferably in simple and easily understandable form. Automated knowledge extraction in specific domain is relatively simple, there are many examples of working systems. Methods used in such systems depend on the area of application and vary from simple pattern word search to complex syntactic and semantic analysis. Also statistical meas-

* Institute of Computer Engineering, Control and Robotics, Faculty of Electronics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland.

ures are frequently used. In contrast to specific domain, knowledge extraction from real-world, unstructured texts is a real challenge.

Two main approaches to knowledge extraction may be distinguished: Knowledge Engineering Approach and Automatic Training Approach. In first approach knowledge engineer develops grammar (rules) for the system. Rules are tested on moderate corpus of texts and possibly modified. This approach requires expertise, skill, detailed knowledge of the system and possibly consultation with an expert in the domain of application. In second approach training algorithm is run on hand-annotated corpus, rules are created automatically. Presented solution, based on the concept of elementary fact, is a combination of the two mentioned approaches.

2. PATTERNS OF ELEMENTARY FACTS

An elementary fact can be roughly defined as a simple assertion that some object is in given state, plays a role, participates in relationship etc. In real-word texts elementary facts are rarely expressed explicitly, usually they are embedded in complex sentences. In texts derived from real language resources there are virtually no restrictions on vocabulary, grammatical structure etc. Thus, extraction of elementary facts requires complex syntactic and semantic analysis. However, combined syntactic-semantic patterns are similar for many elementary facts (although expressed in different vocabulary). Moreover, the patterns may recur in seemingly unlike complex sentences. It is difficult to prepare in advance the catalogue of such patterns and contexts in which they may appear. Instead, we propose a way to identify the patterns and their context in the process of learning. As an illustration let us consider example sentence (1), taken from *Wikipedia* entry on Opel Astra car.

The Opel Astra is a small family car manufactured by the German automaker Opel since 1991. (1)

From the sentence (1) simple sentences (1a–1d) may be derived. They explicitly convey elementary facts contained in sentence (1):

Opel Astra is a small family car. (1a)
Opel Astra is manufactured by Opel. (1b)
Opel Astra is manufactured since 1991. (1c)
Opel is German automaker. (1d)

The sentence (1) and corresponding set of sentences (1a–1d) will be used in the process of learning patterns of elementary facts by examples. The process is performed in the following stages:

– complex sentence, along with corresponding set of simple sentences, are subject to syntactic and semantic analysis,
– syntactic-semantic images of sentences are created (these images reflect both components of sentences and relationships between them),
– obtained images are superimposed (the images of simple sentences are located within the structure of complex sentence),
– the patterns of elementary facts are created.

The result patterns, possibly with elements of close context, are stored in knowledge base of the system. They may be used later, in automated discovering of new elementary facts from sentences that appear at the input of the system.

For syntactic analysis Link Grammar is used [1, 2], a representative of the class of dependency grammars, where relations between words in sentence are described as labelled arcs. The result of syntactic analysis for sentence (1) is presented in Figure 1.

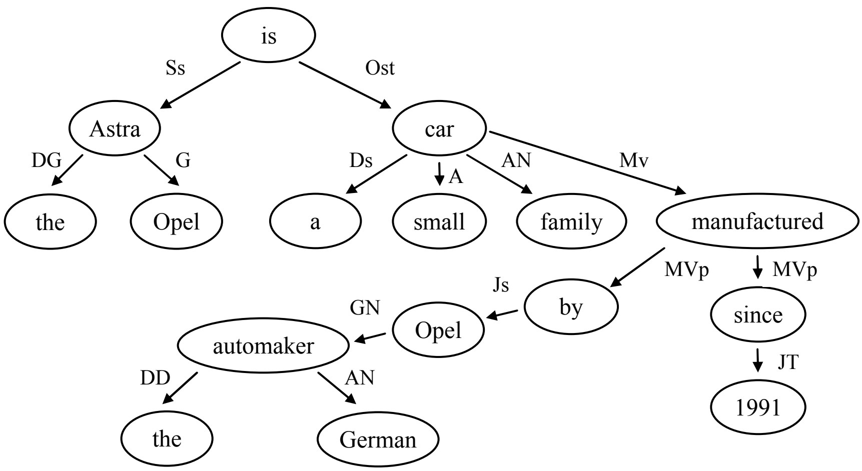


Fig. 1. Result of syntactic analysis for the sentence (1)

Nodes in the tree in Figure 1 represent words and directed, labeled edges describe relationship between words. The meaning of labels may be found in [1].

To obtain combined syntactic-semantic image of the sentence (1), the following operations are performed on the tree presented in Figure 1:

– the words in nodes are replaced by lexical categories (N stands for noun, V for verb, ADJ for adjective etc.),
– the nodes representing words with semantic contents (mainly nouns and verbs) are supplemented with semantic features.

Semantic features are obtained from Wordnet [3] (this is discussed in more detail further). Words without semantic content (prepositions and some types of determiners) remain unchanged in the structure of the tree. Syntactic-semantic image of the sentence (1) after described above transformations is shown in Figure 2.

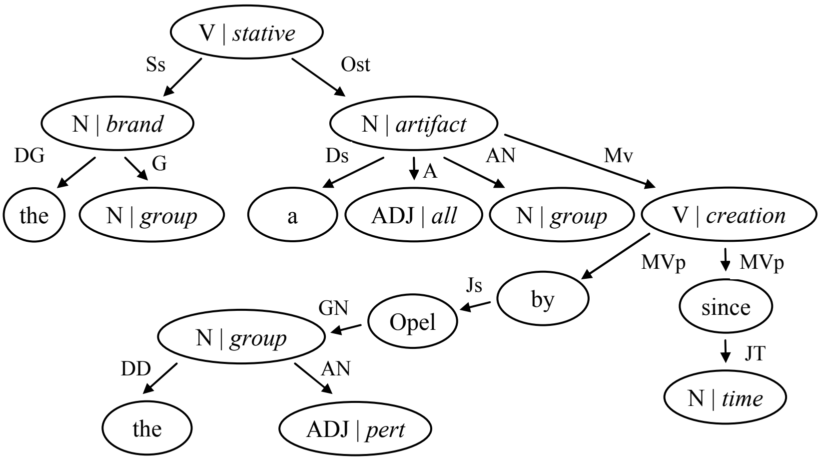


Fig. 2. Syntactic-semantic image for sentence (1)

Similar transformations are performed for simple sentences (1a–1d). In the next step syntactic graph, representing single elementary fact, is superimposed onto main graph representing complex sentence. The result of operation for simple sentence (1c) is shown in Figure 3 (bold lines denote graph representing sentence (1c), dashed line is used for edge which has no equivalent in main graph).

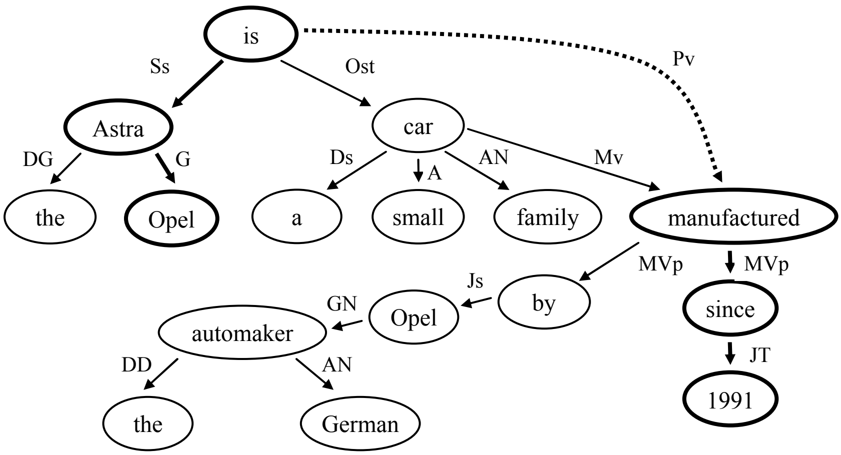


Fig. 3. Syntactic structure of sentence (1c) superimposed on sentence (1)

Parallel operations are also performed on syntactic-semantic images. Minimal connected subgraph of main graph, including complete syntactic-semantic structure of sentence (1c), represents pattern corresponding to elementary fact described by simple sentence (1c). Some parts of main graph, not present in the graph representing simple sentence (here this is the path Ost–N | *artifact*–Mv), reflect close context.

The final form of pattern, representing elementary fact described by sentence (1c), is shown in Figure 4.

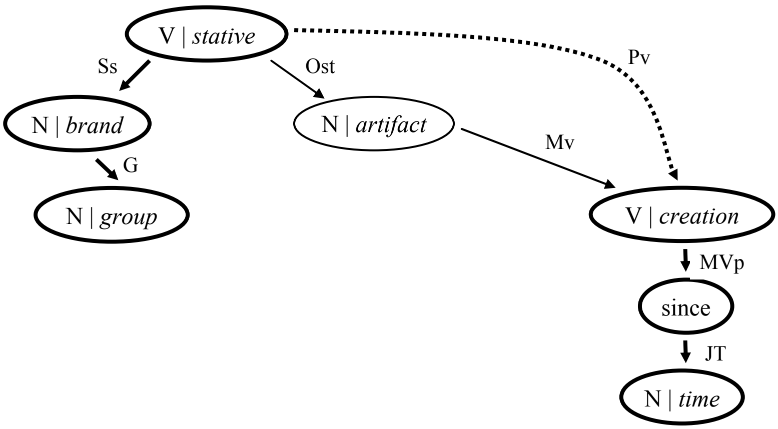


Fig. 4. Pattern of elementary fact derived from sentence (1c)

3. SYNTACTIC AND SEMANTIC ANALYSIS

In order to create patterns, which enable transformation of complex sentence into the set elementary facts, contained in the complex sentence, first syntactic and semantic analysis should be performed. It results in syntactic-semantic image of analyzed sentences (see Figure 2).

3.1. SYNTACTIC ANALYSIS

One of the main problems that must be faced in natural language processing is ambiguity. At syntactic level ambiguity may result in many potential structures representing the grammar structure of the sentence. Typical example is interpretation of prepositional phrases. They may be either components of nominal phrases or may act as independent phrases, appearing as object or adverbial. Sophisticated semantic analysis is necessary to select appropriate structure.

The problem of ambiguity also appears when Link Grammar is used as a tool for syntactic analysis. Experiments carried out on sample texts show that (if extensive dictionary is available) it is possible to perform syntactic analysis even for very complex sentences. However, the number of obtained alternative syntactic structures (linkages) is increasing significantly along with the number of words in the sentence. In extreme case more than 10000 alternative linkages were generated for one sentence. This relationship is illustrated in Table 1.

Table 1. Relation between number of words and number of alternative linkages

Number of words in sentence	Min.	Max	Average
Up to 5	1	2	1.5
6–10	1	12	4.0
11–15	1	72	17.6
15–20	8	420	129.4

As it was mentioned above, advanced semantic analysis is necessary to select the most appropriate syntactic structure from many alternatives. Unfortunately, there are no effective tools for such analysis, particularly in the case of complex sentences. Therefore, different solution is proposed. It is based on statistical analysis of alternative syntactic structures. The following algorithm is used to select appropriate structure $g_i(s_x)$ for sentence s_x:

- set value N_{max}, specifying maximum number of analyzed structures,
- get value N, the number of obtained structures $g_i(s_x)$ for sentence s_x,
- if $N > N_{max}$, assume $N = N_{max}$,
- for each link l_j = (word$_1$, word$_2$, label), for example l_j = (*car*, *family*, AN), compute coefficient $c(l_j)$, representing the number of appearances of given link in structures $g_i(s_x)$ (where $i = 1, 2, ..., N$),
- compute normalized values of coefficients $c'(l_j) = c(l_j)/N$,
- for each structure $g_i(s_x)$ (where $i = 1, 2, ..., N$) compute the sum $t(g_i)$ of coefficients $c'(l_j)$ attributed to all links appearing in that structure,
- compute normalized values of sums $t'(g_i) = t(g_i)/M$, where M – number of links in structure $g_i(s_x)$,
- select structure $g^*_i(s_x)$ with maximum value of normalized sum $t'(g_i)$.

Statistical analysis, used as a substitute for full semantic analysis, may certainly result in selection of improper structure. However, it must be emphasized that statistical methods, extensively used today (at different levels of analysis of natural language) usually produce acceptable results.

Effectiveness of proposed algorithm may be increased by supplementary mechanisms. Originators of Link Grammar introduced parameter called LEN (based on concept of link length) for evaluation of generated linkage. Link length is defined as a distance between linked words (e.g. link length for adjoining words has value 1). Parameter LEN represents the sum of all link lengths for given linkage [2]. It allows to evaluate complexity of particular structure and consequently to select less complicated structure for further analysis. It should be stressed, however, that this criterion is not always justified. There is no guarantee that simplest structure is always correct.

In the process of learning, patterns of elementary facts are obtained by superimposition of the structures of simple sentences onto the structure of complex sentence. It seems justified, that better matching structures should be preferred. This may be also considered at the stage of linkage selection. From Table 1 it may be concluded that the number of alternative structures for simple sentences (elementary facts) is considerably low. Therefore, in the process of evaluation of alternative structures for complex sentence, it is reasonable to prefer structures better matching with structures of elementary facts.

Another extension of the proposed approach may consist in introducing at least partial semantic analysis. This kind of analysis may be based on syntactic-semantic links. In such links syntactic requirements (e.g. link is acceptable between noun and adjective) are supplemented by some semantic requirements (link is acceptable only between words with proper semantic features). Application of syntactic-semantic links for sentence analysis is presented in [4].

3.2. SEMANTIC ANALYSIS

In presented approach semantic analysis is limited to determining semantic features for some words in the sentence. Only semantically significant words (content words) are considered. Semantic features are obtained from Wordnet, a kind of dictionary or rather large lexical database of English. Nouns, verbs, adjectives and adverbs in Wordnet are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concepts [3]. Each of synsets is linked to other synsets by means of a small number of conceptual relations (*synonymy*, *antonymy*, *hyperonymy*, *hyponymy* and so on). Basing on Wordnet, words may be divided into some groups according to semantic similarity. For instance, nouns are divided into 25 categories (*person*, *animal*, *artifact*, *natural object*, *substance* and so on). These categories constitute semantic features and replace content words in syntactic-semantic image of the sentence.

It should be pointed out that designation of semantic features may be ambiguous. The main reason for possible ambiguity is phenomenon called polysemy, which means that given word has multiple meanings. For example, the word *part* may function as noun (12 meanings), verb (5 meanings), adjective (one meaning) or adverb (one meaning). To illustrate the issue all 12 meanings of noun *part*, obtained from Wordnet, are presented in Figure 5.

The noun part has 12 senses (first 12 from tagged texts)

1. <noun.relation> **part**, portion, component part, component -- (something determined in relation to something that ir
2. <noun.location> region, **part** -- (the extended spatial location of something; "the farming regions of France"; "religi
3. <noun.cognition> **part**2 -- (so far as concerns the actor specified; "it requires vigilance on our part" or "they resiste
4. <noun.artifact> **part**, portion -- (something less than the whole of a human artifact: "the rear part of the house"; "gl
5. <noun.cognition> **part**, section, division -- (one of the portions into which something is regarded as divided and wh
6. <noun.act> function, office1, **part**1, role -- (the actions and activities assigned to or required or expected of a pers
7. <noun.object> **part**, piece -- (a portion of a natural object; "they analyzed the river into three parts"; "he needed a pi
8. <noun.cognition> character1, role, theatrical role, **part**1, persona -- (an actor's portrayal of someone in a play; "she
9. <noun.possession> share, portion, **part**, percentage -- (an asset belonging to or due to or contributed by an individu
10. <noun.act> contribution2, **part**3, share1 -- (any one of a number of individual efforts in a common endeavor: "I a
11. <noun.communication> **part**, voice2 -- (the melody carried by a particular voice or instrument in polyphonic musi
12. <noun.body> **part**1 -- (a line where the hair is parted; "his part was right in the middle")

Fig. 5. Meanings of noun *part* (from Wordnet)

Selection of proper meaning of word, depending on its context in sentence, is complicated task because specific knowledge of the real world is usually required. The most straightforward solution (at current state of research) is storing all possible semantic features for given word in dedicated part of knowledge base. It is also possible to build contextual dictionary, including only meanings for specific domain. Currently semantic features are obtained directly from Wordnet.

Another problem that must be considered at the stage of semantic analysis is the way of recognizing proper nouns [5]. Wordnet includes only common proper nouns such as names of states and nationalities, basic geographical names etc. Thus we can find in Wordnet such words as *German*, *American*, *Baltic* whereas words *Opel*, *Astra* (name of company, make etc.) are not present. Therefore, supplementary dictionary is necessary to store proper nouns, specific to given domain. In order to obtain compatibility, entries in such dictionary should have structure similar to that of Wordnet.

Syntactic-semantic patterns, obtained by replacing specific word with its semantic feature, are generalizations (they may be used not only for specific concept but also for the class of concepts with similar meaning). However, excessive generalization may produce of false elementary facts. For instance, words *car*, *plane*, *ship*, but also *bridge*, *house*, *chair* have (in Wordnet) the same semantic feature *artifact*. All these concepts have common feature (were created by human), but there are also noticeable differences in meaning which may lead to errors. Therefore, generalization should be broad enough to encompass noticeable semantic similarities but should not lead to significant number of false identifications of elementary facts. The problem may be solved with the help of relationships available in Wordnet, namely hyponymy and hypernymy. These relationships make possible to organize vocabulary (in consequence also concepts) in the form of hierarchy. The word *car* has the following hypernyms: *motor vehicle*, *conveyance*, *instrumentality* and *artifact*. Thus, one of these hypernyms may be used as semantic feature (not only the most general *artifact*). Selection of hypernym affects the level of generalization of the concept. Distance between word and its hypernym may be used as a measure characterizing the level of generalization. This measure may be one of significant parameters influencing the process of building patterns for elementary facts.

4. EXTRACTION OF ELEMENTARY FACTS

Let us assume that syntactic-semantic patterns of elementary facts, obtained from complex sentence (1) and the set of simple sentences (1a–1d), are already stored in knowledge base. These patterns may be used when it is necessary to extract new elementary facts from new sentences. Now the sentence (2) is presented at the input of the system.

The Volkswagen Golf is a small family car manufactured by Volkswagen since 1974 and marketed worldwide across six generations. (2)

The sentence (2) is transformed in a manner similar to that shown before for sentence (1). The result of syntactic analysis is shown in Figure 6.

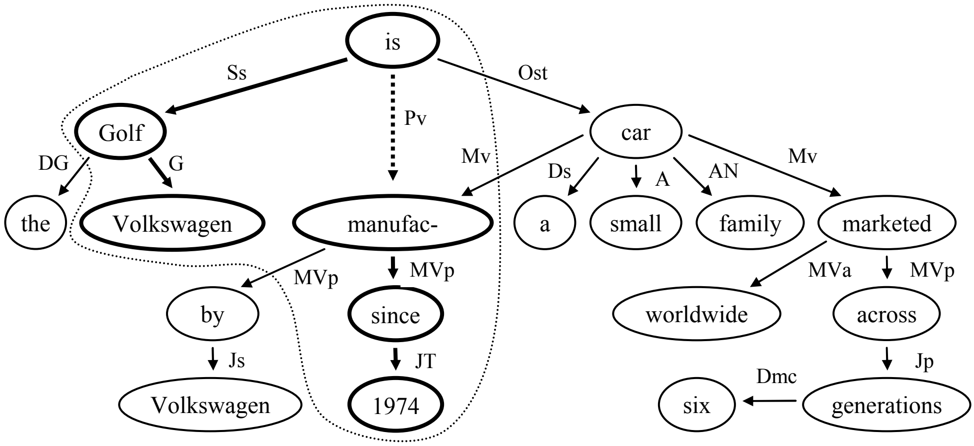


Fig. 6. Result of syntactic analysis for the sentence (2)

In second stage, after semantic analysis, combined syntactic-semantic image of the sentence is created (it is not shown here). Then, attempt is made to match patterns of elementary facts up with syntactic-semantic image of sentence (2). Here the match was found for elementary fact pattern (shown in Figure 4) derived from simple sentence (1c). For the sake of simplicity identified pattern is shown (in bold lines) in Figure 6, whereas, actual matching is performed on syntactic-semantic images. Dashed line represents edge from pattern which was not found in the structure of the sentence (2). However, there is indirect path connecting vertices (*is*) and (*manufactured*) (Ost–*car*–Mv). It means, that close context requirements are fulfilled.

Identified pattern is transformed back into words and thus new elementary fact is obtained in the form of sentence (2a).

Volkswagen Golf is manufactured since 1974. (2a)

5. CONCLUSION

In presented approach it is assumed that essential part of knowledge, contained in complex sentence, may be transformed into the set of simple statements. These statements have equivalent linguistic patterns in source text. Manual creation of patterns may be time consuming and unreliable, even for a small application domain. Therefore, in proposed solution, patterns are acquired from training texts. Although this technique may be quite useful for informative texts (e.g. biographies [6] or descriptions of cars), it may fail in case of texts concerning abstract concepts and relations. Knowledge expressed as elementary facts is not only easy to comprehend but may be also further processed conveniently, using reliable formal mechanisms (e.g. predicate logic as a model for reasoning). Further research is needed, especially on advanced syntactic-semantic patterns matching.

REFERENCES

[1] SLEATOR D.D.K., TEMPERLEY D., *Parsing English with a link grammar*, Technical Report CMU-CS-91-196, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1991, 7–26.

[2] GRINBERG D., LAFFERTY J., SLEATOR D., *A robust parsing algorithm for link grammars*, Technical Report CMU-CS-95-125, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1995, 1–9.

[3] MILLER G.A., BECKWITH R., FELLBAUM C., GROSS D., MILLER K., *Introduction to Wordnet. An on-line lexical database*, International Journal of Lexicography, 1990, 3 (4), 235–312.

[4] BANASIAK D., MIERZWA J., STERNA A., *Modified Link Grammar in syntactic-semantic analysis of natural language texts*, Polish Journal of Environmental Studies, Vol. 18, No. 3B, 2009, 11–15.

[5] MYKOWIECKA A., *Inżynieria lingwistyczna. Komputerowe przetwarzanie tekstów w języku naturalnym*, Warszawa, Wydawnictwo PJWSTK, 2007, 259–311.

[6] BANASIAK D., MIERZWA J., STERNA A., *Acquisition of knowledge pieces from natural language texts in intelligent information system*, In: Information Systems Architecture and Technology: New developments in Web-Age Information Systems, Borzemski L. (Ed.), Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2010, 187–197.

System monitoring, log processing
operational profiles

Marcin KUBACKI*, Janusz SOSNOWSKI*

ANALYSING EVENT LOG PROFILES IN LINUX SYSTEMS

The chapter presents the problem of monitoring computer system operation based on various event logs. We concentrate on Linux systems and available standard log programs. The basic idea is to collect event logs from many computers and correlate them with operational and user profiles. To deal efficiently with the bulk of collected data we use a special software module developed in the Institute (QLogAnalyser), which provides useful capabilities to deal with variety of log formats and facilitates detecting interesting situations using regular expressions. It supports also visualisation of various statistics. The practical usefulness of the developed approach to system monitoring has been illustrated with results related to one of didactic laboratories in the Institute. In particular we present the specificity of usage profiles, workloads and identified problems including some erroneous situations which occurred within one semester.

1. INTRODUCTION

Practically in all systems we have various HW and SW monitoring mechanisms; in particular the most popular are logs which register various events [1,2,12]. Analysing these logs is to some extent boring task due to a large amount of registered data at first glance looking not interesting, the more that most of the time systems operate reliably. However some strange situations may occur and are not easy to explain based on external observations. Dealing with a large number of computers (e.g. student laboratories) generates many questions on their availability, user and administration problems. To get reasonable answers to these problems we have to perform more systematic observation of system operation. In this process a well-organized log collection and analysis are of great help. Some experience with this approach for Windows systems

---

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

we have published in [3, 10–13]. This research was targeted at detecting well known problems related to system downtimes, power problems, etc. In this work we deal with Linux based systems used in our didactic laboratories. It is worth mentioning that the operational profiles of computers in didactic laboratories differ significantly from those encountered in research of commercial applications or large scale computers (mostly described in the literature e.g. [4, 5, 7,9, 14] and references therein).

The goal of our research was to study dependability issues and operational profiles of didactic laboratories. For this purpose we have developed a centralized system of collecting and analysing logs within Linux based computers used in our laboratories. It is composed of program scripts (collectors) embedded on each monitored workstation and collaborating with them a centralized analyser. Collecting event logs for such set of computers for a longer time period we have identified some interesting operational features and problems which usually are difficult to notice from the external perspective. To facilitate the analysis we have introduced some form of regular expressions to specify properties of searched events or define additional conditions of their occurrence. In particular we can define a time window around some specific events, correlate them with user sessions, etc. This approach allows identifying administration and user problems as well as evaluating system availability. This can be used to improve performance and service qualities; monitor not accepted student activities, etc.

Section 2 outlines the specificity of Linux event logging and describes the features of developed log analyser. Section 3 presents some illustrating statistics related to the monitored workstations. The gained experience is concluded in section 4.

2. PROCESSING EVENT LOGS WITH QLOGANALYSER

Most of the applications and operating systems generate log files with important information about events that has occurred during their run. Here we give some general outline of several aspects of log files in Linux and UNIX. Typically they are text files comprising lines consisting of timestamp (some files do not comprise this), message and optionally few additional fields. It is quite hard to process text files if there is no single format used in log files. Hence it is reasonable to reformat the files and filter out not important data, etc.

We concentrate on syslog daemon, which is the main facility for logging in Linux system, used by many applications. Syslog log files comprise information on application crashes, starts and terminations of services, mail delivery information, etc. Every program can use specific API to insert an entry in the logs. Moreover a special Linux program (logger) can be used to put a message into a log file. Each log entry may consist of the following fields: month ID, day of the month, exact time, host name, program (subsystem) that generated message with optional PID in square brackets, and the message (there is no information on the year).

Syslog entries in log files do not contain severity level of the event. However, syslog daemon can put events in different files according to their severity level. There are several subsystems (related to different log message sources) defined in the syslog: *kern* (kernel), *user* (user processes), *mail* (mail daemon), *lpr* (printing daemon), *daemon* (messages from daemons), *auth* (not critical information about user authentication), *authpriv* (critical information about user authentication), *ftp* (FTP daemon), *cron* (information about scheduled tasks and result of their run), *syslog* (internal syslog messages), *mark* (periodically inserts marks into logs), *local0-7* (eight subsystems that can be used for other purposes). The generated messages may comprise severity level of the event: *emerg* (highest severity), *alert* (needs immediate attention), *crit* (critical errors, like hardware faults), *err* (other errors), *warning*, *notice*, *info* (informational), *debug* (verbose), *none* (ignored).

Syslog has been created in 1980's, primarily for sendmail (mail daemon). Today another daemon called syslog-ng offers more log files management options, secure files transfer, use of the database like Oracle, PostreSQL or SQLite, data buffering and multi-platform support (agent to collect events from Windows systems also exists), advanced event filtering with classification and pre-processing of events (syslog-ng is even able to modify event data); however log entries format did not change. The real advantage of syslog-ng can be noticed in event processing. For example, a program writes a message to the specific file, which is treated by syslog-ng as an event source. By default all of the messages are written to pseudo-file /dev/log. Syslog-ng searches for configuration entry for this event source. In another step, it applies all of the filters defined for event type. In the last step, event is being send to the target, which in turn can be a file or remote system (i.e. another log server).

Information on user sessions (logins, etc.) can be found in last log (binary log file). Unfortunately, this information is stored in binary file called – /var/log/wtmp. This file cannot be used directly – it needs to be read by "last" program. Its output can be stored in other file (in text form). Each entry (with "-a" switch) is similar to the following: xeonic pts/2 Tue Feb 2 13:58–14:08 (00:09) 192.168.30.1. The entries include username, terminal that user had been attached to, session start and end time, session duration time (in brackets), and host (IP address or name if available) from which the user started this session. There are some details worth to mention. Firstly, there exists a pseudo-user called "reboot", which appears in listing from "last" program each time the system is being started. Secondly, in sessions with X-windows we have double session entries (from the system and X-windows). Moreover, the currently logged users do not have session end timestamp, but "still logged in" message. If a session lasts longer than 24 hours, session duration time is shown in the form d+hh:mm (additional "d+"), where d is the number of days, hh – hours, mm – minutes. As a result, it is advised to use short script to transform "last" entries into more easy to process form like: Feb 12 08:10 (00:06) xeonic, where all unnecessary information is removed, leaving only date and time of the session start, its duration in constant format (hh:mm).

To deal with the multitude as well as with diversity (different formats) of logs and support their analysis a special tool has been developed – QLogAnalyser. It imports text log files in formats defined by the user and provides useful functions to process and analyse the collected data. Analysing event logs we can filter their contents using regular expressions and add (if needed) other conditions (e.g. observation time window, predecessor or successor reference points). These capabilities are similar to those included in other tools (dedicated for Windows) developed in the Institute [10–13]. QLogAnalyser provides also the capability of calculating various statistics and visualizing them in different forms.

In QLogAnalyzer the user can define log file format by regular expressions and date/time expressions used in Qt library. The regular expressions are composed of character strings, operators (quantifiers – specifying number of expression occurrences) and assertions (specifying expression location in the text). All characters except the special ones are specified explicitly. Using backward slash \ we define specific characters or words e.g. \n – line feed, \r – carriage return, \d – decimal digit, \D – anything except decimal digit, \s – blank character (space, tab, etc.), \S – anything except blank characters. Square brackets are used to specify any listed character – [list of characters], or any character except those from the list – [^list of characters]. Simple brackets (,) are used to create groups in expressions. Quantifiers may follow an expression to denote the number of occurrences: ? – no or a single occurrence, + – single or multiple occurrence, * – no or any occurrence, {n} – precisely n occurrences, {n,} – at least n occurrences, {,m} – at most m occurrences, {n,m} – at least n and at most m occurrences. Assertions define locations of specified expressions in character strings: ^ – the beginning of the string, $ – the end of the string, \b – boarder of a word, \B – no word boarder, (?=E) – identifies an expression which is followed by the specified expression E, (/!E) – identifies an expression not followed by expression E.

Using date/time and regular expressions the user defines each field in the considered log line, that should be treated as separate column. For example for syslog log file, the first field is the date and time. User defines its name, sets type to “date/time” and enters expression describing first field – in this case it is “MMM d hh:mm:ss”, where “MMM” is month, “d” – day of the month, “hh:mm:ss” – hours, minutes and seconds. Blank characters like spaces are removed after each field, so next field (host name) can be defined by regular expression meaning “everything but space”: “[^]*”. Another field, process name has expression much more complicated: “([^\[:]*)\[?[^:]*:”. The first bracket shows beginning of the group. This part is the actually name of the process – expression “[^\[:]*” means “everything but opening square bracket or colon” – both punctuation marks show the end of the process name. However, second part of this expression “\[?[^:]*:” captures optional opening square bracket and everything that is not colon followed by the ending colon – this is PID field. It is necessary to describe the context of the "program name" field, but in this case the user can inform QLogAnalyzer to capture only part of the text in the first expression group. It does not mean, that the part of the log entry matched by the second part of this expression is lost – it will be matched by expression defined in another field "PID" which is: "\[([^\]]*)\]". This expression directs QLogAnalyzer to match the text starting with the square bracket and then group everything between those square brackets. Inside group expression "[^\]]*" matches everything but square bracket, which in fact is the process ID. Now, the hardest part is the "Message" field. It should be noticed, that PID is an optional field, so the user has to show which part of the text is optional, and which is important. To complete this task, the following regular expression has been proposed: "([^\]:]*\])?:? *(.*)". First group (beginning from round bracket) is optional (notice question mark after closing round bracket) and matches everything but square bracket and colon. This group is matched, when PID in square brackets is given. After optional colon and spaces another group is created. Second expression group matches the rest of the line, and is marked to be captured by QLogAnalyzer.

QLogAnalyzer parses each line in the log file using a defined set of expressions (multiline messages have to be joined into a single line – it can be accomplished with shell or awk script). The user can specify which fields in log format should not be used when comparing two events (if they have to be treated as identical). This feature is called "wildcard", for example we can use it to eliminate timestamp field. The "wildcard" flag can also be applied to regular expression fields e.g. PID field. Typically, it is only important to keep program name not its process ID, which is different each time program starts. "Wildcard" flag is important during statistical analysis, when QLogAnalyzer has to decide whether the analysed event is a new event instance or another occurrence of already known event.

Many log formats comprise additional information in "message" field like UID, PID, memory addresses, file system paths, etc. So the log with the entries describing the same situation can be counted separately, because the "message" field differs and the user has no ability to cut off the identification part from this field. One solution is to pre-process log files and remove these identifiers from the message field. In [6] the authors count the occurrences of each word on each position in the log entry to classify words as parameters or a constant text for each event type.

In the next section we give some practical illustration of the QLogAnaliser capabilities in relevance to the monitored computers in a student laboratory. We have derived statistics characterizing computer load, users activities, correlated them with courses and projects. We have identified various I/O errors, authentication errors, restart statistics, NFS connection problems, operational profiles of user and courses, etc. These statistics can be presented in various time perspectives: fined grained linear plots, aggregated statistics in day, week and month perspectives.

3. PRACTICAL RESULTS

Using QLogAnalyser we have monitored events in one didactic laboratory composed of 16 IBM PC computers (C1–C16) and one server (galleon). The computers do not comprise hard discs, their software environment is configured from the server during power on each morning. Typically all computers are switched off centrally at the end of each day. The laboratory is used for practical classes of regular courses on computer architecture (ARKO), Discrete Random Processes (EDRP), Elements of Statistics (ESTA) and projects on Analysis of Algorithm (AAL), Software Project (PROS), Event Programming (PROE), Compilation Techniques (TEKO), Internet Techniques (TIN). During semester the laboratory is open to the students from 8.15 till 21.00 during weekdays Monday–Friday. The practical classes and projects are attributed to a fixed time schedule resulting in 9, 6, 4, 7 and 3 hours for subsequent days (starting from Monday, respectively. During the remaining hours the laboratory is available to any students (e.g. for their own work or other irregular projects). Sporadically the laboratory is used on weekends for special courses. It is worth noting that during projects the students appear usually at the beginning of the semester, then they use the reserved hours at their own need. The goal of monitoring was three fold;

1. Exploring the informative significance of collected logs.
2. Finding characteristics of computer workloads and user activity.
3. Detecting anomalies and problems.

The first goal was just to get knowledge on the scope and accuracy of monitoring, the quantitative and qualitative contents of logs, their formats availability, etc. Some remarks have been given in section 2 and they were the basis in developing the functionalities of QLogAnalyser. In particular we have found that the collected data is useful to realize the two subsequent goals.

In the sequel we give some selected results related to two month observation (March and April) of the laboratory within the summer semester. This period represents a typical usage of the computers in accordance with the above specified time schedule and student courses.

Computer and user activities

Tab. 1 shows the number of logins attributed to the most active 10 users (among 279 registered users). Moreover it gives the distribution of user activities: the number of users (N_U) attributed to the specified logins ranges ($[N_L]$). Over 54% of users appeared sporadically in the laboratory (low number of logins). The total number of logins was 760. The distribution over weekdays was as follows: 97, 194, 115, 195, 81,78 (for Monday–Saturday, respectively). The distribution over day hours was as follows: 9, 80, 45, 144, 92, 95, 90, 87, 70, 25, 13, and 12 (for subsequent hours starting from 7 a.m. till 6 p.m.). These distributions are to some extent correlated with courses. The distribution of logins over different courses is given in tab. 2. The distribution of logins over computers was in the range 28–80, in most cases close to 35 and for 4 computers (close to the laboratory door) 60–80.

Table 1. User activity distribution
(N_L – number of user logins, N_U – number of users with logins within [N_L] range)

10 most active users				User activity distribution			
User ID	N_L	User ID	N_L	[N_L]	N_U	[N_L]	N_U
U1	49	U6	29	[1–5]	133	[26–30]	3
U2	40	U7	28	[6–10]	54	[31–35]	1
U3	38	U8	26	[11–15]	23	[36–40]	1
U4	33	U9	25	[16–20]	19	[41–45]	0
U5	30	U10	24	[21–25]	9	[46–50]	1

Table 2. Computer load vs. courses (the number of user logins)

Course	March	April	Σ	Course	March	April	Σ
AAL	17	10	27	PROE1	8	0	8
ARKO1	42	13	55	PROE2	41	8	49
ARKO2	51	13	64	PROE3	30	11	41
ARKO3	36	16	52	TEKO1	6	1	7
EDRP1	12	3	15	TEKO2	4	3	7
EDRP2	8	7	15	TIN	18	3	21
ESTA	15	0	15	TEKO3	15	15	50
PROS	3	1	4				

Errors and anomalies

Looking for erroneous and anomalous situations we have to detect all warning and error events in logs as well as we have to trace deviations from normal profiles of event images, correlated with user profiles. In tab. 3 we give a distribution of I/O errors related to the use of USB pen drive devices.

Error distribution over computers is random due to different usage. Moreover we have found that errors dominated on Thursdays (877 errors vs. 100 errors on Mondays and Wednesdays, 10 and 20 on Fridays and Saturdays, respectively). Different error numbers over computers do not result from their specificity but from the distribution of users activities with USB devices. Hence, more interesting is the error distribution over users. Here we have observed significant fluctuations. Errors dominated for some users (U11, U12, U2, U20). In general they resulted from user missoperation (e.g.

inserting USB device before login, detaching the device before disconnection procedure) or device incompatibilities with the computers (this happens for newer devices with bigger capacity) or erroneous devices. Tracing the moments of USB connection we have found that many users did not disconnect the USB device before removing it, so the error has been reported in the log. For some users we have found many errors for the same USB disc connection, this can be attributed to sector errors or incompatibility problems.

Table 3. Distribution of I/O errors over computers

Comp.	Errors	Comp.	Errors	User	Errors	User	Errors
C1	109	C9	308	U2	93	U17	2
C2	283	C10	118	U11	306	U18	3
C3	0	C11	8	U12	95	U19	2
C4	168	C12	0	U13	2	U20	27
C5	0	C13	2	U14	2	U21	2
C6	97	C14	2	U15	2		
C7	4	C15	27	U16	2		
C8	5	C16	2	U17	4		

In the monitored laboratory it was assumed that all computers are switched on centrally in the morning and switched off in the evening. The computers do not comprise hard disc, so the operating system, program applications, user data, etc. are loaded from the galleon sever via NFS protocol. The connection IP is established using DHCP protocol. The configuration process is mostly performed between 7 and 8 hours a.m. However sporadically the configuration processes appear also up to 18 hours (due to local switching off/on), In the log files we have identified 1032 events “NFS server not responding” – related to some connection problems. All of these events except 14 were followed (after some delay) with “NFS server O.K.” So the communication problem was temporary. The remaining events related to some incompatibility problem between the NFS and system kernel versions (resulting from the resynchronization effect [17]).

Quite important is tracing system startups and closings. In the ideal case we should have the same numbers of these both events. In the monitored laboratory we have found 709 start-ups, 107 correct closings and 66 restarts events (rather abnormal situation). This anomaly resulted from the simplified management of the laboratory, in particular at the end of the day the power supply of computers C1–C16 is switched off centrally to all the computers, instead of individual system closing. This procedure is acceptable due to the lack of hard discs in the computers and sufficient time to store the necessary data in the server. Nevertheless some correct closing appeared due to specific courses or some users activities. The registered restarts related to encountered problems during practical courses in fact they resulted from programming errors e.g. system hang-ups. Within the time period of 2 months no physical error (in hardware) occurred.

Another important issue is monitoring the user authenticity problems. We have found that on average 25% logins generated authenticity problems. This level can be accepted due to the specific use of computers (different courses, many students). However we have observed a slightly larger percentage in late afternoon hours (maybe due to some fatigue of the users). For a few users this percentage was over 40%.

According to the goal two we were interested in overall usage of subsequent computers in correlation with different courses, user classes, etc. Moreover we tried to find user profiles (typical behaviors and deviations).

4. CONCLUSION

Our research proved the need and usefulness of monitoring operational profiles of computers. This is especially important in institutions admitting many people to use their computer resources (e.g. universities). It is important to correlate various logs and perform an integrated analysis taking into account normal, abnormal operational events, user logins and profiles, laboratory operational policy, etc. Due to some loose formats of registered messages in different types of logs an important issue is parsing them. The developed QLogAnalyzer has proved that using regular expressions to define log formats enables user to process different types of text log files. User is always able to convert binary event sources like those used by “last” program to suitable text form that can easily be described as a set of regular expressions.

The collected and analysed results from one of our laboratories confirmed practical significance of integrated event monitoring. In the retrospective analysis we have also found the need of logging user and administrator remarks during system operation, this may simplify interpretation of detected anomalies. The gained experience with QLogAnalyser and previous research on system performance [12] resulted in developing a new more universal tool which includes also performance logs. We plan to enhance it with data exploration algorithms (e.g. [2, 15, 16]) useful for failure prediction [8, 14].

REFERENCES

[1] CINQUE M. et al., *A logging approach for effective dependability evaluation of computer systems*, In: Proc. of 2nd IEEE Int. Conf. on Dependability, 2009, 105–110.

[2] HUANG L., et al., *Symptom based problem determination using log data abstraction*, In: Proceeding of CASCON 2010 Conference of the Center for Advanced Studies on Collaborative Research, ACM 2010, 313–326

[3] KRÓL M., SOSNOWSKI J., *Multidimensional monitoring of computer systems,* In: Proc. of IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, Brisbane, Australia, 7–9 July 2009, 68–74.

[4] LI M. L., et al., *Understanding the propagation of hard errors to software and implications for resilient system design*, In: ACM, ASPLOS'08 Conference Proceedings, 2008, 265–276.

[5] LIANG Y. et al., *Bluegene/L Failure analysis and prediction models*, In: Proc. of Int. Conf. on Dependable Systems and Networks, 2006.

[6] NAGGAPAN M., VOUK M.A., *Abstracting Log Lines to Log Event Types for Mining Software System Logs,* In: Proceedings of Mining Software Repositories (Co-Located with ICSE 2010), 2–3 May, 2010, Cape Town, South Africa. 114–117.

[7] OLINER A. STEARLEY J., *What Supercomputers Say: A Study of Five System Logs*, In: Proc. of the IEEE/IFIP International Conference on Dependable Systems and Networks, 2007.

[8] SALFINER F., LENK M., MALEK M., *A survey of online failure prediction methods*, In: ACM Comput. Survey, 42:10:1–10:42, March 2010.

[9] SHARMA A., KALBARCZYK Z., BARLOW J. IYER R.K., *Analysis of Security Data from a Large Computing Organization*, In: Proceedings of the 41st IEEE/IFIP International Conference on Dependable Systems and Networks, 2011, 506–517.

[10] SOSNOWSKI J., POLESZAK M., *On-line monitoring of computer systems,* In: Proc. of IEEE DELTA Workshop., 2006 , 327–331.

[11] SOSNOWSKI J., KRÓL M., MACHNICKI J., *Techniques and goals of monitoring computer systems,* In: A. Górski (Ed.), Information Systems Architecture and Technology, Advances in Web-Age Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, 2009, 235–246.

[12] SOSNOWSKI J., KRÓL M., *Dependability evaluation based on system monitoring*, In: Computational Intelligence and Modern Heuristics (Ed. Al-Dahoud Ali), Intech, 2010, 331–348.

[13] SOSNOWSKI J. *Tracing anomalies in computer systems*, In: In: Proceedings of the 41st IEEE/IFIP International Conference on Dependable Systems and Networks Supplemental, Poster Session, 2011, 127–130.

[14] Yu LI, ZHENG Z, LAN Z., *Practical Online Failure Prediction for Blue Gene/P: Period-based vs Event-driven*, In: Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, 2011, 259–264.

[15] XIA X. et al., *Multi-level logs based web performance evaluation and analysis*, In: Proc. of Int. Conference on Computer Application and System Modelling, 2010, 4.37–4.41.

[16] XU W., et al., *Detecting large scale system problems by mining console logs*, In: Proc. of ACM SOSP'09 Conference, 117–131.

[17] http://www.mail-archive.com/ltsp-discuss@lists.sourceforge.net/msg29611.html

internet advertising, online marketing systems
demographic advertising, behavior advertising

Karol KRÓL*

CONSTRUCTING AN IT SYSTEM FOR PROVIDING TAILORED AND EFFECTIVE INTERNET ADVERTISEMENTS

Modern internet advertising is a world of big business and big IT systems. Over the years many different platforms have emerged and provided their services to savvy advertisers all over the globe. One of them - Google AdWords - has become the leader and now controls circa 69% of the market. The most popular billing methods are: CPM (Cost per Thousand), CPC (Cost per Click), CPV (Cost per View) and CPA (Cost per Action). The ads can be delivered in many different ways. The most popular of them is called contextual advertising. The principle of it is to display a targeted ad to the user while they are browsing the internet for some specific topic.

Such an approach has its flaws though. Flaws that can be fixed by a more sophisticated system. The system proposed in this work consists of two main parts: the Server Z and the Client X. The inner structure of the Server Z consists of four components: the users module, the websites module, the advertisements module, and the central module (fig. 2). Each of these modules executes different operations in order to provide a targeted ad to a user who is browsing the internet.

The goal of the system is to provide an ad based on the previous activity of the user. Gathering and processing various data concerning this activity is the core task of the system. In its final stage the system can be integrated with other existing advertising systems.

1. DESCRIPTION OF THE PROBLEM

1.1. A PAGE OUT OF HISTORY

The internet advertising market has its beginning in 1994 [1, 2, 7]. It was the time when first IT systems for displaying online advertisements were created. At first, the

* Silesian University of Technology, Akademicka 16, 44-100 Gliwice Poland.

systems were very simple, and they did not have most of the features that are considered the standard nowadays.

Since 1994 many different IT systems were developed only to be shut down without gaining any significant popularity during their lifespan. Along with IT systems many different revenue models for internet advertising were developed too, and, similarly, many of them have not withstood the test of time.

The most important date in the history of internet advertising is 10/23/2000. On that day, the main player in the search engine world – Google has introduced their new online platform – AdWords [3]. Since that day the system had numerous incarnations, each one altering its main features and characteristics. All for the purpose of fulfilling customers' needs and using the newest developments of information technology.

1.2. THE MARKET

Nowadays, Google and their flagship product – AdWords is the undisputed leader on the market. Available data indicate that in 2008 Google controlled circa 69% of the market share, and their revenue at the end of 2008 reached 21 billion dollars (US) [4].

Google has only a small number of main competitors, but one of them is getting close really fast. Facebook, the competitor in question, is the biggest online social network with over 600 million users around the world. Numbers of such a scale create an enormous advertising potential, which Facebook has started to take advantage of not that long ago. In 2007 their internal advertising platform has been launched and is running ever since.

The other big competitors for Google are Yahoo! and Microsoft. In 2010 these companies combined their efforts to create one big advertising platform which would be able to compete with Google more efficiently.

Of course, the market of internet advertising is much bigger than just four companies. There is a big number of other companies, but their market share is insignificant when compared to the four giants.

1.3. MODERN WAYS OF DELIVERING INTERNET ADVERTISEMENTS

The best approach to categorize the ways of delivering internet advertisements is to look at what the creators of the main platforms are providing. Nowadays, the most common and popular models are: CPM, CPV, CPC and CPA [5].

The CPM model (Cost per Thousand) is the first, original delivery and revenue model. Despite this fact it is still in use in modern advertising platforms till this day. The main idea is that every advertiser is settled for a package of 1,000 views of their advertisements. From a programmer point of view it is the easiest model to implement because the only thing there is to do is to count each ad impression and stop displaying them when the number reaches the amount which the advertiser paid for.

The CPV model (Cost per View) is a model very similar to CPM. The only difference is that the advertiser is charged for every separate view of their advertisement instead of the package of 1,000 views.

The CPC model (Cost per Click) is currently the standard for modern internet advertising. The main idea is that the advertiser is charged for each click that their advertisement receives individually. It is a model much safer for less experienced advertisers because there is no cost for displaying the advertisement like there is with CPM and CPV models.

The CPA model (Cost per Action) is the latest and most advanced revenue model for internet advertising systems. Both in terms of business and programming. In this model the advertiser is charged only when a person surfing the internet sequentially: sees the advertisement, clicks on it, gets redirected to the advertiser's website, and performs some kind of action on that website. The action is defined by the advertiser. It can be anything. For example: purchasing a product, filling out a contact form, subscribing to a newsletter. The advertising platform does not restrict any kind of actions. This makes this model extremely customer-friendly and highly customizable. On the other side it also makes it somewhat difficult to implement because the system has to be able to record a single occurrence of every action, so some kind of universal method has to be used.

1.4. MODERN WAYS OF MATCHING ADVERTISEMENTS TO THEIR RECIPIENTS

People are present and active in many parts of the internet. Of course, each of these parts can be a great place for advertising. Currently, the most popular type of advertising is contextual advertising. In simple words, the advertisements are matched to the specific context that the user is in at a given moment. This is done in two ways.

When it comes to advertising in search engines the ads are matched to the keywords users put into the search field. When it comes to advertising on various websites around the internet the ads are matched to the contents of the websites themselves.

There is one more approach. A new one, called demographic matching. In this case each user has a more or less precise profile that presents a set of demographic traits. The advertiser uses these traits to target them with ads. For example, it is possible to show an ad only to single men over 27 years old living in Paris. Even though it sounds very tempting the systems using this model do not perform much better than ones using other, more traditional approaches.

1.5. HOW ADVERTISING SYSTEMS ARE DESIGNED AND BUILT

The details of the way modern advertising systems are constructed are not fully known, which is not surprising for obvious reasons. However, some typical elements are common for many of them.

The most popular advertising systems are built as web-based applications. They are constructed with the usage of common internet technologies that are widely used in numerous other internet applications as well as simple websites. The programming is done in languages such as: PHP, ASP.NET, Ruby on Rails, Python, JavaScript, HTML. The most common database engines are MySQL and PostgreSQL.

The client side of these systems is targeted towards advertisers, people not necessarily computer savvy, so it needs to be easy to use and intuitive. Therefore the user interfaces are usually designed to look somewhat like bigger websites that use a number of HTML forms, menus, and sub-pages to provide the necessary functionality.

1.6. COMMON PROBLEMS AND POSSIBLE SOLUTIONS

The most lively problem of the internet advertising world are the numbers describing its effectiveness. A well-performing ad displayed among the content of a website is one that receives a 0.3% click through ratio [6]. A well-performing ad displayed on search engine result pages is one that receives a click through rate between 1% and 3%. Such a situation looks a bit worrisome because, as it turns out, even very modern advertising systems are not able to improve these results.

Over the years internet users have developed a new skill for immunizing on internet ads. Each day they are less effective and bring fewer results (when we look at an advertisement as a single unit not on the whole market in general). One of the reasons for that is the fact that advertising systems of today still do not take into account the current state of the user nor their history nor any kind of pattern of their activity. The demographic profiles are usually too simple to accurately predict which ad could have the best effect on a user or even what is it that the user is currently interested in (what they are searching for). The profiles also do not predict what the possible commercial value of this interest is (the probability that the user is likely to spend money on something they are searching for). Introducing each of these elements surely is a promising prospect for the future. Especially considering the fact that the current market will soon reach the level of efficiency beneath an acceptable value, and therefore advertisers will start to abandon it looking for other possibilities elsewhere. It all happens despite the fact that the market as a whole is still growing both in terms of revenue and number of users. But the truth is that those users have to invest bigger amounts of money every year to hope for a positive return on their investments.

2. BUILDING AN ALTERNATIVE SYSTEM

2.1. INTRODUCTION

The following part of this work describes a preliminary draft of a new system designed with different assumptions and operating in a different way than the systems of today. The system that is being described here is possible to implement using existing technology and resources.

2.2. PROJECT ASSUMPTIONS

In the beginning stages of the system's existence the client side of the system (hereinafter the Client X) will be installed on computers of voluntarily collected group of people. The main task of this element of the system is to gather various data about the users and their activities on the internet.

The central part of the system (hereinafter the Server Z) is responsible for receiving data from users' computers and analyzing them. An additional task for the Server Z is to manage the advertisements that are present in the system. These ads are assigned to a number of categories which makes it easier to deliver them to end users.

The essence of the system's work and its innovation at the same time is the method of tying together three different types of data.

The first one is the data about the users themselves (their demographic traits, types of websites they visit, patterns of behavior, momentary trends of these behaviors) which is used to create a profile for each user that describes their behavior and the way they use the internet.

The second one is the data about the websites that users visit. Data such as: the category a given website can be assigned to, the marketing weight of the website (websites that are directly selling something have the highest weight while others that are focused on completely non-sales goals have the lowest weight), website's popularity and the most common demographic traits of the users who are the most common visitors of the website.

The third one is the data about the advertisements. Each advertisement stores some historical data about its efficiency, number of clicks and CTR (Click Through Rate). Each advertisement is also identified by its format (e.g. text, graphic) and the category it is assigned to.

These three types of data are tied together when the system compares the database of advertisements with the profile of the current user and the website they are browsing at the moment. Other piece of data that is being taken into account is the information about the previous websites that have been visited by the user. After analyzing all this data the system can deliver an advertisement that has the best chance of being noticed and clicked on by the user. That is because the advertisement is closely related to the user's current state of mind and online activity.

3. THE SYSTEM'S STRUCTURE

3.1. INTRODUCTION

This description has been divided into two parts. The first part describes the physical construction of the system, consisting of two elements: the Client X and the Server Z. The second part describes the construction of the Server Z itself, its main elements and methods of operation.

3.2. CLIENT X AND SERVER Z – MAIN ELEMENTS OF THE SYSTEM

Figure 1 presents a simple diagram of the system.

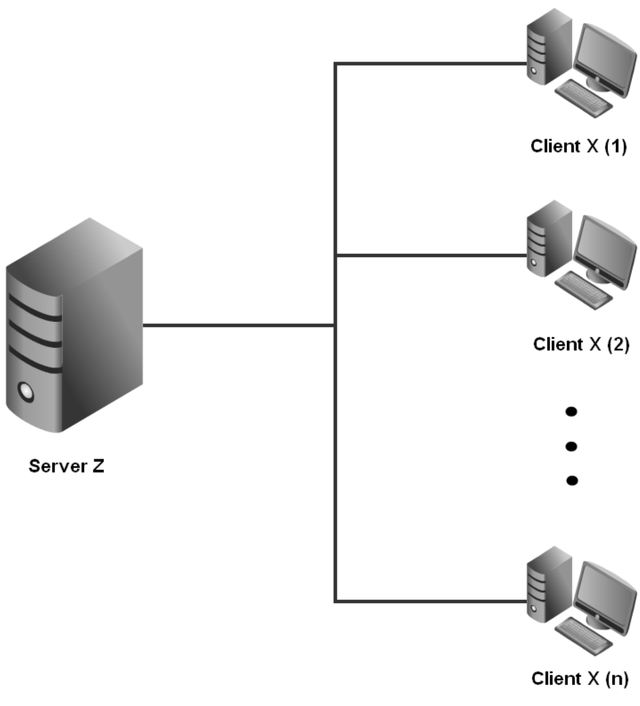


Fig. 1. Two main elements of the system

Client X is constructed as a plugin for a popular web browser – Mozilla Firefox. The user is notified and aware of what operations the plugin is performing. Furthermore, the plugin can be deactivated at any time. The plugin gathers data about the websites that the user is visiting, processes this data, compresses it and sends them to the Server Z. The data itself is stripped from any elements that might be used to identify a user, so they always remain anonymous to the extent that is possible. Another task of the Client X is to receive a specific advertisement from the Server Z and display it to the user in a way that the Server Z marks as the most effective. Finally, Client X registers the fact if the user clicked on the ad and sends this information back to the Server Z.

Server Z is the place where all incoming data is stored and where all operations of analyzing and processing this data take place. Server Z is responsible for creating demographic profiles of users, profiles of websites, advertisements and their categories. Server Z is available via internet browser as an online application. This application makes it possible to send new ads to the Server Z, manage existing campaigns, and analyze the statistical data about the system's efficiency. The reports generated by Server Z can be used to further improve the whole system.

3.3. ELEMENTS OF SERVER Z

Figure 2 presents a simplified model of interaction between the main elements of Server Z.

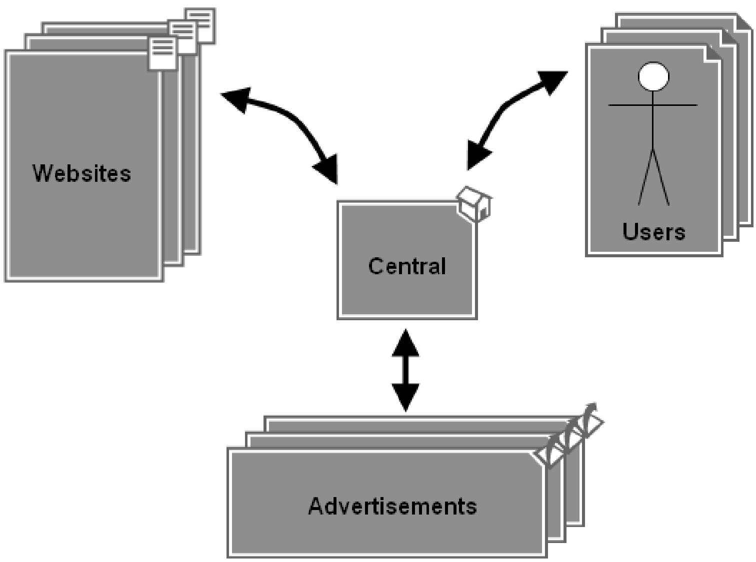


Fig. 2. Main elements of Server Z

There are four main elements of Server Z: websites module, advertisements module, users module, central module.

The websites module groups all encountered websites in predefined categories depending on the topic of the website. Furthermore, each website has a weight assigned to it depending on its commercial potential. On one side there are sites like single product pages in e-commerce stores and other typical sales pages – having the score of 10. On the other side there are sites without any commercial or sales potential, such as entertainment sites or social networks.

The advertisements module is responsible for managing every ad that is stored within the system. Every ad is assigned to a category. These categories describe the topic of the ad (e.g. guitar, computers), its business type (e.g. branding, direct response), type of the offer (e.g. product sale, mailing list subscription, free offer). Each ad is also described by its format (e.g. text, graphic) and its history in terms of efficiency (i.e. number of clicks, views, CTR). All this data is analyzed to improve the effectiveness of the system and the ads served by it.

The users module is responsible for gathering data about users and analyzing these data further along the way. The last stage is the creation of user profiles. These profiles are created based on things like: the history of websites visited by the user, the amount of time spent on each of them, types and categories of the sites visited. Having the profiles of common web activities the system can try to assign a certain set of demographic traits to the user, which gives a more complete view. Because of all the data that have been gathered the system can present the user with a highly targeted advertisement in the perfect-timed moment when the ad has the best chance of generating a conversion.

The central module is responsible for analyzing all the data that has been gathered in the system. This module matches the advertisements present in the system to the newly created user profiles. This is done by calculating a value called the probable success factor. This factor is a numerical value describing the degree to which a given ad is congruent with the user, their profile, and their current situation. While the user visits more websites the probable success factor is calculated for many ads simultaneously, and when it reaches a certain threshold for a given ad, this ad is displayed to the user. The user clicks the ad, or not, and the feedback about this fact is sent back to the Server Z, so the historical data can be updated, eventually resulting in the improvement of the whole system.

4. POSSIBLE PROBLEMS AND ISSUES

The method of operation of this system is a new concept and it hasn't been yet implemented in any of the popular systems of today. Many problems may come up to the surface while creating and testing the system, both in terms of technology and business. These problems and issues may, of course, stand in the way of the system's final success, but such work is a big contribution into the whole market of internet advertising and is surely worth pursuing as it will eventually improve the quality and results of modern advertising systems.

One of the first challenges is the recruitment of voluntary users that will be the subject of the beginning stages of the development. These are the people that will install the Client X plugin on their computers. It is important for these people to be a representative group consisting of individuals of different characteristics and different patterns of spending time on the internet. Only then satisfactory results can be hoped for. The other, probably not so challenging task is to convince a number of advertisers to upload their ads to the system and enable them to be displayed to the users. This should be easier because the advertisers will not be paying any money for this (in the developing stages of the system) which means that they will receive free exposure for their marketing campaign on the screens of many internet users.

Having significant number of users and ads is crucial for the development of the system. The bigger the size of both of these groups the better the results will be.

5. CREATING THE SYSTEM – MAIN STEPS AND MILESTONES

The first part of the system to be created is the Client X. Its capabilities and efficiency, as well as the ability to gather and transfer compressed packages of data back to the server is going to be one of the main concerns on every stage of development. Server Z is the next element in line. The most important thing when it comes to Server Z is the ability to contact and transfer data with multiple instances of Client X at the same time. When it comes to the specific elements of Server Z, the websites module and the users module will be created first. It is the interaction between these two elements that will result in the creation of the user profiles. The final module – advertisement module will be created as the last one, and then all three modules will be connected with each other to create one complete system.

Next phase is to keep the system running for a significant amount of time to further improve it by taking into account all feedback that has been gathered. It is obvious that the first versions of algorithms, data structures and the principles of operation will not be the ones that will bring the best results, therefore gathering feedback and testing new options is one of the most important actions to take. In the final version Client X will be replaced with a more advanced and practical software form than a Firefox browser plugin. This software form is yet to be decided upon.

The system itself is a challenge in terms of technology, business aspects, and the relationship with participating users (at least at the beginning stages). Nonetheless, the final result can be significant for the market as a whole and the people spending big amounts of money on internet advertisements each day. That is why this kind of undertaking is worth pursuing. Similar projects are probably already being developed behind closed doors in labs of Google and other major players, but the results of their work are rarely shared with the public, especially when it comes to new and innovative projects. That being said, the way that the system is built does not exclude the possibility of integrating it with Google's or Yahoo!/Microsoft's systems.

REFERENCES

[1] *First banner ad ever in the world. AT&T Hotwired.*, http://thelongestlistofthelongeststuffatthelongestdomainnameatlonglast.com/first66.html

[2] *The Short, Inglorious History of Online Advertising,* http://www.fastcompany.com/magazine/139/click-here.html

[3] *AdWords*, http://en.wikipedia.org/wiki/Adwords

[4] *2010 Financial Tables - Investor Relations – Google*, http://investor.google.com/financial/tables.html

[5] KRÓL K., *Evolution of Modern Internet Advertising, or Practical Implementation of a Complex Data-Driven System for Gathering and Analyzing Multiple Streams of Incoming Data*, Information Systems Architecture and Technology – IT Models in Management Process, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2010, 179–190.

[6] *Do People Hate Advertisements (And What Can We Do About It)?*, http://newinternetorder.com/do-people-hate-advertisements-and-what-can-we-do-about-it/

[7] *Banner ads tenth birthday!*, http://adland.tv/content/banner-ads-tenth-birthday

IBM, Rational, IT University Competence Center
Jazz Environment, IBM Requirements Composer,
IBM RequisitePro

Cezary ORŁOWSKI*, Paweł MADEJ**, Łukasz SZCZYGIELSKI***

CREATING A MODERN EDUCATIONAL SYSTEM BASED ON IT COMPETENCE CENTERS

This work indicates the need to support University Competence Centers. These are specialized centers which aim to support teaching and the development of the latest software technologies, educating students, PhD students and scientific workers and also raising the competitiveness of the university on the higher education market. The work presents the characteristics of: IBM Rational RequisitePro, IBM Rational Requirements Composer, IBM Rational Team Concert as systems to manage requirements around the product development cycle.

1. CHANGES IN INFORMATICS

Deep changes occurring in IT, which rely on the transformation from fundamental applications and the IT infrastructure supporting them to treating software as a SaaS service and the IT infrastructure as an IaaS service, force radical changes in computer science education at technical universities. These changes, although they have a strategic dimension, require systematic, consequent operational efforts, from the point of view of the university, which are aimed at improving the quality of the education process. These improvements are realistic thanks to bringing some external instru-

* University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk,
e-mail: Cezary.Orlowski@zie.pg.gda.pl
** University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk,
e-mail: Pawel.Madej@zie.pg.gda.pl
*** University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk,
e-mail: Lukasz.Szczygielski@zie.pg.gda.pl

ments to the system generating the internal syllabus to upgrade both the preparation of syllabuses and their further support. An element of these activities is the creation of, in cooperation with IT companies which set the standards of changes in the market, University Competence Centers.

These University Competence Centers are specialized centers which aim to support teaching and the development of the latest software technologies, educating students, PhD students and scientific workers and also raising the competitiveness of the university on the higher education market.

This work is based on the experience of working in the IBM University Competence Center in Gdańsk.

2. DIDACTIC ACHIEVEMENTS – ESTABLISHMENT OF THE UNIVERSITY COMPETENCE CENTER

December 2010 – signing of an agreement for establishing the IBM Rational Competence Center. This center was established stage by stage, to support educational processes:

- Signing of the IBM Academic Initiative agreement: software selection, employee training, academic classes, training programs for students – 2007.
- Agreement extension, software reviews by the Department of Information Technology Management employees, collective scientific publications, training programs for students connected with writing Master's theses – 2008.
- New forms of training programs for students – participation in transparent software development for IBM (academic and IBM mentors) – 2008.
- Establishing a research laboratory at Gdansk University of Technology – Ministry of Science and Higher Education and IBM funds – 2008.
- Conducting of collective business training for companies, among others: PSE Operator S.A. – 2009.
- Signing an agreement to establish the University Competence Center – 10 December 2010.
- Applying the University Competence Center employees' knowledge and competencies in academic courses: Management and Information Technology and Econometrics.

3. RANGE OF SERVICES OF THE IBM UNIVERSITY COMPETENCE CENTER

Established in cooperation with the Technical University of Gdansk and IBM, the Competence Center provides support for students, PhD students and the education of scientific workers in the following fields:

- IBM Rational (main field of competence),
- Project Management,
- Other IBM software technologies.

The Faculty of Management and Economics delegated four representatives, who were also trained by IBM, to run the UCC. A collaboration with IBM was established in the areas of projects hosted by IBM – the integration of models and tools for supporting software development life cycles. Classes were conducted using IBM Rational courseware (Rational Unified Processes, Rational Method Composer, Rational Team Concert). All IT-based subjects were verified and modified to obtain authorization from IBM in order to give students IBM Qualification Program pre-certificates. There were preparations for establishing own environment based on IBM CloudBurst, which will also be used for educational purposes.

A Students' Research Group on Smarter Planet hosted by the Faculty of Management and Economics and mentored by Mr M. Sc., Eng Pawel Madej was established in order to support the UCC and co-operate with IBM. Mr Bartosz Chrabski, who is an expert in Rational software products is the IBM mentor, helping to develop the University Competence Center.

Additionally, IBM supports the UCC in planning the proper use of their courseware during classes. The company also helps with the process of verifying and modifying subjects in order to give students IBM Qualification Program pre-certificates.

4. IBM UNIVERSITY COMPETENCE CENTER – REALIZED PROJECTS

Ongoing work in the UCC relates to the integration of tools around the product development cycle. One of the projects concerns integrating tools in managing requirements.

4.1. CHARACTERISTICS OF IBM RATIONAL REQUISITEPRO

Rational RequisitePro helps project teams to manage their requirements, to write good use cases, to improve traceability, to strengthen collaboration, to reduce project rework and to increase quality.

- Avoid rework and duplication using advanced, real-time integration with Microsoft® Word
- Manage complexity with detailed traceability views that display parent/child relationships
- Mitigate project risk by displaying requirements that may be affected by upstream or downstream changes of requirements
- Achieve collaboration for geographically distributed teams through fully functional, scalable Web interface and discussion threads
- Capture and analyze requirements information with detailed attribute customization and filtering
- Improve productivity by tracking changes using project version comparisons with XML-based project baselines
- Align business goals and objectives with project deliverables through integration with multiple tools in the IBM Rational software development and delivery platform
- Operating systems supported: Windows family [7]

4.2.CHARACTERISTICS OF IBM RATIONAL REQUIREMENTS COMPOSER

Many stakeholders in the development process create and use requirements to drive business goals. Better quality requirements, effective management and a good process lead to a reduction in rework, faster time to market, a reduction in costs, and an overall better business outcome.

- Engage a wide range of globally distributed customers and other stakeholders in a requirements driven development process.
- Use intuitive, rich online documents, storyboards, process diagrams, use cases, and other visual techniques to capture customer needs and drive your business.
- Analyze, organize and manage requirements and their changes efficiently using attributes, collections, tags, filters, in dynamic information views or use out-of-the-box templates and custom reports to understand a project's position (including requirements specifications, audit history, traceability reports).

- Encourage group collaboration across development projects combining team expertise to improve clarity and accuracy in requirements for stakeholders, developers, managers, etc.
- Move beyond office documents, spreadsheets, work items, or note cards to express and manage requirements.
- Bring agility, customer focus and predictability with light-weight requirements practices.
- Expose requirement and development gaps or change the impact using traceability, which provides visibility across requirements to implementation and testing.
- Reuse requirements artifacts across an enterprise or program.
- Align requirements and development, change management and quality management activities through Collaborative Lifecycle Management (CLM) by relating requirements from the Requirements Composer to work items in IBM Rational Team Concert, and tests in IBM Rational Quality Manager.
- Where teams are capturing, organizing, and planning requirements in a work item backlog, elaborate these work items with visual and textual notations to accelerate communication with and commitment from the customer or product owner.
- Rational Requirements Composer 3.0 delivers a powerful combination of definition, management, traceability, templates, history, reviews and approval, task management, planning, shared filters and views, a customizable dashboard, reporting and shared viewlets with Team Concert and Quality Manager capability, using your favorite Web browser. [6]

4.3. COMPARISON: REQUISITEPRO AND REQUIREMENTS COMPOSER

Due to the fact that two applications were analyzed, it was decided to evaluate the suitability of the connection between both.
In the evaluation, the main factors considered were: Word integration, Manage requirements, Req. Lifecycle Traceability, Web-based access, Detailed reporting for standards compliance, Tool integration, Rich text capture, Storyboards, Diagramming and Dashboards.

A scale and rate was attributed to each factor:
1 – none, 2 – medium, 3 – good

Table 1. Comparison: IBM RequisitePro and IBM Requirements Composer

Capability	IBM RequisitePro	IBM Rational Requirements Composer
Word integration	3	1
Manage requirements	3	2
Req. Lifecycle Traceability	3	1
Web-based access	3	2
Detailed reporting for standards compliance	3	2
Tool integration	3	3
Rich text capture	1	3
Storyboards	1	3
Diagramming	1	3
Dashboards	1	3
∑	22	23

The Requirements Composer complements RequisitePro by providing capturing, modeling, diagramming, and collaboration capabilities. Rational Requirements Composer is an application that greatly facilitates the collection of requirements data. Rational RequisitePro manages very well previously gathered requirements. These two applications complement each other. This means that their integration will create a perfect tool to work with requirements. Some features are not implemented in one single tool but work very well in the second tool, which is shown in the table above. This improves the possibility of effective and efficient work with requirements.

4.4. INTEGRATION OF IBM REQUIREMENTS COMPOSER AND IBM REQUISITEPRO

The integration of both tools is done through the communication servers of each of the tools. To work with Requirements Composer, it is necessary to have Apache Tomcat, which serves as a database. However, RequisitePro needs the installation of IBM WebSphere [2] to connect, which acts as a communicator. In order to visualize the connection, a model has been created that illustrates the connection taking place between the tools and servers. The model is shown in the figure below. [1]

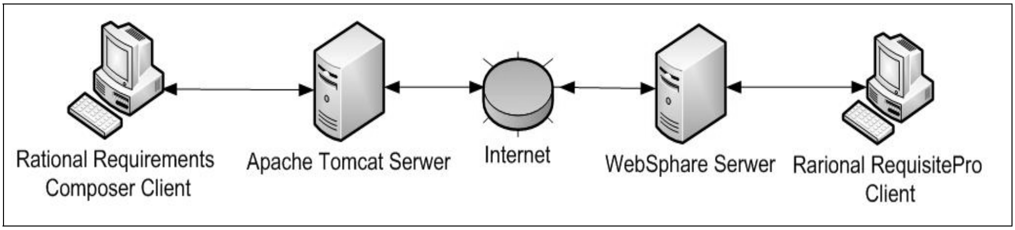


Fig. 1. Model of integrated applications

A sample of the export attributes from Requirements Composer to RequisitePro is shown in figure 2 below.

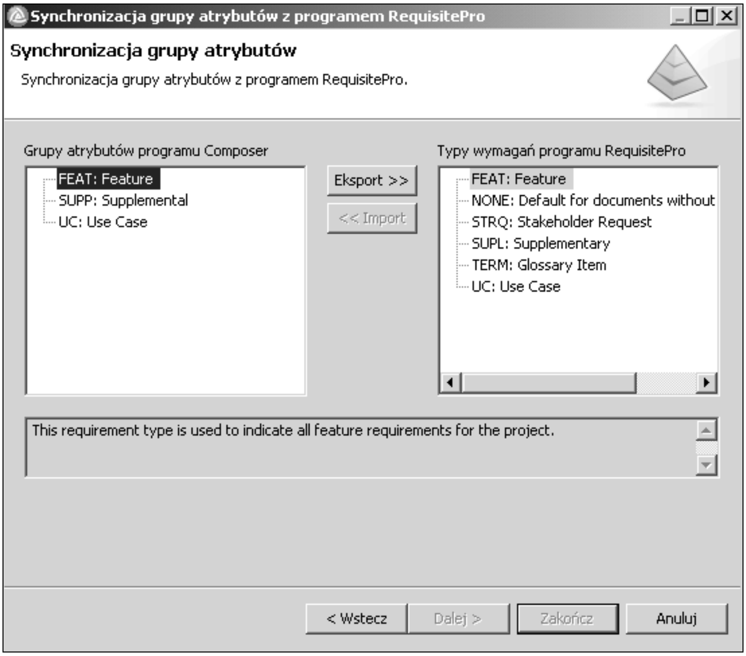


Fig. 2. Sample of export attributes from Requirements Composer to RequisitePro

4.5. PLANS FOR THE FUTURE – INTEGRATION PROJECTS

Proposed improvements to the management of requirements using the system. In the selection of a suitable environment of management requirements and project management around the product development cycle, the tools available on the market were analyzed. It was found that one tool supports processes of this type and this is the IBM Rational Team Concert (RTC). [3]

Characteristics of the Jazz Environment.

The Jazz environment was built using the Eclipse environment. It includes a Jazz application named RTC. It formalizes the management processes and software development carried out by distributed generation teams. This environment is based on client-server architecture in which the access client is the RTC (supplied by IBM as a solution to commerce), and the Rational Team Concert server (based on the architecture of Apache Tomcat or Websphere). For the Jazz environment, the portal cooperation named Jazz.net was created for the exchange of knowledge and information via instant messaging and newsgroups. Within this portal, IBM made available the entire process of documentation of the Jazz environment and presented the history of the previous iteration and the phases of the project. Due to this transparent approach to software development it becomes possible to share the knowledge base and experience for both system developers and customers to implement this system at home. It is possible to recognize the possibility of installing their own plug-in functionality to support environmental technologies management. This idea was called "Open commercial software development".

The Jazz environment supports manufacturers and the management of production. It creates conditions for the organization of work through the assignment of roles and supervision of all project participants. It facilitates the gathering of knowledge from manufacturing and project management. By defining members involved in the project, it defines the teams to which they belong (there may be a team dedicated to the user interface and different to the requirements). The essence of this system is the power of communication in the work of the teams (by selecting Jabber instant messaging or IBM Lotus Sametime). Access to the Jazz platform is possible via a web interface based on Ajax. Methods of communication within the Jazz environment and accompanying team work are shown in figure 3.

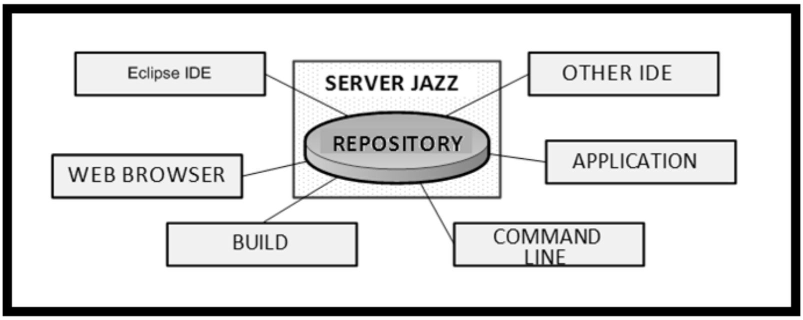


Fig. 3. Infrastructure of the Jazz Environment (for access to the server, Jazz RTC)

The essence of the presented method of the software development team is the possibility of a differentiated approach to the management of manufacturing processes. The templates used, which support methods (Agile and Scrum), provide opportunities for the selection of an IT project template for: the maturity level of the supplier, the entropy level and the client maturity level (previously discussed conditions for the construction of the model). By default, the environment provides the following templates: Eclipse Way, OpenUP, Scrum, Agile Process and Simple Process Team (editable to suit your needs). The template does not require us to follow a route. It shows a set of rules in accordance with a method. The approach to the use of templates in the Jazz environment is presented below. The diagram shows the dependencies between them. This system is not the only one and the structure can be adapted to the needs of the project by adding new components. [5]

The figure below shows a diagram of the integration of applications around the product development cycle.

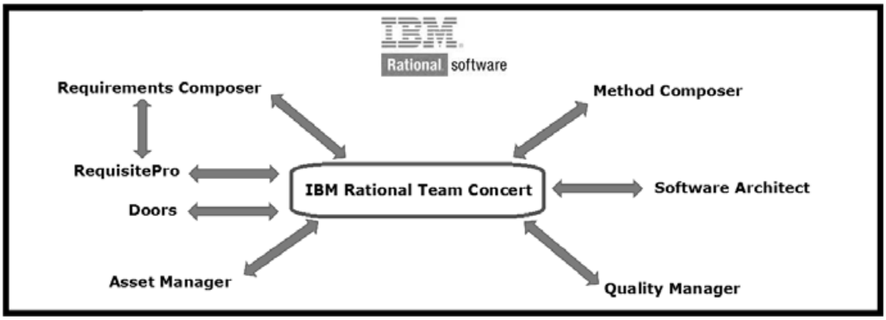


Fig. 4. Model of the integration applications around the product development cycle

5. PLANS FOR THE FUTURE AND EXPECTATIONS – IBM UNIVERSITY COMPETENCE CENTER.

University Competence Centers are expected to be the factor that will change the way business and universities co-operate from static to dynamic, forcing activity from both sides. It will be possible thanks to an accepted model of operating procedures, where many other Competence Centers located at other Universities will operate in the following way:

- the selection of future partners in the form of scientific research units will be managed by interdisciplinary scientific research units
- IT laboratories will be established in order to research concrete environments and co-operate (knowledge and technology centers funded by different sources)
- specific processes will be generated at university level to transform the existing state, where universities are using technologies proposed by IT companies in the

process of education to the state and where universities will become Competence Centers – supporting both business and education. These processes will be supported by: signals from business regarding changes in syllabuses; propositions from universities addressed to business of educational and development programs; the need for obligatory student internships in IT companies; the need for training programs for students connected with writing Master's theses for business purposes; execution of future projects (EU structural funds).

REFERENCES

[1] ZARZYCKI Z., *Play together*, Think! #3, Corporate press, Warszawa 2008.
[2] IYENGAR A., JESSANI V., CHILANTI M., *WebSphere Business Integration Primer*: Process Server, BPEL, SCA and SOA, IBM Press, Boston, USA, 2008.
[3] RADOSIŃSKI E., *Systemy informatyczne w dynamicznej analizie decyzyjnej,* Wydawnictwo Naukowe PWN, Warszawa 2006.
[4] TIWANA A., McLean E.R., *Expertise Integration and Creativity in Information Systems Development*, M.E. Sharpe, Inc. Armonk, New York, USA, 2005
[5] ORŁOWSKI C., ZIÓŁKOWSKI A., MADEJ P., *Information Systems Architecture and Technology*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2010
[6] www.ibm.com
[7] www.jazz.net

financial risk, capital structure, company value

Zofia WILIMOWSKA*
Marek WILIMOWSKI*

RELATION BETWEEN CAPITAL STRUCTURE AND THE COMPANY VALUE

Capital structure is an important factor of financial management of the firm. Capital structure decisions have a strategic meaning and for that should be good prepared and using of debt capital rational and reasonable. This chapter concerns making analysis of capital factor influencing the firm value. Understanding the trade-off that have to be made between financial risk and expected company value is a base to financial decision making. The risk – value trade-off concept is one of the most important principles. Various factors affect the company value growth. The chapter discusses relation between capital structure and company value, which future growth is the main reason for capital structure optimization. The chapter presents analysis of ten domestic companies from the point of view of the considered relation. Web pages were used to collect required financial information.

1. INTRODUCTION

Properly formulated capital structure influences maximization of the capital, therefore optimization attempts can be observed. However financial leverage affect leading to equity capital growth as a result of using debt has been known for a few dozen of years, Polish companies use external long-term debt capital sources to a limited extent. Even though access to finance sources is increasing, unfortunately there is no rational capital structure management. In Polish economy which has been a free market economy for only the last twenty years, a subject matter of capital structure optimization is the key area of interests of many researchers.

---

* Wrocław University of Technology, Institute of Industrial Engineering and Management, Smoluchowskiego 25, 50-372 Wroclaw, Poland; e-mail: zofia.wilimowska@pwr.wroc.pl.

* The State Higher Vocational School in Nysa, Armii Krajowej 7, 48-300 Nysa, Poland; e-mail: mwilim@kn.pl.

Capital structure must be carefully created and skillfully managed, taking into consideration all aspects influencing the debt using efficiency. Many factors influence the external capital use – these are both micro- and macroeconomic. Companies' decisions of external debt capital search are not always comprehensively thought over, often made short-term in a situation of debt using necessity. It is not possible to take into consideration all factors when making a decision of external debt capital use. Particular factors may also influence companies in many different ways, depending on a company's activity type, organizational and legal form or a company's line of business.

Analysis of financial literature gives the idea of how much space is devoted to the issues of financial structure. D. Durand [3] was the precursor in this field, and F. Modigliani and M. Miller also contributed to the development of the theory. We also have to mention the theories and studies of H. DeAngelo and R. Masuliis, N. Baster, Meckling, Myers and Majluf. The works of mentioned authors and many other economists aimed to find the answer to question if there is an optimal capital structure for a company and if this influences the company's market value.

Making capital structure decision it should be remembered that financial leverage increasing causes risk increasing so it cause increase of investors' required rate of return. Capital structure is some compromise between higher shareholders benefits and company's advantage of using cheaper debt capital.

2. CAPITAL STRUCTURE

Capital structure must be carefully created and skillfully managed, taking into consideration all aspects influencing the debt using efficiency. Many factors influence the external debt capital use – these are both micro- and macroeconomic. Companies' decisions of debt capital search are not always comprehensively thought over, often made short-term in a situation of debt using necessity. It is not possible to take into consideration all factors when making a decision of external debt capital use. Particular factors may also influence companies in many different ways, depending on a company's activity type, organizational and legal form or a company's line of business.

2.1. THE VALUE OF A COMPANY AND ITS CAPITAL STRUCTURE

Capital structure is a combination of equity and debt sources of company's financing. There is much publication about optimal capital structure. The researchers were looked for optimal capital structure which maximizes the company's market value. If optimal structure exists that means that is possible get value maximization by capital structure molding.

The development of the theory of capital structure actually dates back to 1950s and the beginnings are associated with the works of Franc Miller and Merton Modigliani [5]. However, it does not mean that attempts to explain the issue were not made earlier. The managers of companies were conscious of the fact that the capital structure influences the value and they were even able to determine the optimal level of it. There were no models though, and to devise the way to manage a structure creating its value. Secondly, there were no instruments to manage the value.

D. Durand's work is considered to be the first comprehensive attempt to explain the influence of capital structure on the value of a company. It was published in 1952 and it really made use systematized the available knowledge in the area. Most of contemporary researchers do not consider Durand's work an acceptable way of explanation. In spite of this, the work is a valuable contribution to the issues of capital structure and definitely worth mentioning here.

The concepts used then to valuate companies were the starting point of Durand's work. The conclusions he arrived at depended much on the method of valuation he used. The application of the concept of net profit inclined him to recognize the positive effect of indebtedness on the value of companies. Operating income method, on the other hand ruled out any kind of influence of capital structure on valuation. Thus a more critical reference to these concepts was necessary.

Durand as a result created his own approach to the problem and called it the flexible approach. It assumes that, in case of moderate use of loan capital (the particular financial leverage); its lower cost will not be balanced by the increase of financial risk connected to discharging of indebtedness to finance a business. In this approach the value of a company will rise with bigger use of indebtedness in capital structure until an optimal point. Upon reaching the point, the advantages of bigger application of financial leverage will be balanced by increased risk of use of loan capital. This situation is presented in fig. 1. The chart shows, that the increase of indebtedness initially results in an increase of company value because of the fall of value of weighted average cost of capital.

The increase of indebtedness over an optimal level however may lead to lowering the market value of a company. In case of crossing an optimal point such increase of equity capital that will result in an increase of weighted average cost of capital. It is related to shareholders' demand of increased return on investment. The increasing demands causes from the rising financial risk, which is a result of bigger use of indebtedness.

In spite of the valuable contribution of Durand's work to the theory of capital structure, the Modigliani and Miller's model from 1958 is considered the first and

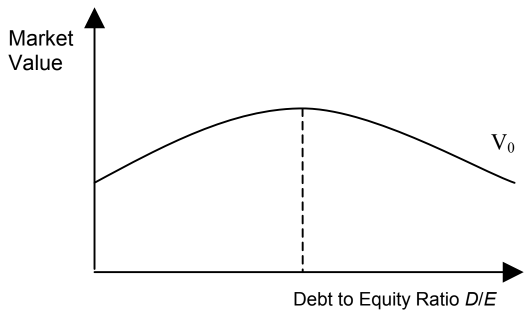


Fig. 1. Traditional approach – the company's market value and financial leverage

fully scientific method of explaining the issue of influence of company's capital structure on its value. In fact, it is a theoretical denial of the influence of capital structure on the value of companies. The model referred to a condition of market without taxes. Only its comparison to actual economical world made its authors change their views on the influence of capital structure on the value of companies. In 1963 upon taking tax into account the authors realized, that there is a possibility to optimally use loan capital and maximize the value of a company at the same time. The optimization results from lowering tax base with interest on debt and lack of such effect of dividends paid to shareholders. The model however does not mention the increasing risk that comes with the bigger use of indebtedness. Miller noticed this apparent oversight and presented his own version in 1977. He took under consideration the income tax from natural and legal persons. Such modification of the MM model proved that parallel introduction of tax on both corporate and personal income eliminates the positive influence of financial leverage on the value of a company. With this theory, Miller confirmed the original 1958 model.

Modigliani himself decided to deal with the original MM 1958 model. His conclusions, however, were completely different from Miller's. In his considerations he dealt with the previously not considered aspect of inflation on capital structure. His model's primary conclusion was the mutual linear dependence between inflation rate and the effect of tax shield. Moreover, according to Modigliani, it is impossible for both factors to have the zero value at the same time. Especially, in the case of tax shields, they were considering personal and corporate tax. In his opinion, the use of indebtedness in the structure of a company's capital will influence its value.

The compromise theory is a supplement of Modigliani and Miller's theory. It is a synthesis of both MM models. It recognizes the legal and natural persons' tax, taking under consideration lowering the interest tax shields resulting from the difference between the rate of income tax paid by legal and natural persons. In compromise theory the non-interest tax shields diminishing the necessity to use high tax shields are considered.

As a result of merging of the abovementioned theories the following conclusions have been expressed:

- When companies are not sure whether the use of high tax shields is profitable, they will not be able to pay high interest on the indebtedness.
- There are different kinds of tax shields in different companies
- The companies which have high variability of income and often have low income should borrow less from those, which have high and constant income.

Theories from the areas of Modigliani and Miller's models do not take under consideration some others factors: bankruptcy cost, agency costs and so on. So the MM models were developed by other authors.

For example the theory of competitiveness that is strongly related to constantly developing and automatic behavior of capital market. It is created by the intensive processes of merging and overtaking of companies which took place in the late 80s in the United States. The scale and range is now no smaller than it once used to be, so we can easily assume that the theory is still valid and gaining in importance in management of value of stock companies all over the world.

The theories in question, in spite of different approaches to the influence of capital structure seem to confirm the influence of it on the value of a company. It can thus be concluded that in theory and in practice it is accepted that the changes in capital structure can work in favor of the value of a company. Most importantly it is indicated that the bigger use of indebtedness can influence the increase of the value of a company, however it can increase the risk of the company at the same time and financial risk especially. Financial risk it is the risk that the firm will not generate sufficient sales revenue to finance cost of financing, for example interest of a long-term debts. Some investors have weaker claims than others do because they are poorer collateral. Collateral risk depends of types of investors – what kind of capital they own. These components of specific risk affect the overall risk of the firm.

One of the risk measures is beta ratio that shows the dependency between rate of return of the share that is equal cost of capital which depends on capital structure, and the market rate of return, fig. 2.

The considered dependency is a linear regression function,

$$R_i = \alpha_i + \beta_i R_m + \varepsilon_t , \tag{1}$$

where:

R_i – rate of return of share *i*,

R_m – market rate of return,

α_i , β_i – structural parameters,

ε_t – random variable.

A random element shows the influence of other parameters – internal and external factors – on share price and rate of return. The beta parameter is an indirect, market risk measure. A random element shows the influence of other parameters – internal and external factors – on share price and rate of return. The beta parameter is an indirect, market risk measure.

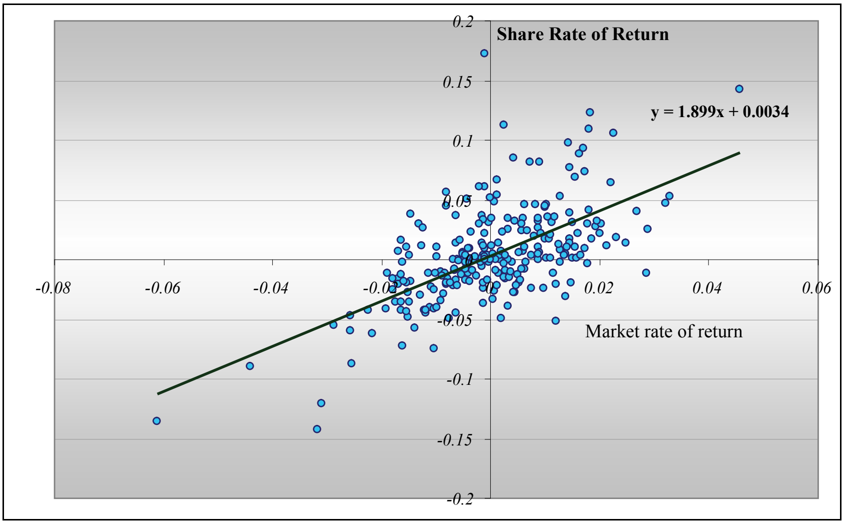


Fig. 2. The dependency between market rate of return
and MCI stock rate of return (2007 year)

Risk is a result of uncertainty of future values quantity causing company value growth. Uncertainty is a feature of the reality that represents impossibility of describing future events exactly. Its sources are objective factors effecting by situation variability and complexity and subjective causing by personal intellectual processes.

3. RESULTS OF RESEARCHES

Ten companies were examined, tab. 1, in the years 2005–2009 to describe the relation between capital structure and company value. The companies financial statements that were delivered from companies web pages, were used to prepare the required evaluations.

Researches of dependency between company value and capital structure were made by capital structure calculation and company value evaluation using computer program MS EXCEL.

3.1. METHODOLOGY OF THE RESEARCH

The research of dependency between capital structure and company value requires some measure definitions that describe capital structure and the company value. In the presented research capital structure is measured by two ratios: capital structure ratio D/E and total debt.

$$\text{Capital structure ratio} = \text{D/E} = \frac{\text{equity}}{\text{debt capital}} \tag{2}$$

A measure of capital structure is ratio that shows the relation between equity capital and debt capital, it means it takes into consideration long-term debt and short-term debt. The ratio is especially important for enterprises dealing on Polish market because the enterprises like to use short-term financing especially.

Total debt is understood as

$$\text{total debt} = \text{short term debts} + \text{long term debts}, \tag{3}$$

Value of the company EV was evaluated by market capitalization method

$$\text{EV} = \text{Market capitalization} = \text{average stock price} * \text{number of stocks} \tag{4}$$

Table 1. Examined Companies

Lp.	Name	Branch
1	Netia S.A.	Telecommunication
2	Bioton S.A.	Pharmacy
3	Ceramika Nowa Gala S.A	Construction material
4	Police S.A.	Chemical industry
5	Travelplanet S.A.	Retail
6	Atlanta S.A.	Wherehousing
7	Orbis S.A.	Hotel and restaurance
8	Wilbo S.A.	Food
9	Elektrociepłownia Będzin S.A.	Energy
10	Lentex S.A.	Chemical artificial material

Examined companies present various branches of industry.

3.2. RELATION BETWEEN FIRM VALUE ON CAPITAL STRUCTURE

The results* are shown in tables 2 and 3 and the figures 3–13.

---

* On the base: Leżoń E., *Zarządzanie strukturą kapitałową w przedsiębiorstwach działających na polskim rynku, praca magisterska*, Politechnika Wrocławska, Wydział informatyki i Zarzadzania, promotor Z. Wilimowska, Wrocław 2011

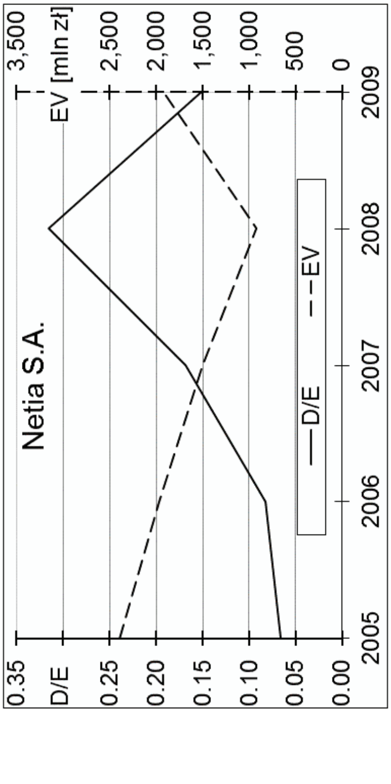


Fig. 3. Netia S.A. – Relation between EV and D/E

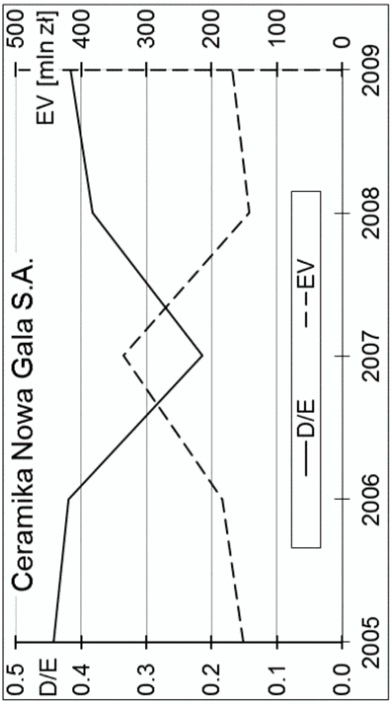


Fig. 5. Ceramika Nowa Gala S.A. – Relation between EV and D/E

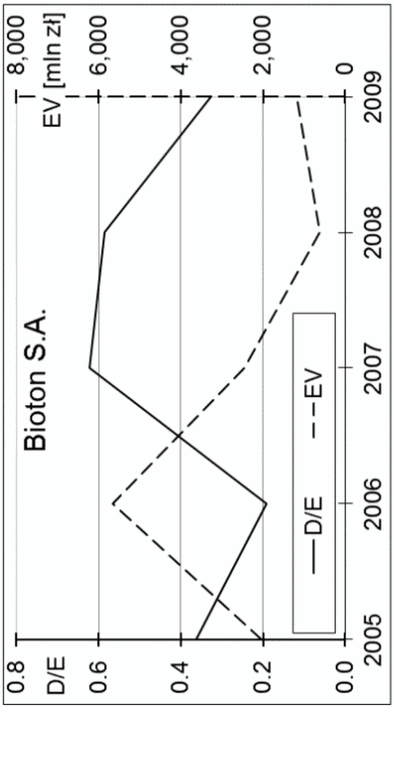


Fig. 4. Bioton S.A. – Relation between EV and D/E

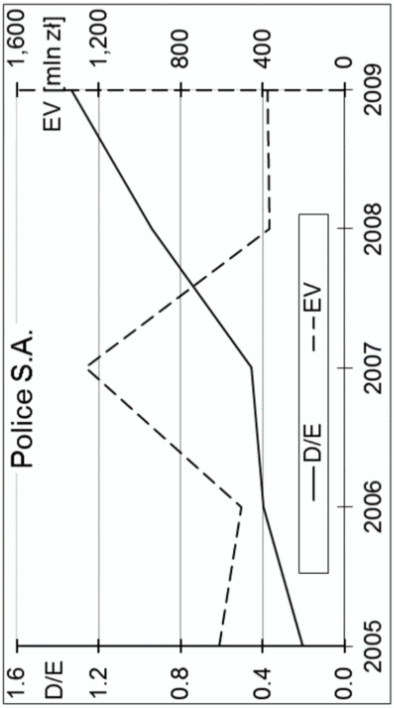


Fig. 6. Police S.A. – Relation between EV and D/E

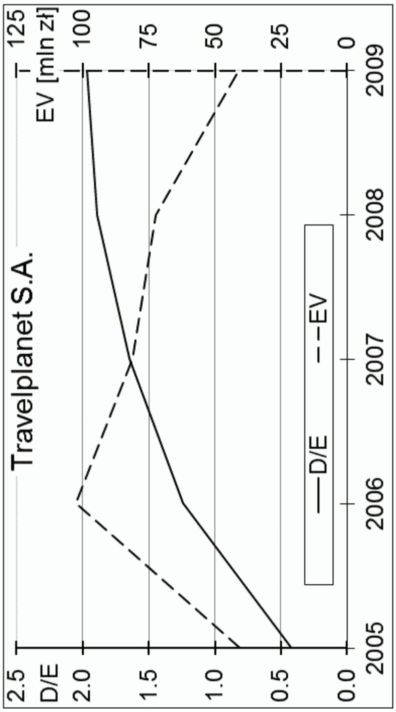


Fig. 7. Travelplanet S.A. – Relation between EV and D/E

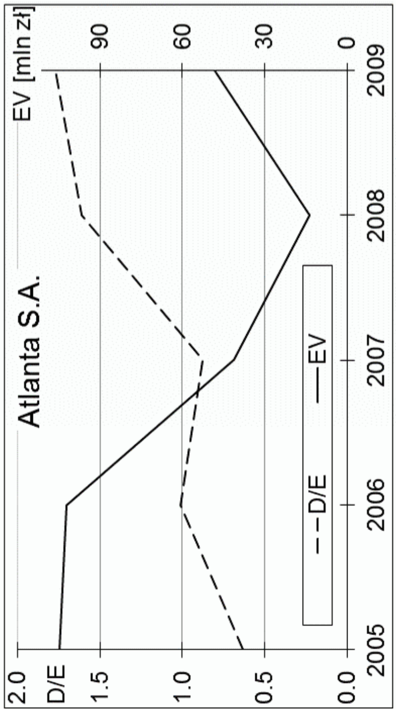


Fig. 8. Atlanta S.A. – Relation between EV and D/E

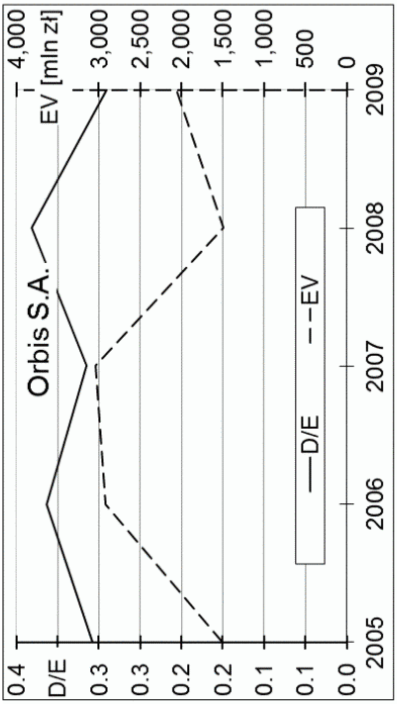


Fig. 9. Orbis S.A. – Relation between EV and D/E

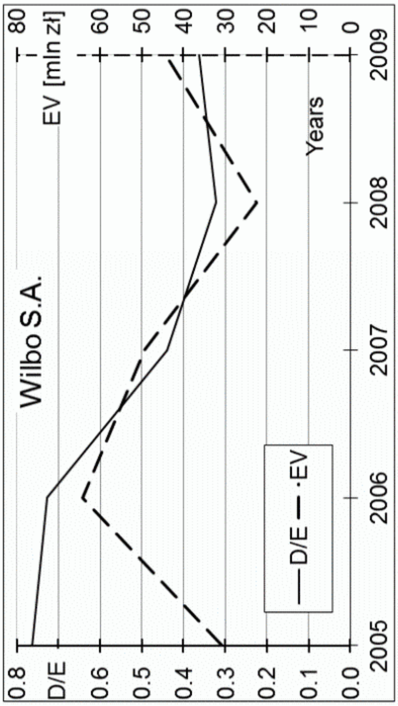


Fig. 10. Wilbo S.A. – Relation between EV and D/E

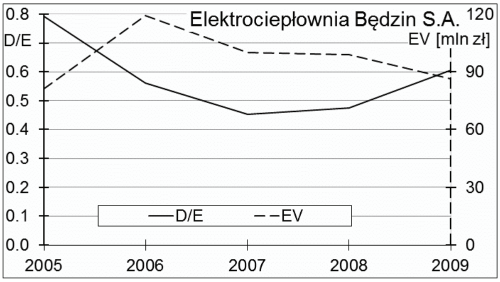


Fig. 11. Będzin S.A. – Relation between EV and D/E

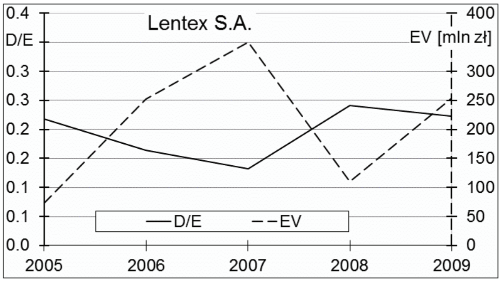


Fig. 12. Lentex S.A. – Relation between EV and D/E

On the base of prepared analysis there can be concluded that there is no unequivocal answer of the relation between capital structure and company value. In some situation there can be observed growth of company value when capital structure is increasing and sometimes the relation is opposite.

In tables 2 and 3 there is presented summery results of the research.

Table 2. The summary results of the research

Company	Netia S.A				Bioton S.A				Ceramika Nowa Gala S.A.				Police S.A.				Travel planet S.A.				Atlanta S.A.				Orbis S.A.				Wilbo S.A.				Będzin S.A.				Lentex S.A.			
Year 200…	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9	6	7	8	9
D/E	↑	↑	↑	↓	↓	↑	↓	↓	↓	↓	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↓	↑	↓	↓	↓	↓	↑	↓	↓	↓	↑	↓	↓	↑	↓
EV	↓	↓	↓	↑	↑	↓	↓	↑	↑	↑	↓	↑	↓	↑	↓	↑	↑	↓	↓	↓	↓	↓	↓	↑	↑	↑	↓	↑	↑	↓	↓	↑	↑	↓	↓	↓	↑	↑	↓	↑

Table 3. The effect of influencing the capital structure on the company value

Effect	number of effects				Σ
	2006	2007	2008	2009	
D/E increases and values increase	2	1	0	4	**7**
D/E increases and value decreases	3	3	7	2	**15**
D/E decreases and value decreases	0	3	3	0	**6**
D/E decreases and value increases	5	3	0	4	**12**

The results of the research show that there is a not unequivocal relation between capital structure and company value. But mostly when D/E is increasing the value of the company is falling (table 3). It means that companies use too much debt capital and capital structure is not optimal – is too risky. But there are some companies when D/E increasing generates growing of the company value. It depends on branch of the company.

4. CONCLUSION

So, managing capital structure the trade-off between financial risk and possibility of company growth should be taken into consideration. More debt capital usage can generate additional profit because debt capital is cheaper than equity capital, but using debt capital increases the financial risk. As was mention before, in this approach the value of a company will rise with bigger use of indebtedness in capital structure until an optimal point. Upon reaching the point, the advantages of bigger application of financial leverage will be balanced by increased risk of use of loan capital.

REFERENCES

[1] Copeland T., Koller T., Murrier J., Wycena: mierzenie i kształtowanie wartości firm, WIG-Press, 1997.
[2] Davies E. P., Debt, Financial Fragility and Systematic Risk, Clarendon Press 1992
[3] Durand D., Cost of Debt and Equity Funds for Business. Trends and Problems of Measurement. Conference on Research in Business Finance, New York, National Bureau of Economic Research, 1952.
[4] Leżoń E., Zarządzanie strukturą kapitałową w przedsiębiorstwach działających na polskim rynku, praca magisterska, Politechnika Wrocławska, Wydział informatyki i Zarzadzania, promotor Z. Wilimowska, Wrocław 2011.
[5] Modigliani F., Miller M. H., The Cost of Capital, Corporation Finance and the Theory of Investment. The American Economic Review, June 1958, vol. 48, nr 3.
[6] Wilimowska Z., Wilimowski M., Sztuka zarządzania finansami, cz. 1, OPO, Bydgoszcz, 2001.
[6] Wilimowska Z., Lewiński vel Iwański S.: Badanie wpływu struktury finansowej na wartość spółki. Zeszyty Naukowe. Finanse, Rynki Finansowe, Ubezpieczenia (Uniwersytet Szczeciński). 2009, nr 16
[8] www.netia.pl
[9] www.bioton.pl
[10] www.nowa-gala.pl
[11] www.police.pl
[12] www.travelplanet.pl
[13] www.atlantapoland.com.pl
[14] www.orbis.pl
[15] www.wilbo.com.pl
[16] www.ecbedzin.pl
[17] www.lentex.com.pl
[18] www.money.pl

PART III

INFORMATION SYSTEMS DESIGN PARADIGMS

systems and software engineering, information systems, system development methodologies, software development methods, small and medium IT companies

Kazimierz CHOROŚ*
Michał MAJEWSKI*

INFORMATION SYSTEM AND SOFTWARE DEVELOPMENT METHODOLOGIES AND METHODS IN SMALL AND MEDIUM IT COMPANIES

In computer engineering a system development methodology and software development methods are frameworks that are used to structure, plan, and control the process of developing and maintaining an information system. The effective design process of information systems requires a formalized methodology for building information systems. Many frameworks have been developed and are used in practice, for example a linear framework called waterfall, an iterative framework such as prototyping or rapid application development, a combined linear-iterative frameworks such as incremental methodology or spiral methodology, and extreme programming as well as object-oriented development methodologies and agile software development methodologies.

The chapter will present the advantages and disadvantages of the methodologies used in software companies as well as their popularity. The data have been obtained in a survey conducted among IT firms. It seems evident that agile methodologies are becoming more and more popular. It is due to the fact what it was strongly expressed by the IT firm managers that the client does not understand design diagrams and is mainly interested in system interface construction of a functional prototype. Furthermore, the most surprising observation is negligent attitude towards system documentation.

1. INTRODUCTION

Software engineering is a domain of computer and information science concerned with the ways to design, implement, and modify software so that it is of high quality, user-friendly, affordable, maintainable, and fast to build and finally that it meets user

* Wrocław University of Technology, Institute of Informatics, Poland, 50-370 Wrocław, Wybrzeże Wyspiańskiego 27, e-mail: kazimierz.choros@pwr.wroc.pl, michal.majewski@pwr.wroc.pl

requirements. Software engineering is a systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software.

Systems engineering deals with the overall system design, dealing also with physical aspects which include hardware design as well as with financial and organizational aspects, negotiations, time schedules etc.

The chapter is organized as follows. The next section describes the main characteristics of system development methodologies and software development methods. The bibliography in these areas is very extensive. The section 3 presents software companies interrogated in a survey. All queries used in a survey are cited. The section 4 presents and discusses the survey results. The traditional design methodologies similar to that used in architecture or machine constructions are widely applied as well as new approaches leading to the acceleration of systems creation. The final remarks and the future research work areas are discussed in the last section 5 of this chapter.

2. SYSTEMS AND SOFTWARE ENGINEERING

Since the beginning of computer system implementations, there have been a proliferation of system and software design methodologies. Different methodologies have been proposed and developed to resolve different types of problems in different real environments. A methodology is defined as a set of procedure that we follow from the beginning to the completion of the system development process. The nature of the methodology is dependent on a number of factors, including the system development environment, the organization's practices, the nature or type of the software being implemented, the requirements of the users, the qualification and training of the system analysts and of the software development team, the available hardware and software resources, the availability of computer-aided packets for designing and implementation, and even the budget and the time schedule.

2.1. SYSTEM DEVELOPMENT METHODOLOGIES

The system development methodologies can be oriented on different aspects, so, we have process-oriented methodologies, blended methodologies, object-oriented methodologies, rapid development methodologies, people-oriented methodologies, organizational-oriented methodologies, and frameworks [2].

Typical activities in system development process include the following [9]:

- system conceptualization,
- system requirements and benefits analysis,
- project adoption and project scoping,

- system design,
- specification of software requirements,
- architectural design,
- detailed design,
- unit development,
- software integration and testing,
- system integration and testing,
- installation at site,
- site testing and acceptance,
- training and documentation,
- implementation,
- maintenance.

2.2. SOFTWARE DEVELOPMENT METHODS

The IEEE Computer Society defines software engineering as [11]:

(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

(2) The study of approaches as in (1).

Software engineering is can be organized into ten different knowledge areas [1]: software requirements, software architecture, software design, software development, software testing, software maintenance, software configuration management, software engineering management, software development process, software engineering tools and methods, software quality.

This variety in methodologies, the multiplicity of activities in system development, and multiplicity of knowledge area of software engineering result in diversity of solutions applied in practice in IT companies.

There are many types of system development methodologies used in practice:

- structural methodologies: structural decomposition methods, hierarchical data aggregation, Yourdon implementation model, structured systems analysis and design method, Hartley-Pirbhai modeling;
- object methodologies: RUP – Rational Unified Process [4, 12], Booch methodology, OMT – Object Modeling Technique of Rumbaugh, Code-Yourdon methodology, Shlaer-Mellor method, data fusion methodology, Tylor prototyping;
- social methodologies: SSM – Soft System Methodology, ETHICS – Effective Technical and Human Implementation of Computer System;
- agile methodologies [10]: XP – eXtreme Programming [6], Scrum [18-20], Crystal (Crystal Clear, Crystal Orange) [3];
- Open System Architecture methodology;

- RAD – Rapid Application Development;
- Model-Driven Architecture methodology.

3. COMPANIES AND SURVEY QUESTIONS

The interrogation on system development methodologies as well as on software development methods has encountered many difficulties and obstacles. IT companies are not so willing to reveal the details of their practice because it is generally treated as secret information. Nevertheless, ten IT companies have responded to the survey and have agreed to answer our questions. The survey was not in formal written form. It was rather an informal conversation with general managers or chiefs of software teams leading to the answers and information we look for. The table 1 presents the general characteristics of the examined IT companies and the main methodology used.

Table 1. Examined IT companies

	Description of IT company	Main methodology used in the company
Company A	The company employs only three people. It produces small programs, simple to use, characterized by a low price.	RAD
Company B	The IT company's offer is addressed to small enterprises, to small businesses, improving their performance in the areas of management. The company's simple programs facilitate faster customer service. The number of users of these simple systems is over 60 thousand. The company is acting over 2 years.	RAD
Company C	The company addresses its software for small and medium enterprises. The number of enterprises using its software is about the 400th.	RAD
Company D	The IT company produces software dedicated mainly for business. It is based on technology from Apple. It has 10 years of experience in Polish and foreign projects.	Own methodology
Company E	The company is realizing technologically advanced IT projects in Poland and abroad. It works on the market for over 13 years, its advantages are experience and the international nature of the projects. Engaged in manufacturing systems, mainly web-based software, but also in the field of telephony and television.	Scrum
Company F	The company is specializing in creating modern software solutions for businesses. The company is present in business for 3 years. It is guided by the simplicity and efficiency using the latest Internet technologies.	Scrum

Company G	IT company's customers are small businesses and large corporations in areas of the USA and Europe. It employs over a hundred people. Its work takes place in a global environment. Most customers is just thousands of miles.	Scrum + XP
Company H	Its offer is directed to small and medium sized companies, mainly solutions for trade, services and manufacturing companies. It proposes solutions to support business management and financial accounting service. The company is present on the market over 20 years.	Document driven methodology
Company I	The products of this IT company are based on the global quality systems and are compatible with the latest ISO standards. The company carries out projects in major European countries in the financial sector and telecommunications.	Rational Unified Process
Company J	The software of this IT company is dedicated to the insurance and banking.	Rational Unified Process

During the interrogation in the form of informal conversation or interviews 25 queries have been addressed. The queries were as follows:

Q01. What kind of system design methodologies and software development methodologies, or some elements of them, have been applied in your company, which are no longer currently used?

Q02. What tools, methods, techniques, diagrams, and practices were used once and are not used today?

Q03. Which methodologies or parts of the methodologies you are using now and why?

Q04. What do you think about the new, innovative Agile Software Development methodology?

Q05. What methodology does the company has worked out?

Q06. Do you use only one methodology, all the time the same, for all projects? If the methodologies are different have they some common elements?

Q07. How has this methodology changed in practice in comparison with the native version?

Q08. What are the main techniques and practices used in the developed and accepted methodology?

Q09. What models are used to describe reality?

Q10. How are data and processes defined in the technical design of a system?

Q11. What are the main documents produced during the system design and software development? Is this a heavy or light methodology?

Q12. What is the composition of a project team? Have the team members formal or informal roles?

Q13. How is the cooperation? Are the team members all in one office room?

Q14. How are user requirements recognized?

Q15. Is the development processes structured?
Q16. At which stage of system design and software development do most errors appear, what are these errors?
Q17. What CASE tools are used and for what, what programs are used, what technologies are used?
Q18. How is feedback information received and how is the collaboration with the client realized?
Q19. Are you using agile techniques?
Q20. Are you using the methodology based on prototyping?
Q21. Is the software used more than once? Are there reusable resources?
Q22. Have you created your own patterns of software (analysis, design, architecture, applications)?
Q23. How many percent of the projects have failed, how many have been delayed, how much over budget?
Q24. What are the criteria for evaluating your systems?
Q25. What is the methodology of system testing?

4. SURVEY RESULTS

Figure 1 presents the popularity of system design methodologies in the interrogated IT companies.

The Scrum methodology seems to be one of the principle methodology used in practice in IT companies. The RAD methodology is also important in software companies.

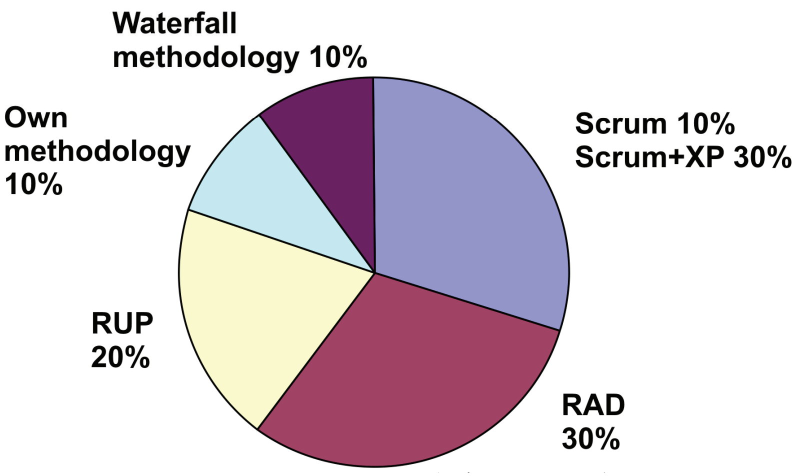


Fig. 1. The most frequently used methodologies

Table 2 and Figure 2 present the most preferred techniques in the examined software companies.

Table 2. The most eight preferred techniques in examined software companies

Company	1	2	3	4	5	6	7	8
A	class diagram	DFD	use case	ERD				
B	use case	class diagram	DFD	flowchart	use case diagram	state machine diagram	activity diagrams	
C	use case diagram	class diagram	ERD	DFD	activity diagrams	flowchart		
D	class diagram	user stories	use case	use case scenarios	Gantt diagram	activity diagrams	communication diagram	state machine diagram
E	use case scenarios	use case	user stories	use case diagram	activity diagrams	communication diagram	sequence diagram	package diagram
F	use case	use case scenarios	user stories	communication diagram	activity diagrams	state machine diagram	other UML diagrams	
G	use case	use case scenarios	DFD	flowchart	use case diagram	state machine diagram	activity diagrams	sequence diagram
H	ERD	DFD	data dictionary	flowchart	Gantt diagram	other UML diagrams		
I	class diagram	use case scenarios	use case	user stories	sequence diagram	activity diagrams	communication diagram	CRC card
J	use case scenarios	use case	user stories	Gantt diagram	state machine diagram	communication diagram	sequence diagram	package diagram

We can define a set of traits that characterize today's software development methodology:

- adapting to changing and unclear requirements,
- flexibility to adapt the functionality to user needs,
- rapid detection of problems,
- frequent inspection and adaptation,
- priorities of functionality,
- alternating roles at time intervals,
- testing an integral part, during the entire life cycle,

– all changes regarded as reversible,
– short development cycles,
– early implementation of certain requirements,
– selection of priority tasks implemented,
– reducing the documentation,
– reliance on oral communication,
– achieving business goals,
– active participation of the client,
– feedback,
– fast and frequent delivery of product,
– incremental process planning,
– courage to improve the system,
– self-confidence of the team.

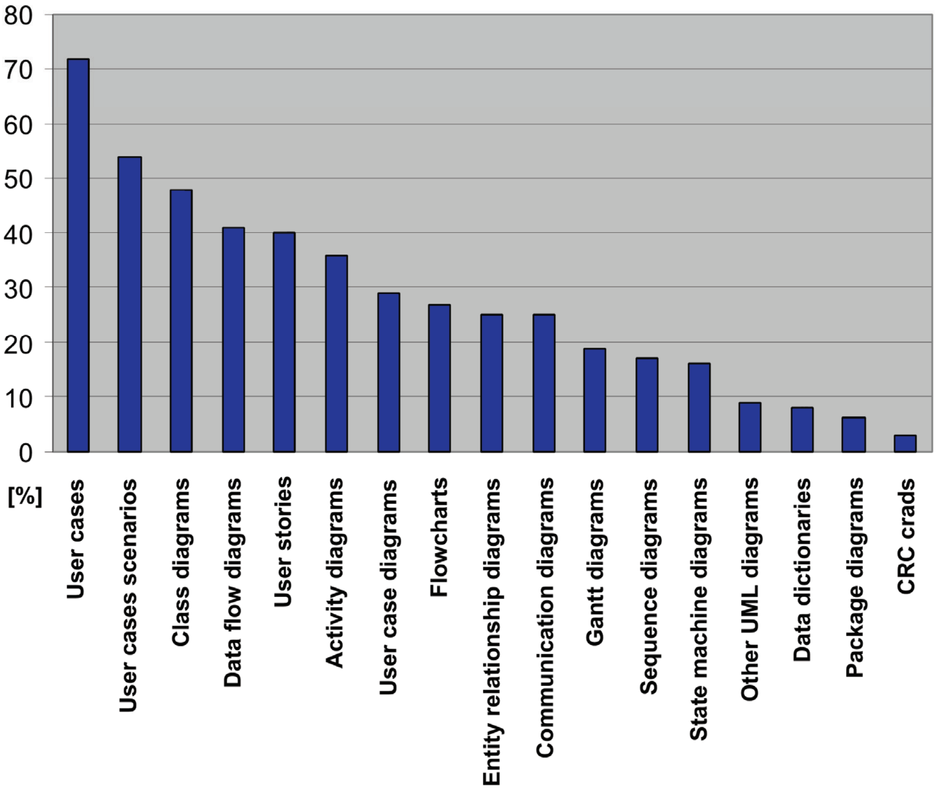


Fig. 2. The most frequently used techniques

The traditional system development methodologies similar to that used in architecture or machine constructions are widely substituted by new methodologies as well as new approaches leading to the acceleration of systems creation. This trend is also noticed in other investigations [15, 17].

5. CONCLUSIONS AND FINAL REMARKS

Several findings and conclusions of the survey of ten small and medium software companies can be formulated:
- there is no ideal method of designing systems and producing software,
- methodologies and designing methods in companies are never used according to the textbook models,
- each IT company has a set of preferred techniques, abandoning or neglecting others,
- most commonly used designing technique is use case with user history, modeled by use cases scenarios,
- paradigm of object-oriented modeling based on UML has been propagated and considered as standard,
- basic problems of modern IT teams is to organize work and communication internally within the team and externally with clients,
- iterative processes are typical for present-day system designing and software production,
- negligent attitude towards system documentation,
- redundant documentation, presenting changing requirements, is replaced by continuous contact with the customer.

REFERENCES

[1] ABRAN A., MOORE J., BOURQUE P., DUPUIS, R. (Eds.), *Guide to the Software Engineering Body of Knowledge – 2004 Version – SWEBOK Guide, A project of the IEEE Computer Society Professional Practices Committee*, IEEE-Computer Society Press 2005, 1–2.

[2] AVISON D., FITZGERALD G., *Information Systems Development: Methodologies, Techniques and Tools*, 4 edition, New York, McGraw-Hill Higher Education, 2006.

[3] COCKBURN A., *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison–Wesley, 2004.

[4] COOPER K., ABRAHAM S.P., UNNITHAN R.S., CHUNG L., COURTNEY S., *Integrating visual goal models into the Rational Unified Process*, Journal of Visual Languages and Computing, Vol. 17, No. 6, 2006, 551–583.

[5] DYBÅ T., DINGSØYR T., *Empirical studies of agile software development: A systematic review*, Information and Software Technology, Vol. 50, No. 9–10, 2008, 833–859.

[6] ERICKSON J., LYYTINEN K., SIAU K., *Agile Modeling, Agile software development, and extreme programming: the state of research*, Journal of Database Management, Vol. 16, No. 4, 2005, 88–100.

[7] FOJTIK R., *Agile methodology and development of software for users with specific disorders*, Proceedings of the 2010 International Multiconference on Computer Science and Information Technology, 2010, 687–691.
[8] GEER D., *Are Companies Actually Using Secure Development Life Cycles?*, Computer, Vol. 43, No. 6, 2010, 12–16.
[9] GREEN D., DiCATERINO A., *A Survey of Major Software Design Methodologies*, Report of the Center for Technology in Government University at Albany, 1998. (www.ctg.albany.edu/publications/reports/survey_of_sysdev)
[10] HAZZAN O., DUBINSKY Y., *Agile Software Engineering*, Springer, London 2008.
[11] *IEEE Standard Glossary of Software Engineering Terminology*, IEEE std 610.12-1990, 1990.
[12] KRUCHTEN P., *The Rational Unified Process: An introduction*, Addison-Wesley, Second Edition, 2003.
[13] LI S., SHANG J., SLAUGHTER S.A., *Why do software firms fail? Capabilities, Competitive Actions, and Firm Survival in the Software Industry from 1995 to 2007*, Information Systems Research, Vol. 21, No. 3, 631–654.
[14] MOE N.B., DINGSØYR T., DYBÅ T., *A teamwork model for understanding an agile team: A case study of a Scrum project*, Information and Software Technology Vol. 52, 2010, 480–491.
[15] NERUR S., MAHAPATRA R., MANGALARAJ G., *Challenges of migrating to agile methodologies*, Communications of the ACM, Vol. 48, No. 5, 2005, 72–78.
[16] PETERSEN K., WOHLIN C., *A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case*, The Journal of Systems and Software, Vol. 82, 2009, 1479–1490.
[17] PETERSEN K., WOHLIN C., *The effect of moving from a plan-driven to an incremental software development approach with agile practices. An industrial case study*, Empirical Software Engineering, Vol. 15, 2010, 654–693.
[18] PINOA F.J., PEDREIRAB O., GARCÍAC F., LUACESB M.R., PIATTINI M., *Using Scrum to guide the execution of software process improvement in small organizations*, The Journal of Systems and Software, Vol. 83, 2010, 1662–1677.
[19] SCHWABER K., BEEDLE M., *Agile Software Development with Scrum, Series in Agile Software Development*, Prentice Hall, 2002.
[20] SHORE J., WARDEN S., *The Art of Agile Development*, Sebastopol, O'Reilly Media, Inc., 2008.
[21] SULAYMAN M., MENDES E., *Quantitative assessments of key success factors in software process improvement for small and medium web companies*, Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, Sierre, Switzerland, 2319–2323.
[22] YU E., GIORGINI P., MAIDEN N., MYLOPOULOS J., *Social Modeling for Requirements Engineering*, MIT Press, 2011.

CBIR, spatial relationship, multi-set,
GUI, feature vector, search engine

Tatiana JAWORSKA*

THE CONCEPT OF A MULTI-STEP SEARCH-ENGINE FOR THE CONTENT-BASED IMAGE RETRIEVAL SYSTEMS

For the Content-Based Image Retrieval System (CBIR) we propose how to put together vectors of features for segmented image objects and a spatial relationship of the objects by constructing a multi-step search-engine, taking into account multi-set data mining and the object spatial relationship. The work presents a combination of two aspects of image representation, namely: features of segmented objects at lower level and spatial relations of objects to compare image similarities at a higher level. The new representation of spatial relationships of the image objects is based upon the Principal Component Analysis (PCA). It makes the method invariant to image rotation.

Additionally, we have constructed a graphical user interface (GUI) to enable the user to build a query by image. The efficiency of our system is being evaluated. In this work we present in detail all the steps of the search engine for our CBIR.

1. INTRODUCTION

Images and graphical data are complex in terms of visual and semantic contents. Depending on the application, images are modelled and indexed using their

- visual properties (or a set of relevant visual features),
- semantic properties,
- spatial or temporal relationships of graphical objects.

Over the last decade a number of concepts of the Content-Based Image Retrieval (CBIR) [1], [2], [3], [4], have been used. In Wikipedia we can also find a list of CBIR engines, used either for commercial or academic research purposes [5].

* Systems Research Institute, Polish Academy of Sciences, 01-447, 6 Newelska Street, Warsaw, Poland, e-mail: Tatiana.Jaworska@ibspan.waw.pl

Proposals can be found for the relational [6], object-oriented [7], [8] and object-relational database models [9]. Nevertheless, programmers have limited tools when they need to develop graphical applications dealing with imperfect pictorial data. Within the scope of semantic properties, as well as graphical object properties the first successful attempt was made by Candan and Li [10] who constructed the Semantic and Cognition-based Image Retrieval (SEMCOG) query processor to search for images by predicting their semantic and spatial imperfection. This new approach has been very important because earlier, and even present-day, queries to the database are put as query-by-example images.

Hence, in order to give the user the opportunity to compose their own image, consisting of separate graphical objects as a query, we have had to create our own system. An image created in GUI has its own unique object location in the image space. Thus, many researchers Chang [11, 12], Chang and Wu [13, 14], Zhou at al. [15] highlighted the importance of perceiving spatial relationships existing among the components of an image for efficient representation and retrieval of images in the CBIR.

We have dealt successfully with numerous problems involved in the CBIR system, with one final issue that still requires our attention. Ultimately, we have managed to form a new paradigm in comparing images with the search engine.

1.1. CBIR CONCEPT OVERVIEW

In general, our system consists of four main blocks (fig. 1):

1. the image preprocessing block (responsible for image segmentation), applied in Matlab, cf. [16];
2. the Oracle Database, storing information about whole images, their segments (here referred to as graphical objects), segment attributes, object location, pattern types and object identification, cf. [17];
3. the search engine responsible for the searching procedure and retrieval process based on the feature vector for objects and spatial relationship of these objects in the image, applied in Matlab;
4. the graphical user's interface (GUI), also applied in Matlab.

A query by image allows users to search through databases to specify the desired images. It is especially useful for databases consisting of very large numbers of images. Sketches, layouts or structural descriptions, texture, colour, sample images, and other iconic and graphical information can be applied in this search.

An example query might be: *Find all images with a pattern similar to this one*, where the user has selected a sample query image. In the QBIC system [3] the images are retrieved based on the above-mentioned attributes separately or using distance functions between features. Tools in this GUI include some basic objects, such as: polygon outliner, rectangle outliner, line draw, object translation, flood fill, eraser, etc.

More advanced systems enable users to choose as a query not only whole images but also individual objects. The user can also draw some patterns, consisting of simple shapes, colours or textures [18]. In the SEMCOG query processor [10], the user could organize an image as a spatial composition of five semantic groups of objects, such as: car, woman, man, house and bicycle. Additionally, the user could choose the colour, size and shape of a graphical object. In order to retrieve a matched image, the system integrated an image query statement with non-image operation statement.

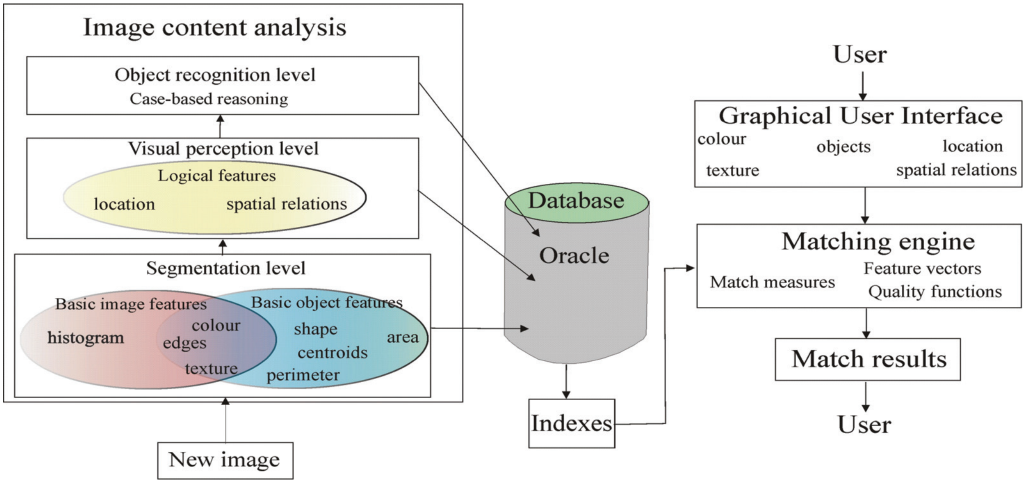


Fig. 1. Block diagram of our content-based image retrieval system

There have been several attempts made by the research community to disperse the demands in the design of efficient, invariant, flexible and intelligent image archival and retrieval systems based on the perception of spatial relationships. Chang [19] proposed a symbolic indexing approach, called the nine directional lower triangular (9DLT) matrix to encode symbolic images. Using the concept of 9DLT matrix, Chang and Wu [20] proposed an exact match of the retrieval scheme, based upon principal component analysis (PCA). Unfortunately, it turned out that the first principal component vectors (PCVs) associated with the image and the same image rotated are not the same. Eventually, an invariant scheme for retrieval of symbolic images based upon the PCA was prepared by Guru and Punitha [21].

2. GRAPHICAL DATA REPRESENTATION

In our system, Internet images are downloaded. Firstly, the new image is segmented, creating a collection of objects. Each object, selected according to the algorithm presented in detail in [16], is described by some low-level features. The features describing each object include: average colour k_{av}, texture parameters T_p, area A, convex area A_c, filled area A_f, centroid $\{x_c, y_c\}$, eccentricity e, orientation α, moments of inertia m_{11}, bounding box $\{bb_1(x, y), ..., bb_s (x,y)\}$ (s – number of vertices), major axis length m_{long}, minor axis length m_{short}, solidity s and Euler number E and Zernike moments Z_{00}, …, Z_{33}. All features, as well as extracted images of graphical objects, are stored in the DB. Let F be a set of features where:

$$F = \{k_{av}, T_p, A, A_c, \dots, E\}. \tag{1}$$

For ease of notation we will use $F = \{f_1, f_2, \dots, f_r\}$, where r – number of attributes. For an object, we construct a feature vector O containing the above-mentioned features:

$$F_O = \begin{bmatrix} O(k_{av}) \\ O(T_p) \\ O(A) \\ \vdots \\ O(Z_{33}) \end{bmatrix} = \begin{bmatrix} O(f_1) \\ O(f_2) \\ O(f_3) \\ \vdots \\ O(f_r) \end{bmatrix}. \tag{2}$$

The average colour is an average of each red, green and blue component which is summed up for all the pixels belonging to an object, and divided by the number of object pixels $k_{av} = \{r_{av}, g_{av}, b_{av}\}$. The next complex feature attributed to objects is texture. Texture parameters are found in the wavelet domain (the Haar wavelets are used). The algorithm details are also given in [16]. The use of this algorithm results in obtaining two ranges for the horizontal object dimension h and two others for the vertical one v:

$$T_p = \begin{Bmatrix} \{h_{\min_{1,2}};\ h_{\max_{1,2}}\} \\ \{v_{\min_{1,2}};\ v_{\max_{1,2}}\} \end{Bmatrix}. \tag{3}$$

Additional features of the low level for objects are shape descriptors. They are also included in the above mentioned feature vector. We apply the two most important shape descriptors such as moments of inertia:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x, y), \qquad p, q = 0, 1, 2 \tag{4}$$

and Zernike moments [22]. Zernike moments are a set of complex polynomials $\{V_{pq}(x, y)\}$ which form a complete orthogonal set over the unit disk of $x^2 + y^2 \le 1$. Hence, the definition of 2D Zernike moments with p-th order with repetition q for intensity function $f(x, y)$ of the image is described as:

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x^2+y^2 \le 1} V^*_{pq}(x, y) f(x, y) dxdy\,, \tag{5}$$

where:

$$V^*_{pq}(x, y) = V_{p,-q}(x, y)\,. \tag{6}$$

For our purpose, the first 10 Zernike moments are enough, it means we calculate moments from Z_{00} to Z_{33}. The scale invariance is obtained by normalizing Z_{00} by the total number of image pixels.

Characteristic features of Zernike moments are:

1. The above-defined Zernike moments are only invariant to rotation.
2. The translation invariance is achieved by the location of the original image centroid in the centre of the coordinates.

3. SPATIAL RELATIONSHIP OF GRAPHICAL OBJECTS

The feature vector F_o (cf. (2)) is further used for object classification. Therefore, we have to classify objects first in order to assign them to a particular class and second in order to compare objects coming from the same class [23].

In our system spatial object location in an image is used as the global feature. Firstly, it is easy for the user to recognize this spatial location visually. Secondly, it supports full identification based on rules for location of graphical elements. Let us assume that we analyse a house image. Then, for instance, an object which is categorized as a window cannot be located over an object which is categorized as a chimney. For this example, rules of location mean that all architectural objects must be inside the bounding box of a house. For an image of a Caribbean beach, an object which is categorized as a palm cannot grow in the middle of the sea, and so on. For this purpose, the mutual position of all objects is checked. The location rules are also stored in the pattern library [23]. Thirdly, object location reduces the differences between high-level semantic concepts perceived by humans and low-level features interpreted by computers.

For the comparison of the spatial features of two images an image I_i is interpreted as a set of n objects composing it:

$$I_i = \{o_{i1}, o_{i2}, \ldots, o_{in}\}\,. \tag{7}$$

Each object o_{ij} is characterized by a unique identifier and a set of features discussed earlier. This set of features includes a centroid $C_{ij} = (x_{ij}, y_{ij})$ and a label L_{ij} indicating the class of an object o_{ij} (such as window, door, etc.), identified in the process described in [23]. For convenience, we number the classes of the objects and thus L_k's are just numbers.

Formally, let I be an image consisting of n objects and k be a number of different classes of these objects, $k \leq N$, because usually there are some objects of the same type in the image, for example, there can be four windows in a house.

Let us assume that there are, in total, M classes of the objects recognized in the database, denoted as labels $L_1, L_2, \ldots, L_M$. Then, by the signature of an image I_i (7) we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \ldots, \text{nobc}_{iM}], \tag{8}$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i, i.e. such objects o_{ij}.

Additionally, for an image I_i we consider a representation of spatial relationships of the image objects. The object's o_{ij} mutual spatial relationship is calculated based on the algorithm below. Now, we consider one image; let C_p and C_q be two object centroids with $L_p < L_q$, located at the maximum distance from each other in the image, i.e.,

$$\text{dist}(C_p, C_q) = \max \{\text{dist}(C_i, C_j)\ \forall i, j \in \{1, 2, \ldots, k\} \text{ and } L_i \neq L_j\}, \tag{9}$$

where: dist(•) is the Euclidean distance between two centroids (see fig. 2). The line joining the most distant centroids is the line of reference and its direction from centroid C_p to C_q is the direction of reference for computed angles θ_{ij} between other centroids. This way of computing angles makes the method invariant to image rotation.

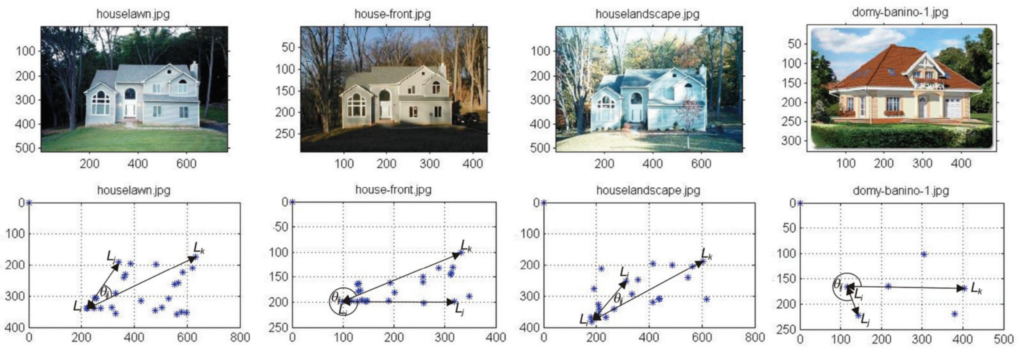


Fig. 2. Determination of angle relative to the reference direction for the construction of matrix S

Hence, we received triples (L_i , L_j, θ_{ij}) where the mutual location of two objects in the image is described in relation to the line of reference (see fig. 2 bottom). Thus, there are $T = m(m - 1)/2$ numbers of triples, generated to logically represent the image consisting of m objects. Let S be a set of all triples, then we apply the concept of principal component analysis (PCA) proposed by Chang and Wu [20] and later modified by Guru and Punitha [21] to determine the first principal component vectors (PCVs).

First, we have to suppose that S is a set of observations for three variables. We construct a matrix of observations $X_{3\times N}$ where each triple is one observation. Next, we count the mean value u of each variable, and we calculate the deviations from the mean to generate matrix $\mathbf{B} = \mathbf{X} - \mathbf{u1}$, where $\mathbf{1}$ – vector of all 1s. In the next step, we compute the covariance matrix $\mathbf{C}_{3\times 3}$ from the outer product of matrix $\mathbf{B}$ by itself as:

$$\mathbf{C} = \mathbb{E}\,[\mathbf{B}\otimes\mathbf{B}] = \mathbb{E}\,[\mathbf{B}\,\mathbf{B}^*] = 1/\mathrm{N}\,[\mathbf{B}\,\mathbf{B}^*] \tag{10}$$

where: $\mathbb{E}$ is the expected value operator, $\otimes$ is the outer product operator, and * is the conjugate transpose operator. Eventually, we find eigenvectors, which diagonalises the covariance matrix $\mathbf{C}$:

$$\mathrm{V}^{-1}\,\mathrm{C}\,\mathrm{V} = \mathrm{D}, \tag{11}$$

where: D is the diagonal matrix of the eigenvalues of C.

Using the Matlab procedure `V = princomp(X)`, we receive three component vectors (PCVs). For further analysis we use the first of them, which is the “spatial component” of the representation of an image I_i, and is denoted PCV_i.

For example, we use centroid coordinates from our CBIR to find angle θ_{ij} (see fig. 2 bottom). Thus, we construct set S of our observations, where N is combinations of the centroid numbers. For instance, $N_{I_1} = C_2^{26} = 325$ and $N_{I_2} = C_2^{21} = 210$, respectively. The obtained results are shown in table 1.

Table 1 Representative principal component vectors for the images shown in fig. 2

Image name	First component	Second component	Third component
House-front	–0,001786	–0,003713	0,999992
Domy-banino-1	0,000206	0,003988	0,999992
Houselawn I_1	0,000388	0,001869	0,999998
Houselandscape I_2	0,004109	0,001557	0,999990

4. CONSTRUCTION OF SEARCH ENGINE

Graphical User Interface (GUI) is a crucial element of our system as the area of human-computer interaction [24]. Hence, the user chooses particular graphical elements from subsequent menus and places them on the appropriate location in the chosen outline. These elements can be scaled in a limited range. In most query-by-example systems, the features for retrieval and their importance are estimated by the system. Even in systems where such information can be provided by the user, users cannot always communicate unambiguously what they are looking for. In our system, these constraints are overcome by the user's selection of specific features (for example, the colour and texture of an object) from numerous menus. After the designing process, the image is sent as a query to the DB; it means that we have feature vectors F_{qi} (where $i = 1, ..., N$) for all objects used to form either query image I_q and PCV_q.

So far, we have described how images are represented in our system. Now, we will describe how the similarity between two images is determined and used to answer a query. Let a query be an image I_q, such as $I_q = \{o_{q1}, o_{q2}, \ldots, o_{qn}\}$ (cf. (7)). An image in the database will be denoted as I_b, $I_b = \{o_{b1}, o_{b2}, \ldots, o_{bm}\}$. In order to answer the query, represented by I_q, we compare it with each image I_b in the database in the following way.

First of all, we determine a first similarity measure $\mathrm{sim}_{\mathrm{sgn}}$ between I_q and I_b computing the Hamming distance $d_H(x,y) \in \mathrm{F}_{10}^{(M)}$ between the vectors of their signatures (8), i.e.:

$$\mathrm{sim}_{\mathrm{sgn}}(I_q, I_b) = d_H(\mathrm{nobc}_q, \mathrm{nobc}_b)\,. \tag{12}$$

If the similarity (12) is smaller than a threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q. Otherwise, we proceed to the next step and we find the spatial similarity $\mathrm{sim}_{\mathrm{PCV}}$ of images I_q and I_b computing the Euclidean distance between their PCVs as:

$$\mathrm{sim}_{\mathrm{PCV}}(I_q, I_b) = 1 - \sqrt{\sum_{i=1}^{3} (PCV_{bi} - PCV_{qi})^2} \tag{13}$$

If the similarity (13) is smaller than the threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q. Otherwise, we proceed to the final step, namely, we compare the similarity of the objects representing both images I_q and I_b. For each object o_{qi} present in the representation of the query I_q, we find the most similar object o_{bj} of the same class, i.e., $L_{qi} = L_{bj}$. If there is no object o_{bj} of the class L_{qi}, then $\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_b)$ is equal to 0. Otherwise, similarity $\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_b)$ between objects of the same class is computed as follows:

$$\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_{bj}) = 1 - \sqrt{\sum_{l} (Fo_{qil} - Fo_{bjl})^2}\,, \tag{14}$$

where l indexes the set of features F_O used to represent an object, as described in (2). When we find highly similar objects (for instance, $\mathrm{sim}_{\mathrm{ob}} > 0.9$), we eliminate these two objects from the following process of comparison [25]. The process is realized according to the algorithm presented below:

Algorithm: Pair matching algorithm with elimination

```
k=0;
i=1;
j=1;
for j=j:Lqi %number of objects in a particular class
 for i=i:Lbj %number of objects in a particular class
  if sim(i,j)>.9
       match(i,j)=sim(i,j);
       row(i)=i;
       col(j)=j;
        j=j+1;
        i=i+1; end;
    end; end;
while k==0
  [k,R]=min(row);
  [k,C]=min(col);
  match(R,C)=sim(R,C);
  row(R)=R;
  col(C)=C;
end;
```

Thus, we obtain the vector of similarities between the query I_q and an image I_b.

$$\mathrm{sim}(I_q, I_b) = \begin{bmatrix} \mathrm{sim}_{\mathrm{ob}}(o_{q1}, o_{b1}) \\ \vdots \\ \mathrm{sim}_{\mathrm{ob}}(o_{qn}, o_{bn}) \end{bmatrix}, \tag{15}$$

where n is the number of objects present in the representation of I_q.

In order to compare images I_b with the query I_q, we compute the sum of $\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_{bi})$ and then use the natural order of the numbers. Thus, the image I_b is listed as the first in the answer to the query I_q, for which the sum of similarities is the highest.

5. CONCLUSION

The construction of a CBIR system requires combining different functional systems, linked together and cooperating with each other. For this purpose, object classification and identification procedures have been established and the GUI prototype has been constructed.

We have prepared a model of image similarity as a three-step procedure. This is, of course, a preliminary model of a three-step procedure to answer a query. There are many other possible ways to compute the similarity between the images, e.g. using different metrices. Intensive computational experiments are under way in order to come up with some conclusions as to the choice of the parameters of the model, including the choice of the above-mentioned metrices. However, the preliminary results we have obtained so far using the simplest configuration are quite promising.

REFERENCES

[1] DEB S. (Ed.), *Multimedia Systems and Content-Based Image Retrieval*, Chap. VII and XI, IDEA Group Publishing, Melbourne 2004.
[2] FLICKNER M. and SAWHNEY H., et al., *Query by Image and Video Content: The QBIC System.* IEEE Computer 28 (9), 1995, 23-32.
[3] NIBLACK W., FLICKNER M. et al., *The QBIC Project: Querying Images by Content Using Colour, Texture and Shape*, SPIE 1908, 1993, 173–187.
[4] OGLE V., STONEBRAKER M., *CHABOT: Retrieval from a Relational Database of Images*, IEEE Computer 28 (9), 1995, 40–48.
[5] http://en.wikipedia.org/wiki/List_of_CBIR_engines
[6] PONS O., VILA M.A., KACPRZYK J., *Knowledge management in fuzzy databases*, Studies in Fuzziness and Soft Computing, Vol. 39, Physica-Verlag, Heidelberg and New York 2000.
[7] LEE J., KUO J.-Y., XUE N.-L., *A note on current approaches to extent fuzzy logic to object oriented modeling*, International Journal of Intelligent Systems, Vol. 16, 2001, 807–820.
[8] BERZAL F., CUBERO J.C., KACPRZYK J., MARIN N., VILA M.A., ZADROŻNY S., *A General Framework for Computing with Words in Object-Oriented Programming*, [in:] International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems, Vol. 15, (Supplement), Feb., World Scientific Publishing Company, Singapore 2007, 111–131.
[9] CUBERO J.C., MARIN N., MEDINA J.M., PONS O., VILA M.A., *Fuzzy Object Management in an Object-Relational Framework*, [in:] Proceedings of the 10th International Conference IPMU, Perugia, Italy, 2004, 1775–1782.
[10] CANDAN K.S., LI W.-S., *On Similarity Measures for Multimedia Database Applications*. Knowledge and Information Systems, 3, 2001, pp. 30–51.
[11] CHANG S.K., SHI Q.Y., YAN C.W., *Iconic indexing by 2D strings*, IEEE Trans. Pattern Anal. Machine Intell. 9 (5), 1987, 413–418.
[12] CHANG S.K., JUNGERT E., LI Y., *Representation and retrieval of symbolic pictures using generalized 2D string*, [in:] SPIE Proc. on Visual Comm. and Image Process., Philadelphia 1989, 1360–1372.
[13] CHANG C.C., WU T.C., *Retrieving the most similar symbolic pictures from pictorial databases*, Informat. Process. Manage., 28 (5), 1992, 581–588.
[14] WU T.C., CHANG C.C., *Application of geometric hashing to iconic database retrieval*, Pattern Recognition Letters 15, 1994, pp. 871–876.
[15] ZHOU X.M., ANG C.H., LING T.W., *Image retrieval based on object's orientation spatial relationship*, Pattern Recognition Letters, 22, 2001, 469–477.
[16] JAWORSKA T., *Object extraction as a basic process for content-based image retrieval (CBIR) system*, Opto-Electronics Review, Association of Polish Electrical Engineers (SEP), Vol. 15 (4), Warsaw 2007, 184-195.
[17] JAWORSKA T., *Database as a Crucial Element for CBIR Systems*, [in:] Proceedings of the 2nd International Symposium on Test Automation and Instrumentation, Vol. 4, World Publishing Corporation, Beijing, China, 2008, 1983–1986.

[18] SMITH J.R., CHANG, Sh.-F., *Integrated spatial and feature image query*, Multimedia Systems, Vol. 7, Springer-Verlag, 1999, 129–140.
[19] CHANG C.C., *Spatial match retrieval of symbolic pictures*, Journal of Informat. Science Engineering, 7, 1991, 405–422.
[20] CHANG C.C., WU T.C., *An exact match retrieval scheme based upon principal component analysis*, Pattern Recognition Letters, 16, 1995, 465–470.
[21] GURU D.S., PUNITHA P., *An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis*, Pattern Recognition Letters, Vol. 25, 2004, 73–86.
[22] TEAGUE M.R., *Image analysis via the general theory of moments*, JOSA, Vol. 70, 8 ed., 1980, 920–930.
[23] JAWORSKA T., *Graphical Object Classification and Query by Image as an Aspect of Content-Based Image Retrieval System*, [in:] Studies and Materials of Polish Operational and Systems Research Society, Vol. 32, ed. J. Owsiński, Bydgoszcz, Poland, 2010, 269–282.
[24] NEWMAN W.M., LAMMING M.G., *Interactive System Design*, Addison-Wesley, Harlow 1996.
[25] MUCHA M., SANKOWSKI P., *Maximum Matchings via Gaussian Elimination*, [in:] Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS ‘04), 2004, 248–255.

decision support, agent system,
air quality management

Cezary ORŁOWSKI*, Łukasz SZCZYGIELSKI**,
Paweł MADEJ***

INTERACTIVE INFORMATION AND DECISION SUPPORT SYSTEM FOR URBAN AND INDUSTRIAL AIR QUALITY MANAGEMENT BASED ON MULTI-AGENT SYSTEM

This work presents conception of interactive information and decision support system for urban and industrial air quality management. The emphasis of the project is on real-time analysis and multi-media information, and the support of distributed and mobile clients through the Internet. The approach integrates meteorological data and forecasts, air quality and emission monitoring, dynamic 3D simulation modelling and forecasting, GIS, expert systems, decision support and reporting tools in a unified, modular client/server framework implemented as a range of web accessible application services.

1.1. INTRODUCTION

The aim of this work is to present a model of a decision support system and its application to air quality management. The project is within the framework of the global undertaking "Eureka WEBAIR".

The first part presents the general characteristics and assumptions of "Eureka WEBAIR". Then the merits of the project for the Tri-City area are discussed and the main objectives of the decision support system created for the project are characterized. The next section discusses the construction of the model, Multi-Agent system

---

* University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk, e-mail: Cezary.Orlowski@zie.pg.gda.pl

** University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk, e-mail: Lukasz.Szczygielski@zie.pg.gda.pl

*** University of Technology in Gdańsk, ul. Narutowicz 11/12, 80-233 Gdańsk, e-mail: Pawel.Madej@zie.pg.gda.pl

role and the stages of system implementation. The presented work is concluded with a summary and an assessment of the system's practicality.

2. BACKGROUND OF THE PROJECT – EUREKA WEBAIR

2.1. EUREKA PROJECT

The emphasis of the project is on real-time analysis and multi-media information, and the support of distributed and mobile clients through the Internet. The approach integrates meteorological data and forecasts, air quality and emission monitoring including mobile sources such as traffic, dynamic 3D simulation modelling and forecasting, GIS, expert systems, optimization, decision support and reporting tools in a unified, modular client/server framework implemented as a range of web accessible application services.

WEBAIR will develop, test, and implement a web based information and decision support system for urban and industrial air quality assessment and management in support of relevant EU Directives such as the Air Quality framework Directive for urban conglomerates (96/62/EEC) and major industrial emission sources such as thermal power plants (88/609/EEC) or incinerators (89/429/EEC, 89/369/EEC), and 90/313/EEC on freedom of access to environmental information.

The basic idea is to offer an integrated set of tools to support regulatory compliance and reporting requirements for and on behalf of cities and industries subject to the above environmental directives. The basic business concept is complete or partial outsourcing of a range of web-accessible application services for distributed and mobile clients. Regulatory reporting as well as public information is the information product derived as an added value service from basic meteorological and environmental data. Value added in public-private partnership, and open access to environmental information as foreseen under (90/313/EEC), packaged as attractive, informative, and educational multi-media content for a range of user communities, are key concepts.

Technically, the emphasis is on real-time analysis, combining on-line monitoring and model-based assessment into multi-media information and report for a broad range of consumers, including the general public interested in environmental quality. The support of distributed and mobile clients through the Internet adds an additional business perspective for network and mobile phone operators.[1]

The approach integrates a range of real-time and on-line data sources and tools [2]: meteorological data and forecasts, air quality and emission monitoring including mobile sources such as traffic from on-line observations and counts, dynamic 3D simulation modelling and forecasting, GIS, expert systems, optimization, decision support and reporting tools in a unified but modular client/server architecture implemented as a range of web accessible application services. In addition to real-time monitoring and assessment with on-line publication of the information in real time as well as forecasts, and regular compliance reporting to meet regulatory requirements, the tools will also support strategic analysis of emission control using complex optimization technologies.

The technological developments in WEBAIR will focus on a number of specific closely related areas:

- Acquisition and real-time processing of monitoring and observation data; the emphasis is on capture of potentially large volumes of diverse data, efficient storage, quality assurance (plausibility, completeness, consistency) and retrieval in support of real-time processing, integration of diverse data sources including meteorological data and forecasts, air quality monitoring, satellite imagery, emission monitoring, and traffic observations.
- Integration of real-time modelling tools to augment the monitoring data from a few locations into a complete yet detailed spatial coverage of air quality information; in addition to the real-time now casting and data assimilation, short and medium term forecasts based on meteorological forecasts and dynamic emission models will be run on a regular basis, synchronized with the observation frequencies. Specific topics will include complex terrain, coastal locations and sea breeze, urban heat islands, behavior of fine particles and the explicit treatment of urban structures in dynamic 3D models.
- Automatic translation of this information into attractive multi-media formats for web access including low-resolution mobile clients, as well as the automatic generation of summary reports over various periods according to the regulatory requirements.
- Testing these components under real operational conditions in a number of applications and developing the necessary exception handling and error correcting methods for an automatic but highly reliable assessment and high availability mission critical performance.

The project invites industries subject to the Major Source Directive or cities subject to the Air Quality Framework Directive as test users of the system. Technology partners sought include manufacturers of monitoring equipment and systems for meteorological data, ambient air quality, emissions, and traffic data. WEBAIR currently includes partners from Austria, Switzerland, Russia, Finland, Portugal, Cyprus, Morocco, Italy and Lithuania and Poland. [1]

2.2. THE ARGUMENT FOR NECESSITY OF THE EUREKA PROJECT IN TRI-CITY AREA

The Tri-City (Gdańsk, Sopot, Gdynia), together with all its urban area is the largest industrial and cultural center in Central Pomerania. Its specific geographical location, the presence of all the everyday means of transport (road, rail and air) and the fact that it is home for some large and environmentally burdensome industries (such as shipyards, power plants, a refinery), are all sources of air pollution, and pose a threat of a major industrial incident. The communication system, and in particular the main thoroughfare that passes the centers of all the three cities, with dense housing either side, is also unfavorable from the viewpoint of environmental risk. From the main thoroughfare, there are connections to such areas as the ports, shipyards and other industrial plants.

It seems appropriate to create a distributed IT system [3,4], to manage air quality which takes into account this specific communication system and the influence of large companies (Lotos, CHP plant, the Port of Gdansk) on the formation of the air pollution map in the Tri-City. The developed solution will be an easily adaptable system for any urban area dealing with environmental problems and industrial hazards and the consequences for residents which arise from these risks.

The system developed under the Eureka WEBAIR project will create conditions for the construction of multi-annual investment plans and development strategies. It will be possible through the use of fuzzy urban development scenarios (Polish partner's contribution) and their verification in an integrated IT urban-ecological environment. This approach will allow for the verification of the effectiveness of company environmental management systems, based on ISO 14000 and EMAS, but will also allow for the rapid identification of possible risks arising from road transport.

3. THE DECISION SUPPORT SYSTEM MODEL FOR AIR QUALITY MANAGEMENT

The aim of the Polish part of the project is to develop a decision support system by building a distributed system to manage air quality in urban areas. It is aimed at the needs of large urban decision-makers to assist their planning decisions. Its design is based on the construction and use of fuzzy decision-making scenarios and their subsequent processing.

The system developed within the Eureka project will help build and verify the long-term investment plans and development strategies for Gdansk up to the year 2025, while taking into account the specific communication system and the influence of large companies (Lotos, CHP plant and the Port of Gdańsk) on the formation of the pollution map in the urban area of the Tri-City. There is also a plan to use an alarm-warning system, in the event of threats resulting from transport, or industrial incidents.

The developed solution will be a system easily adaptable to any urban area dealing with environmental problems and industrial hazards and the consequences for the inhabitants resulting from the aforementioned threats. Therefore, an open solution will be implemented (in view of the other project partners) which will enable the system to cooperate with other systems to provide and publish data by these systems. It is also assumed that it will cooperate with the central national air pollution base JPOAT, and the system developed under the SUTRA project. In both cases, the system will automatically deliver online to the other systems the revised time series describing air pollution and the meteorological situation. A general decision support system model for air quality management is shown below:

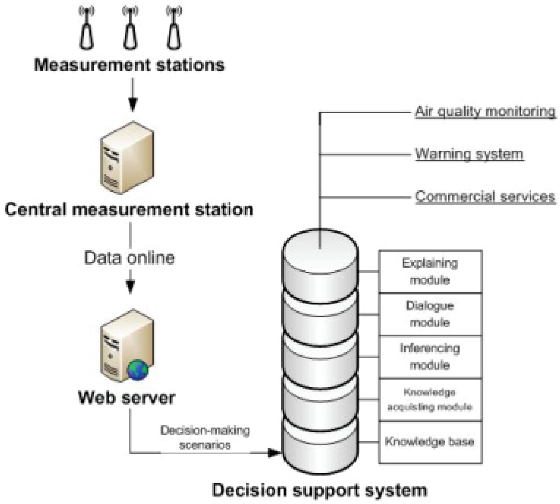


Fig. 1. Decision support system

Data from scattered measurement stations are transmitted to the central station. Then, the collected data are sent to the server in real time mode, which allows the measurements to be published online. These data are collected, processed and verified for the construction of decision-making scenarios. The revised scenarios for decision-making feed into the knowledge base of the expert system. Expert system will be based on Multi Agent System (MAS) [5,6].

During the years many definitions about agents and agent systems have been made. In 1995 Russel wrote about agents as “anything, what might be recognized as observing attendance throughout sensors and operating within this attendance effectors.” The second from the well known definitions (elaborated by IBM) refers to agents as program unit who take an operation on behalf of a user or other programs and which are somewhat self-dependent and autonomous and they use in its operation certain knowledge or goals representation and users needs. In Wooldridge’a article from 1997 it can be read about agent system as certain closed computer system which is occurred in attendance, and (that system) has abilities of adaptable operations in this attendance and operations consisted of accomplish goals it has been created for. The most precise definition of agent system which is being found in literature is Maes' definition, which was made in 1998, spoke about agent system as autonomous system being in dynamic attendance, acting regardless of imposed limitations and executing within its goal set or commands, it has been made for.

By multi-agent systems should be meant computer's systems based and built on programs base – agents. Making that systems – agent programming – consist in making agent programs set and environment for its operations [7]. Certain set of immanent features (attribute) is attributed which are characteristic for them. For those attributes belong for example reactivity, directing at achieving goals or autonomy. Autonomy means that relevant agent should make charging or imputing self-depended tasks. When we say about imputing tasks, there should be mentioned the next attribute of agent system that is acting a.j. (on behalf of principal) It is said about agent as delegate who behalf of user, other program or even other agent. Determinant of agents are also adaptation skills (accommodation with attendance), learning and communication. Mainly, it means about acting in concert with other agents (but communication with users or other programs such as date base are possible). Thanks to this communication, agent acting might be more effective [5, 7]. It is worth to mention that mobility requirements are put to agents that agent programs are able to relocate in different place (different machine, different environment) and there continue performed tasks – charging acts.

Necessity of agent cooperation in various system need also abilities their reverse cooperation and liaison. In accordance with previous things, agent systems should be useful in acquiring knowledge, its processing and inferring, therefore there is so big importance of agents in ontology management (that is "date base about knowledge"). The result is that apart from previous attributes, more important is that agent or whole system should distinguish of its skills, predictions, reasoning and inferring. Attendance of this attributes in agent system let to affirm that it is reasoned to use agents for rating various occurrences. It is easy to notice that agent system attributes are rather specific and don't mean wholly district attributes but in many areas they infiltrate and complementary to one another [5].

After acquaint with abilities of agent system functions and its attributes and in accordance with inferring, agent system concept must be presented, which we can depute data obtained from monitoring stations evaluation for.

The first primary function, which should be noticed is the terms of agent system functioning. Remembering about system's attributes such as knowledge acquiring, processing and findings' generation, we should provide power (supply) for agent system in shape of information and knowledge. In order to attain this is reasonable to use meta-knowledge bases such as ontology which are built and based on knowledge of experts who estimate obtained data and decision-making scenarios. In addition, to process inferring accomplish, in agent system should be also expert system which, from one hand, can transfer supporting knowledge, and on the other hand to harness the expert knowledge which is included in knowledge bases. Furthermore, in order to streamline knowledge transfer to system, it is worth to include in it applications which are based on formal methods of knowledge representation such as fuzzy modelling or neural network which enable transfer of imprecise and uncertain knowledge. Thanks this, the system can be obtained, where ontology, expert system as well as other applications which support knowledge acquirement are primary element of this system. Expert system's task is going to be knowledge delivering to relevant with data and scenarios evaluation.

This agent system concept allows to fulfil double function. The first as the inferring tool which in accordance with knowledge is possessed let to estimate obtained data and decision-making scenarios on account of defined criteria. On the other hand, that system's construction, where both ontology and expert system or other applications which can represent separate agents or report to agents who are responsible for each measurement station, are able to recognize this agent system as management system which manage on tools to assist air quality data estimate. In Figure 2 the structure of discussion system concept has been presented.

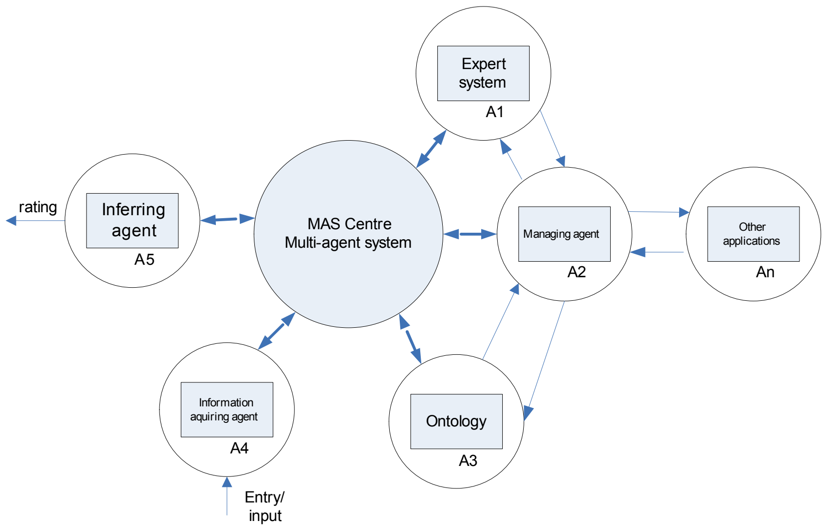


Fig. 2. Model of MAS based system

The stages of the implementation of the EUREKA system model are shown below:

Stage 1. Extending the activity of measurement stations with devices and functions for checking air quality

The provision of data for decision-making scenarios will be implemented by the Technical University of Gdansk in cooperation with the ARMAAG Foundation and the Marshal's Office. The ARMAAG Foundation has 10 stations and central station software (CAS) assisted by applications for the verification, validation and visualization of information. The standard network activity of automated stations is carried out on the basis of the QA / QC quality assurance plan for (Quality Control / Quality Assurance).

Due to the needs of the decision-making scenarios, it will be necessary to ensure the quality of the automated measurements, which will require the application of appropriate quality control procedures throughout the measuring process. The data from all the automated stations will be transmitted to the central station in real time. The results and statuses of the measurements will be archived in local stations and stored in a raw data database without the possibility of changes being made by the operator. In addition, to ensure the quality of data, a plan for calibration is to be developed, according to which operations will be made to be included in the full calibration of an instrument.

It is also planned to expand networks in certain zones which, after consultation with the ARMAAG Foundation, were considered insufficient in terms of providing measurements (e.g. along the main thoroughfare of the Tri-City, as well as in areas of increased industrial hazard). To achieve this aim, modern monitoring stations, designed at the Multimedia Systems Department, are to be used. Apart from providing noise measurement and image analysis, they will cooperate with a highly integrated autonomous weather station. Moreover, thanks to its modular design, the system can be extended with pollutant sensors: SO2, NO2, CO, C6H6, O3, WWA, PM10, PM2.5 as required. Such a plan will provide sufficient accuracy and quality of calibration, and thus will ensure the high quality (accuracy, completeness and consistency) of data collected with the use of analyzers prepared in such a way.

The concept of using online data in the construction of decision-making scenarios assumes that the data will be provided by a coherent measurement program and a model showing how the emitted substances spread, which will be well-suited to the local (regional) conditions. In the construction of decision-making scenarios, the policy makers – planners will need to take into account the specific meteorological and functional conditions of the Gdansk urban area (Sopot has the status of a spa), for which the objectified measurement system will become an indispensable and irreplaceable tool. Therefore, the quality of the measurements will be critical for the overall decision support system [4].

Stage 2. The development of a solution to provide pollution and meteorological data online to the needs of a web server.

It is suggested to develop a platform for a web application based on OpenACS (Open Architecture Community System – http://openacs.org). This offers a range of services necessary for creating web-based applications:

- a system of access rights defining precisely which functions can be implemented by individual users of the system;
- a secure user authentication using encrypted transmission SSL and user sessions based on a mechanism of digitally signed cookies;
- standard components of the user interface, a template system which allows:
  - separation of the logic of the application and of the presentation layer in accordance with the MVC design pattern (Model – Viewer – Controller),
  - mechanisms for putting application data into packets to allow software modularity,
  - mechanisms to monitor the process of running the application, watchdog, security logs and backup,
  - mechanisms for load-balancing, clustering and database replication, allowing an application to be built within a system which ensures high reliability and scalability,
  - the handling of database connection pools and a mechanism for SQL dialect abstraction of a specific product, the so-called Query Dispatcher.

On the basis of the above-mentioned services, it will be possible to efficiently create web-based applications with high performance and reliability.

Stage 3. Data acquisition and verification

For the selection of decision-making scenarios, it is planned to provide historical data from monitoring the stations of the ARMAAG network and eleven upgraded stations belonging to the Provincial Sanitary-Epidemiological Station, where automated equipment will be implemented. The responsibility for this task will lie with: the Technical University of Gdansk and the ARMAAG Foundation. Data obtained from monitoring stations, and IT solutions for transferring and storing the data will be subject to verification processes. Therefore, it is assumed that the prepared solution should have the following features:

- Automation of the measurement data publishing process and the simplicity and convenience of publishing supplementary information, giving the opportunity to operate an updated Internet service;
- A mechanism to automatically create back-up copies to protect against physical damage to the measurement system and its central station;
- The possibility to adapt the model for the collection of environmental measurements to any system for measuring and collecting results from multiple measurement systems simultaneously;
- The function of automated standardization of measurements, allowing the independence of changes in the configuration of the measuring system;
- The possibility of full control over published volumes of measurements thanks to modular validation of standardized measurement data;

- The possibility to export the measurement data, giving the option of processing them using external specialized tools;
- The publishing of up-to-date messages about environmental quality and detailed measurement data, with access to historical information;
- A simple and convenient user interface, accessible from anywhere on the Internet with a standard web browser, which does not require the installation of client applications;
- A modular construction, which allows the implementation of the system in stages, and helps to respond to a client's needs;
- A remote maintenance system which eradicates the need for employing highly qualified personnel to work on site.

Stage 4. The requirements of decision-making scenarios

It is suggested to develop decision-making scenarios in collaboration with the Municipal and Marshall's Offices. These scenarios will be built based on the requirements of specialist decision makers from the Departments of Spatial Planning and on the basis of zoning plans prepared for the Tri-City. An analysis of zoning plans, selecting a group of experts and the construction of zoning plans with the help of specialists in scenario implementation are also planned.

Stage 5. The construction of decision-making scenarios

Detailed linguistic scenarios are suggested on the basis of the developed groups of scenarios. On the basis of these developed linguistic scenarios, their fuzzy implementation is planned [1,3]. First, the number of input and output variables should be determined. It is assumed that this number should not exceed six in the first case, and three in the second. The choice of variables should be consulted with experts in the field of planning and pollution.

Then the construction of membership functions for input and output variables will be carried out. Multiple verification of parameters and shape is assumed. Then the process of fuzzy modeling, including inference and sharpening processes, will be conducted. Experts will suggest the values of output variables for the inference processes. The staff from the Technical University of Gdańsk will perform the fuzzy modeling processes, as well as coordinating the cooperation with experts from the planning department.

Stage 6. The implementation and verification of decision-making scenarios

The implementation of decision-making scenarios and the fuzzy model will be carried out in two stages. First, using such a design tool as a spreadsheet for the easy modification of scenarios and membership functions. In the second stage, the implementation of the model through a scripting language and its evaluation by the project partners are planned.

It then, in addition, becomes necessary to build a graphical user interface in English, as well as to construct an additional base for storing customer records and their feedback. This solution will shorten the verification time of the system by project partners. After the development of the model, decision-making scenarios will be subject to assessment. As was the case in the previous point, the use of the graphical user interface and the new database to record customer feedback is also suggested here.

Stage 7. Constructing the model of the decision support system

The system will be built based on the architecture of open expert systems [2, 4], consisting of the following modules:

- A knowledge base containing a compartmentalized object description of decision-making scenarios;
- A module of knowledge acquisition – enabling, in dialogical form, the acquisition of linguistic knowledge from experts in the field of spatial planning;
- Inference module – which creates conditions for the processes of prognostic and diagnostic reasoning, based on the decision-making scenarios in the knowledge base;
- Dialogue module – containing commands that allow the user to communicate with the system;
- Explaining module – software for the interpretation of the decision.

Stage 8. The implementation and verification of the decision support system

The result of the implementation process will be a web content management system (CMS), the functions of which will be implemented using the following modules:

- The management of the published information structure, enabling the addition and deletion of pages as well as their organizing into hierarchical structures to be used as a basis for clear navigation, a selection of published content on individual web pages and templates determining the appearance of the pages;
- A content repository for storing and organizing content for publication;
- A system of presentation templates for different types of information stored in the content repository.

The CMS system typically supports content in an unstructured form. The solution also offers the definition of additional attributes of objects which enables the process of storing structured information in the form of records. Structured information related to objects will also be published with the use of presentation templates.

Furthermore, adding support for structured information can simplify the management of large data sets. The CMS system and the content repository will also enable the management and publishing of information in the form of illustrations, files, multimedia, electronic maps, numerical data in graphs, tables, summaries and reports. Its capabilities will also encompass the process of allowing access to objects managed by additional modules, which possess some logic, such as polls, surveys, forms, interactive reports and statistics. The verification of the system will be implemented via expert sessions. Such sessions with the system will be carried out by different partners of the Eureka project. It is planned that the results of the sessions be collected by use of, as was previously mentioned, graphical user interfaces and databases to keep records of the session results. They will form a basis for making changes to the system in accordance with the observations of potential users.

4. SUMMARY

This work presents the possibility of using an MAS based expert system for decision support related to the management of air quality. The suggested approach allows the use of an IT system to monitor air quality in urban areas and to support decisions (the alarm-warning role of the system). The use of fuzzy decision-making scenarios (easy to implement by experts and users of the system) as well as their subsequent processing, will provide users with accurate information about air pollution prognosis.

The system will comply with any spatial infrastructure arrangement, including the specific communication system of the Tri-City and the influence of large companies in shaping the map of air pollution in the Tri-City. The suggested solution is an indispensable tool for the implementation and monitoring of regulations under the Environment Protection Act. The open approach, which assumes the easily adaptable nature of the system, allows the use of the system in any urban area dealing with problems of environmental protection. Making analyze of all disadvantages, advantages as well as agents and agent system use, it is worth to have a think on the outlook of its use to obtained data and scenarios evaluation. It is easy to notice that in the pressure information, intelligent tools bring about selection and information transferring are desirable especially in complex data evaluation. They let to decrease the risk of operating in dynamic environment and in so-called “information hype”.

It seems to be reasonable that the lapse of years the use of agent system to complex data and scenarios evaluation (such as mentioned earlier assistant of decision process) has been increasing. Furthermore it might suppose that apart from assistant of complex system evaluation, agent system increasingly will harness hybrid methods of knowledge representation (for example fuzzy neuron networks). Taking into consideration that an agent is able to acquire information, convert then and infer making certain knowledge and remembering about learning skills, it is plausible of using agent system in WWW structures and the quick implementation of Semantic Web concept within which evaluation system are going to find a use, for example the data and decision-making scenarios evaluation concept which is proposed.

REFERENCES

[1] ORŁOWSKI C., SZCZYGIELSKI Ł., *Conception of interactive information and decision support system for urban and industrial air quality management - model and evaluation*, Springer Verlag, Poznań 2011.
[2] GÓRSKI J., ORŁOWSKI C., *Inżynieria oprogramowania w procesach integracji systemów informatycznych*, PWNT, Gdańsk 2010.
[3] NIEDERLIŃSKI A., *Regułowo-modelowe systemy ekspertowe*, Pracownia Komputerowa Jacka Skalmierskiego, Gliwice 2006.
[4] STEFANOWICZ B., *Systemy eksperckie*, PWN, Warszawa 2003.
[5] ALTMANN J., GRUBER F., KLUG L., STOCKNER W., WEIPPL E., *Using Mobile Agents in real World: A Sufvey and Evaluation of Agent Platforms*, Proceedings of the Second International Workshop on “Infrastructure for MAS and Scalable MAS”, Montreal 2001.
[6] ZIELIŃSKI J., *Inteligentne systemy w zarządzaniu. Teoria i praktyka*, PWN, Warszawa 2000.
[7] HENDLER J., *Is There an Intelligent Agent in Your Future ?*, Nature, 1999.

information system, date, audit.

Dmitriy GENERALOV*, Marian MOLASY**,
Oleg SHLEGEL***

CHECKING OF WHOLENESS CONFIDENTIAL DATA AND AUDIT EVENT IN INFORMATION SYSTEM

The way of counteraction is offered to not authorized access with use of an algorithm filtration audit events information system. The control of integrity confidential data and audit of events in information system allows to prevent and identify a significant amount of attacks and also to reduce the financial losses connected with information leakage.

1. INTRODUCTION

The control of integrity confidential data and booked audit of events in information system allows to prevent and identify a significant amount of attacks and attempts of not authorized access and as to reduce the financial losses connected with information leakage.

By consideration of a problem the control integrity data it is necessary to define the realization purpose. At correct realization of a differentiating policy access to resources their integrity cannot not authorized access be broken. Hence, the control of integrity of resources is necessary for carrying out in a case when there is no possibility of the organization correct differentiation of access, or in that case when the differentiating policy of safety can be overcome the malefactor [1].

Realization of a differentiating policy access to confidential data at applied level is not admissible, as is easily overcome by the malefactor. At applied level problems of the control integrity can be solved only, are based on realization of functions comparison with the original (standard). In the course of information protection at applied level identification is carried out and the fact of occurrence not authorized

* Volga Region State University of Service, Togliatti, Russia.
** Wrocław University of Technology, Wrocław, Poland.
*** Volga Region State University of Service, Togliatti, Russia.

event attack insiders is fixed. The basic advantage of protection at applied level is that the occurred not authorized event can, is registered in any case, without dependence from the reasons of its occurrence. Not authorized event cannot be prevented, as at applied level recording, hence, probably operative reaction to event is carried out only.

One the basic mechanisms of protection confidential data in information system from not authorized access is the mechanism of maintenance isolation the program environment. Thus it is necessary to forbid start of foreign processes and appendices, without dependence from a way their introduction [2]. The given problem at system level in such a manner that the module of protection frames fixes dares and analyzes all inquiries about start of an executed file, providing start possibility only the resolved processes and appendices. At applied level the given problem by the analysis of a priority processes and the started appendices for the purpose of revealing not authorized event and its subsequent blocking dares. Protection mechanisms solve unequivocal problems, both at system level, and at applied level, supplementing each other. At realization of effective protection information system the primary goals of protection confidential data should dare simultaneously in two ways.

2. MECHANISMS OF THE CONTROL INTEGRITY THE DATA FOR INFORMATION SYSTEM

Statistical data of financial losses the companies of the Russian Federation show (fig. 1), that in the course of statement problems protection of the information realized by means modules the control integrity confidential data and audit of events, are carried out not effectively enough, and also problems of the decision protection counteraction not authorized access the malefactor are not allocated.

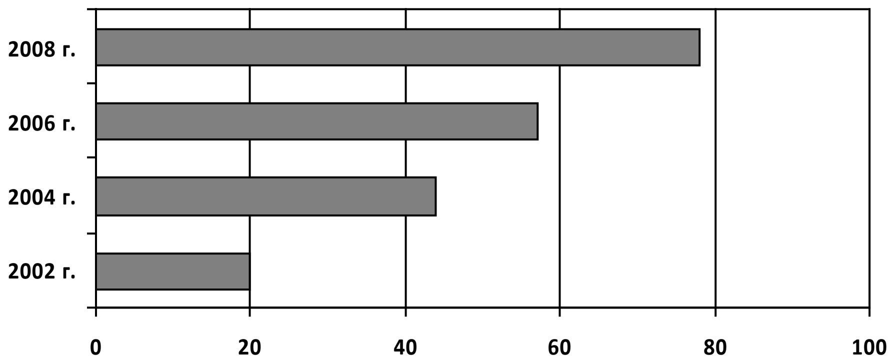


Fig. 1. A parity of financial losses (%) the companies, caused by leak of confidential data from information system on years (2002-2008 of)

Mechanisms of the control integrity the data, in information system, the method of «a cyclic control code» and unidirectional functions hashing is applied.

Way of the control integrity data is, a method of «a cyclic control code» (Cyclic Redundancy Check – CRC). The basic advantages and method lacks of «a cyclic control code» include various positive and negative sides (fig. 2).

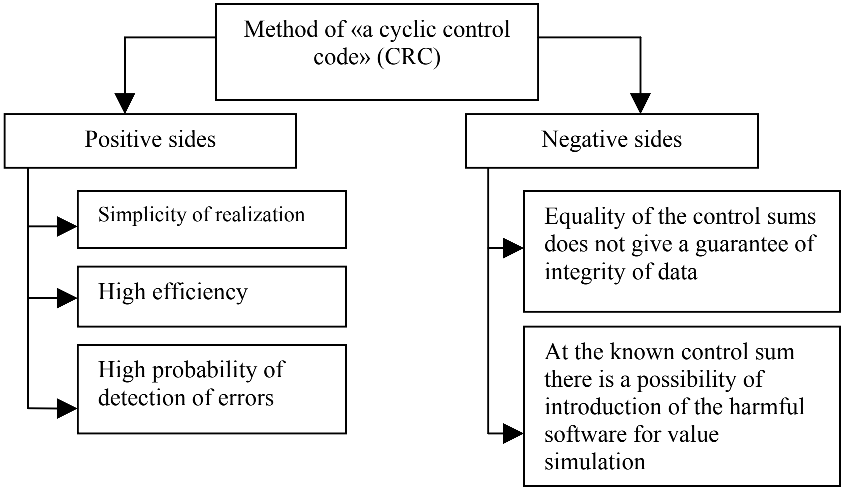


Fig. 2. Advantages and method lacks of «a cyclic control code» applied to the control of integrity data in information system

The algorithm of the control integrity data is used for verification an invariance of the entrance and target information, and also in many information systems for revealing of errors at data transmission on communication channels [3].

At the heart of method CRC the concept a polynom or a multinomial lays. Each bit of some block data corresponds to one factors a binary polynom. Any block of confidential data in information system represents sequence of bits which it is possible to present in the form of binary polynom A (x). One more polynom G is necessary For calculation of a control code (x), named a generating polynom. For each realisation of algorithm control CRC the generating polynom gets out in advance arbitrarily, for example, $R(x)$ – the certain polynom which is a control code of polynom $A(x)$ at generating polynom $G(x)$, at $R(x)$ being remainder of division of polynom $A(x)$ x^r on $G(x)$, where r – degree of polynom $G(x)$.

$$R(x) = (A(x) \cdot x^r) \bmod G(x). \qquad (1)$$

Use of method CRC allows to identify shift of two byte or unit addition to one and subtraction unit from another. By means of a method «a cyclic control code» it is possible to find out casual changes in confidential data of information system [4].

In information system those events which are necessary for supervising and recording (fig. 2) are allocated only. The reference copy of checked group (the list of the authorized events) is for this purpose created and personal level of the control is formed.

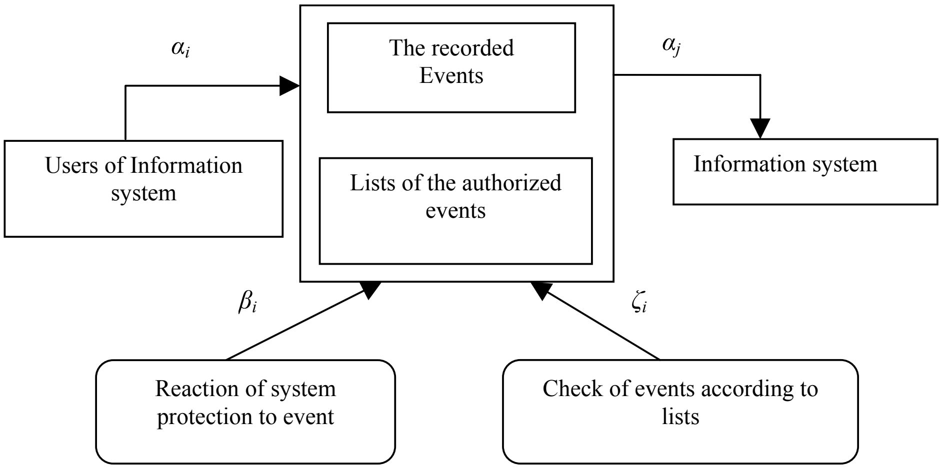


Fig. 2. Process of the control actions of the user in information system

3. MECHANISMS OF THE CONTROL INTEGRITY CONFIDENTIAL DATA ON INTERCONNECTED WITH AUDIT

Process of the control actions the user in information system consists in the control of events by means reference copies of lists and values. At a divergence the condition of infringement safety system after that there is a restoration of information system from a control point is generated. Restoration from a control point allows restoring information system in an initial working condition.

Mechanisms of the control integrity confidential data are interconnected with audit of events by that critical data of audit are transferred to protection frames the information for operative counteraction to not authorized access of the malefactor to information system. The basic mechanisms of protection information system as a part of protection frames the information from not authorized access are the system drivers realizing the differentiating policy of access. The data transferred in audit, in most cases help to provide operative reaction to not authorized access инсайдера and to provide the subsequent control.

Realization of mechanisms the control at construction system audit safety allows to allocate two levels audit differing with modes of processing inquiries: processing in real time (automatically) and interactive processing (on demand). Division of audit into levels allows to raise efficiency processing arriving data about critical events occurring in information system, and also to provide reaction in real time for not authorized access. In real time data of the second level audit are transferred only – the data of the audit registered considered mechanisms protection applied level (the integrity control), data of the first level audit (registered by mechanisms of the control of access to resources) can arrive in an interactive mode (by its inquiry, according to any regulations) [5, 6].

Generally it is possible to allocate some problems of audit which should be present at protection frames of information system: tool audit of events (fig. 3a); interactive audit of safety (fig. 3b); operative audit of safety.

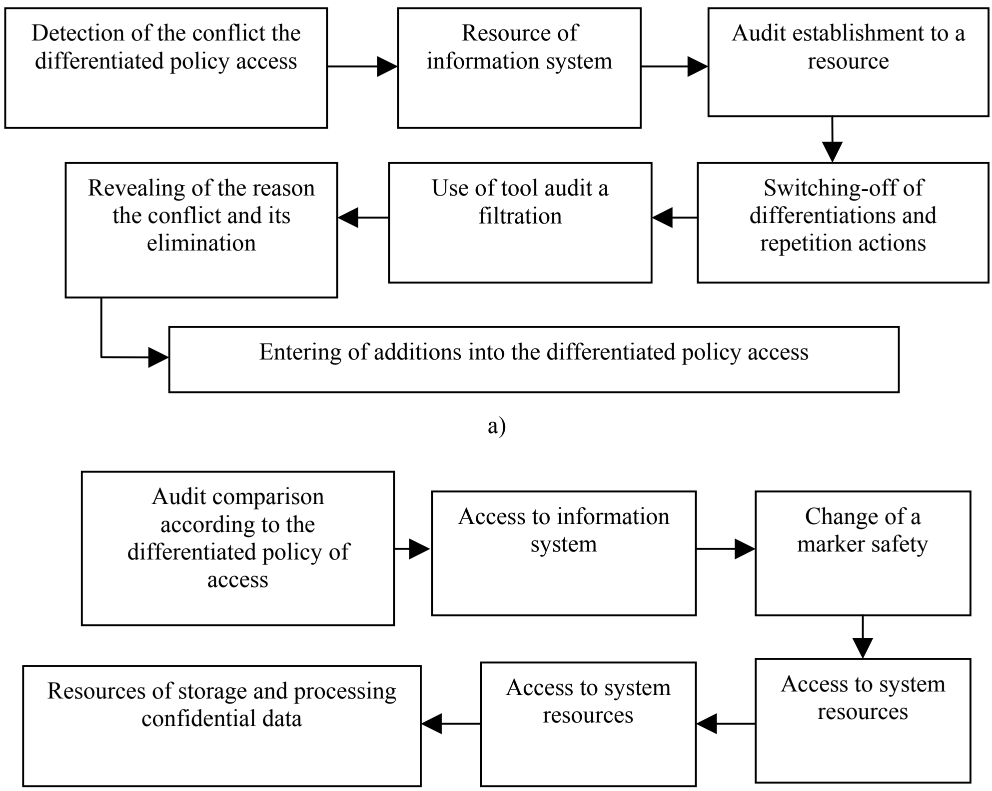


Fig. 3. The Primary goals of audit the authorized and not authorized events, data of information system integrated into protection frames:
a) – sequence of problems carried out in tool audit at revealing not authorized access the malefactor to information system;
b) – sequence of options mechanisms protection for interactive audit for revealing not authorized access the malefactor to information system

Mechanisms of the control integrity data are one the primary goals protection information system from not authorized actions the malefactor. The control mechanism registers the fact occurrence of attack, hence, it is necessary to have a sufficient set of mechanisms for the control information system [7, 8]. Realization of a sufficient set mechanisms the control integrity data allows to construct system of effective and operative audit.

The control system of information safety in information system solves protection against not authorized access of the malefactor. The substantive provision in a control system of information safety is necessary on system of operative audit [9]. The system of operative audit of safety is intended for association of means audit safety data used as a part information system. Operative audit allows to lower considerably risk from the possible damage put to data at infringement politicians of safety and loss of integrity.

4. THE WAY OF COUNTERACTION WITH USE ALGORITHM FILTRATION AUDIT EVENTS TO NOT AUTHORIZED ACCESS TO INFORMATION SYSTEM

The algorithm of a way counteraction with use algorithm a filtration audit events to information system is presented not authorized access in fig. 4.

Action of a way is based on the following. After start of a session information system 1 there is an initialization of the user 2, further data are transferred to the central computing processor 3 for processing, and is started authorized a part of information system 4. Further the information is transferred to the module of a choice an operating mode information system 5.

If the information system is in an operating mode the filtration of audit events on blocks 6 is carried out. Further cycles of inputs/conclusion values of a file 7, 8, 9 are started. After passage of cycles assignment to 10 variables for values a file is carried out, and cycles inputs/conclusion of values a file 11, 12 are started. The cycle passes all values of a file at performance a condition 13, in case of positive result a condition calculation of the sums the authorized events 14 is carried out. If the condition 13 is not carried out, there is a further search of values a matrix 12, 11, after there is an exit from information system 16.

If the information system is in a “sleeping” operating mode 15 the filtration of all audit events is carried out. After filtration performance there is an elimination of the authorized and not authorized events and the exit from information system 16 is carried out.

In the offered way of prevention not authorized access main principles a substantiation procedure placing problems in system of a filtration events with the matrix topological organization events audit connected with the authorized and not authorized access have been used. The matrix filtration of audit event is represented in the form of superposition ring structures. Basic "part" of communications problems takes places in these rings, and remained not placed communications "keep within" between rings (work: Information systems on enterprise of the service – Erochina, Molasy, Shlegel, Vlasilthuk).

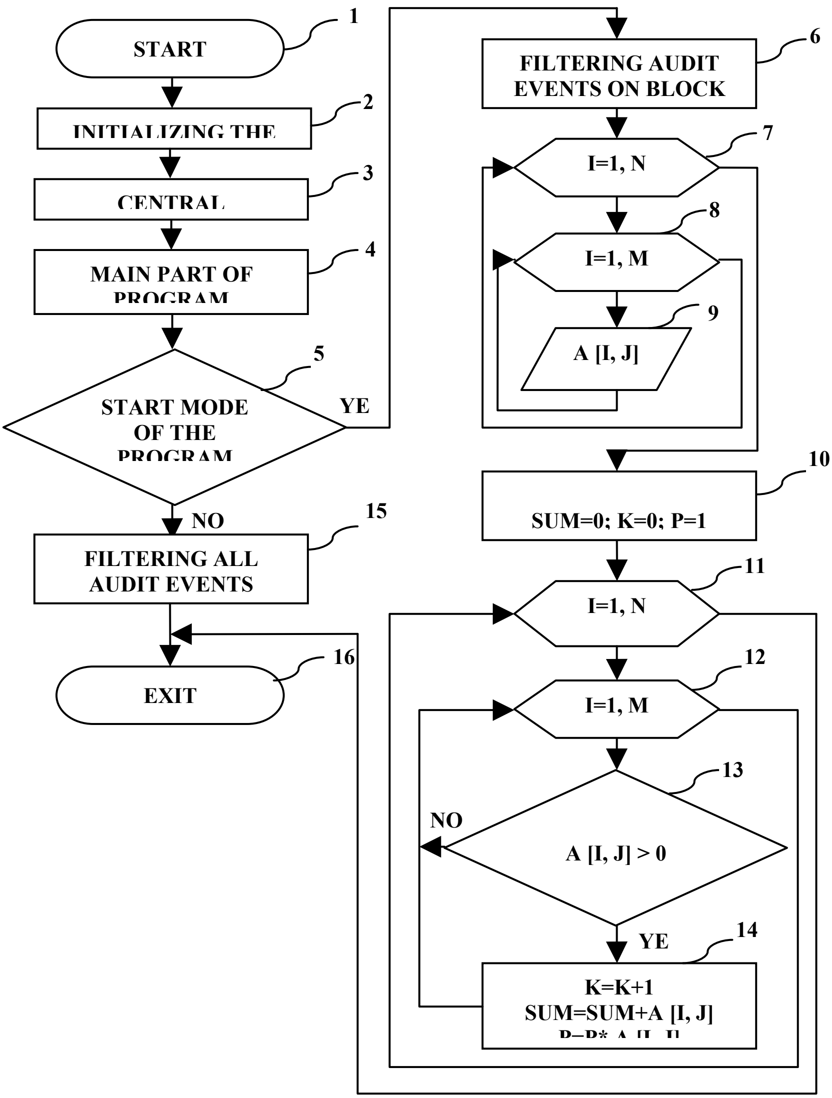


Fig. 4. Algorithm of a way counteraction with use algorithm a filtration audit events to not authorized access to information system

5. TRANSFER AND CODING OF THE ECONOMIC INFORMATION

The code consisting of several simple codes refers to complex. The complex code refers to connected if raisings included in form some uniform concept. If in a code each code included in it is represented with independent value the code refers to untied. The serial code is serial numbering of positions of the nomenclature located in beforehand caused sequence.

The serial system of coding is the most simple. To apply it is recommended at coding steady nomenclatures as this system does not provide an opportunity of expansion of the nomenclature.

The decimal code for each kind of attributes provides quantity of numbers.

Decimal code applies to an encryption of materials, the process equipment, etc. Advantage of a decimal code is its universality and simplicity of construction, and lack – strictly certain number of positions for each attribute (10, 100, 1000, etc.) irrespective of real need that results in significant redundancy of this code.

The serial code represents a series of the serial numbers allocated for group of positions of the nomenclature, possessing the general attribute. The serial information allows increasing quantity of positions of the nomenclature within the limits of a series that it is impossible to tell about serial numbering. Serial codes are convenient for using at complex nomenclatures which precisely are grouped, exposed to changes and reception of many results to various attributes demand.

At a choice of a variant of technological process of processing the economic information use two groups of parameters an estimation of efficiency: reliability of reception and processing of the information and labor and cost expenses for designing of system and processing of the information.

For maintenance of performance of these requirements it is necessary to choose first of all high-efficiency and reliable technical base, to develop structure of the basic operations and methods of their realization. However for achievement of high reliability of processing and reception the designer should the information, besides it, organize the monitoring system behind reliability of processing of the information. For development of such system the designer is obliged to analyze frequency of occurrence of mistakes as decided problems, on classes of operations of technological process, by kinds of mistakes and owing to their occurrence. With this purpose it is necessary to collect statistics of mistakes and to receive distribution of frequency of their occurrence on the following directions:

- two kinds of decided problems: for example, analytical, scheduled, statistical, registration;
- two classes of operations technological process;

- two kinds of the mistakes connected to a condition of primary documents, with carry of the data on machine carriers, with processing on a computer, with the control and release documents;
- two the reasons of occurrence mistakes: negligence of users and bad development of operations on input of the information in a computer (fault of the executor of documents), mistakes in the project (fault of designers). Then it is necessary to choose the certain quality monitoring behind each operation or group of operations and to execute an estimation of a degree of reliability received after processing information.

At a choice of the best technological process of processing of the economic information, besides use of parameters of reliability, apply an estimation, comparison and a choice on a parity of a performance level of this or that variant of process and value of sizes of parameters of labor and cost expenses for designing and operation of these processes.

The basic requirements to economic codes are:

1. Codes should be developed in view of the basic information carrier used in projected system.
2. It is necessary to provide reliability and reliability of the transmitted information for what it is necessary to provide an opportunity of the control and correction of accepted codes.
3. The code should be under construction in view of an opportunity of increase in the coded nomenclature and possible necessity of joining with the adjacent enterprises and the organizations.
4. By development of codes it is necessary to aspire to theoretically minimal average length of code words.
5. One should meet to each code and only one kind of an industrial output or any other coded economic parameter.
6. The system of coding should be under construction in view of use of perspective means of processing of the information.

Difference in treatment of the term the information and bases of the information in economy that in the first case pithiness of the information is completely ignored, in the second – quantity of the information depends on, when to whom, that is transferred.

Utility of the information circulating in the given system should be defined on reaction of system to the given kind of the message.

The codes intended for transfer and processing of the economic information should be under construction in view of use of perspective means of processing the information, kinds of the basic information carriers and an opportunity the control of accepted codes. They also should meet to the certain requirements.

By development of codes it is necessary to aspire to theoretically minimal average length of code words.

6. SUMMARY

Application of the control integrity confidential data and audit of occurring events in information system allows lowering quantity the attacks connected with not authorized access, to prevent financial losses at the expense of blocking channels information leakage. Mechanisms of the control integrity confidential data and audit of events can be applied in a complex to more effective protection of information system against not authorized access.

REFERENCES

[1] MOLASY Marian, MOLASY Mateusz, SHLEGEL A., SHLEGEL O., *Innovations in Management of Personnel Potential of an Enterprise*, SYSTEMS, Journal of Transdisciplinary Systems Science, Wrocław University of Technology, Poland, Volume 10, Number 2, 2005.

[2] ГЕНЕРАЛОВ Д., *Повышение безопасности и аудит информационной системы мобильных устройств*, Вестник Поволжского Государственного Университета Сервиса серия «Экономика», № 6. – Тольятти: Изд-во ПВГУС, 2009.

[3] ГЕНЕРАЛОВ Д., ШЛЕГЕЛЬ О., *Угрозы и политика безопасности информационных ресурсов для мобильных устройств*, VI Международная научно-практическая конференция «Синергетика природных, технических и социально-экономических систем». – Тольятти: Изд-во ПВГУС, 2009.

[4] ГОСТ Р34.11-94, *«Информационная технология. Криптографическая защита информации. Функция хэширования»*, Постановление Госстандарта России от 23.05.94, № 154.

[5] ШЛЕГЕЛЬ О., *Аудит безопасности информационных систем предприятия*, Материалы Международной научно-практической конференции Ч.1.– Саратов: Издательство «Научная книга», 2008, 355с.

[6] ШЛЕГЕЛЬ О., *Интегрированные информационные технологии с учетом особенностей структуры, специализации организации*, Материалы научно-методического семинара «Научно-методическое обеспечение инновационного развития образовательной деятельности», Тольятти, 2008.

[7] SCHNEIER B., *Applied cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, 1995.

[8] SCHWADERER W., DAVID C., *Programmer's Guide to Netbios*, Howard W. Sams&Company, 1988.

[9] CLARK D., WILSON D., *A comparison of Commercial and Military Computer Security Policies, Proc. of the 1987 IEEE Symposium on Security and Privacy*, Oakland, Cal., 1987.

information systems, audit event,
information control systems

Lidija EROCHINA*, Marian MOLASY**,
Oleg SHLEGEL***, Olga VASILTHUK****

INFORMATION SYSTEMS ON ENTERPRISE OF THE SERVICE

Information systems allow the enterprise to carry out as the internal financial analysis, so external from a position of the budget, supervising departments, creditors and shareholders. The way of counteraction is offered to not authorized access with use of an algorithm filtration audit events information system.

The control of integrity confidential data and audit of events in information system allows to prevent and identify a significant amount of attacks and also to reduce the financial losses connected with information leakage.

1. INTRODUCTION

The word "system" means the whole, made of the parts connected with each other. Other definition is given with cybernetics. Here the system is a set connected with an environment of elements or parts which functioning is directed on reception of concrete useful result. Each economic object can be considered as the system aspiring in the functioning to achievement of a definite purpose (fig. 1).

Complexity of system depends on components included in it and from complexity of internal and external communications [1, 2, 3]. The system is divided, if it will consist of lines of subsystems or the elements allocated to a certain attribute, adequate to specific goals. For example, in structure of system «Enterprise» subsystems can be

* Togliatti State University of Service, Togliatti, Russia.
** Wrocław University of Technology, Wrocław, Poland.
*** Togliatti State University of Service, Togliatti, Russia.
**** Togliatti State University of Service, Togliatti, Russia.

allocated: shops, services of management and so on. Integrity of system means, that all its elements are subordinated to the uniform purpose.

The variety of elements of system and distinction of their nature are defined by their functional specificity and autonomy. The raw material, the basic and auxiliary materials, fuel, semi finished items, and spare parts, finished goods, and monetary resources in material sphere are such elements.

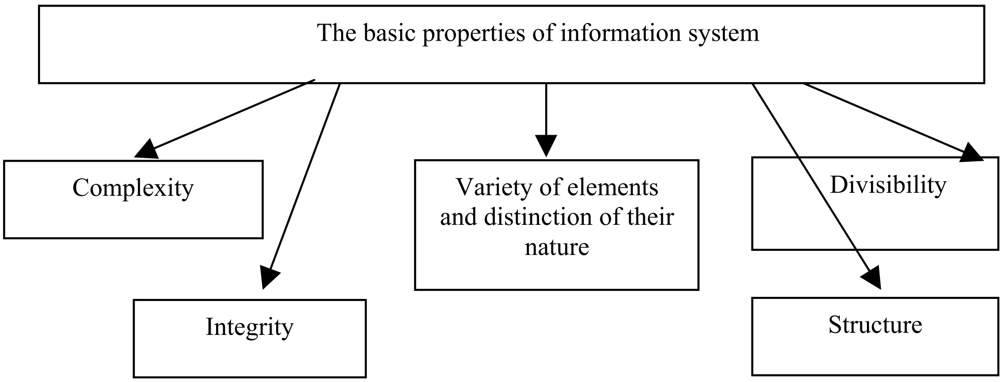


Fig. 1. Properties of information system

2. STRUCTURE OF SYSTEM MEANS DISTRIBUTION OF ELEMENTS SYSTEM

Structure of system means distribution of elements system on levels of hierarchy. Systems carry out many functions. One of the main functions is management.

Management is the kind of the purposeful activity providing a coordination of all carried out works for achievement of certain result [4, 5]. Management is connected to information interchange between components of system and with an environment. In managerial process receive data on a condition of system during each moment of time, about achievement (or not achievement) the set purpose to influence system and to provide performance of administrative decisions.

The system realizing functions of management, name a control system (fig. 2).

In a control system of economic object of any level it is possible to allocate managing and controlled subsystems. The managing subsystem carries out functions of management, establishes overall aims of functioning of economic object. As a managing subsystem at the enterprise divisions and services of management personnel act: a staff department, accounts department. The managing subsystem on behalf of heads of divisions and services of management personnel uses data on industrial-economic activities of economic object and the information from an environment for development and acceptance of administrative decisions which are transferred in a controlled subsystem (fig. 3).

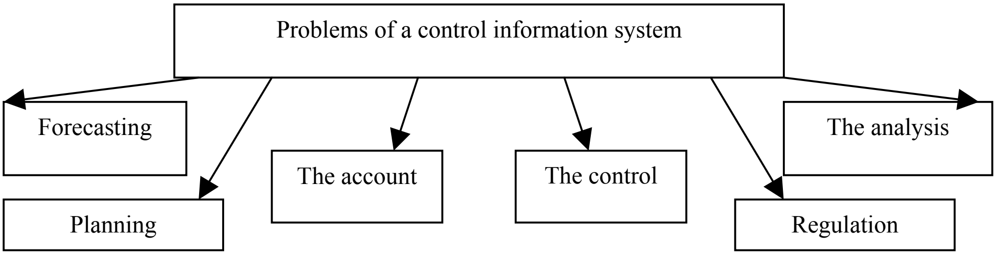


Fig. 2. Problems of a control economic information system

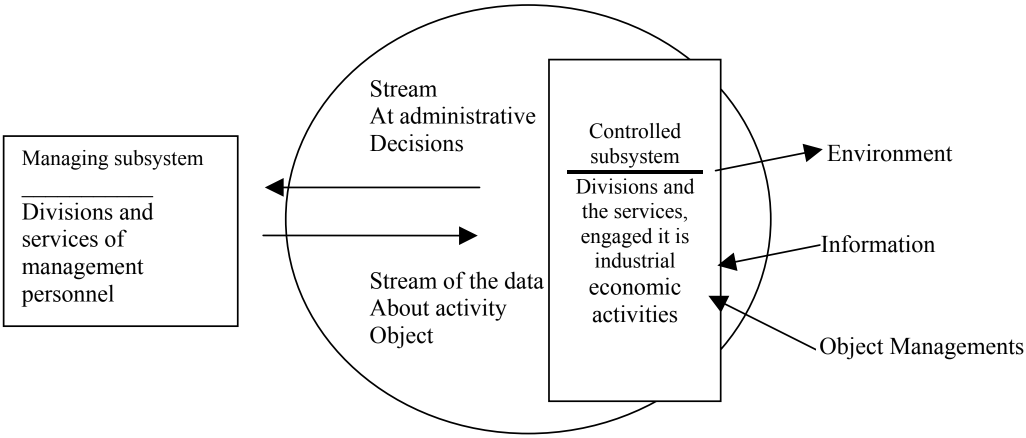


Fig. 3. A control information system

The controlled subsystem carries out the functions connected to manufacture and release of finished goods or performance of socially necessary works.

Managing and controlled subsystems have a feedback. The information is a kind of communication which arises in managerial process. Due to it influence of a managing subsystem on controlled, and on the contrary is carried out. Thus, to any control system there corresponds the information system, for example, to a control system of economic object – economic information system.

The economic information system is a set of internal and external streams of direct and return information communication of economic object, methods, means, the experts participating during processing of the information and development of administrative decisions.

3. INFORMATION CONTROL SYSTEMS OF TECHNOLOGICAL PROCESSES

Information control systems of technological processes are most widely applied in various kinds of the industry, to management of fusible steel, process of reception of pig-iron, etc.

With the help of information systems of organizational management the management of the big collectives of people is carried out. Examples of some such information systems are the systems submitted in the circuit 4.

In automatic systems all operations of management are carried out with the help of a computer automatically. The role of the person in these systems is reduced only to supervision over work of machines and performance of functions of the control. Automatic systems are applied to management of technical objects and technological processes (fig. 4).

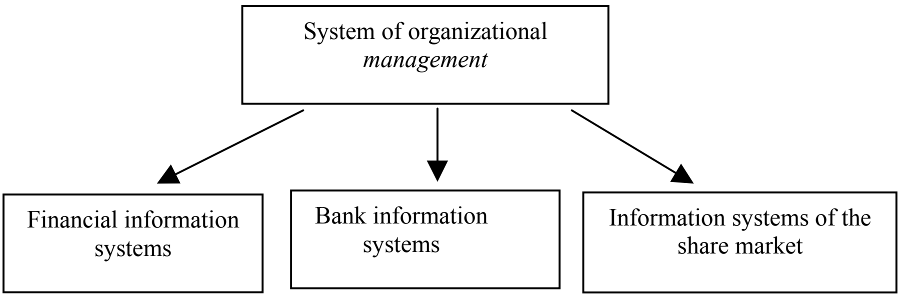


Fig. 4. Systems of organizational management

The structure of providing subsystems does not depend on the chosen subject domain. It can change depending on complexity of information system. The providing part promotes effective functioning of system as a whole and its separate subsystems. All providing subsystems are connected among themselves and to functional subsystems.

Providing subsystems are of great importance for the enterprises. Classification of providing subsystems is shown in fig. 5.

Any system has the structure. The structure of system is the organization of its separate elements in view of their interrelations and the purposes put before system. The structure of functional subsystems is defined by features of economic system, its branch accessory, character of activity of the enterprise. Their structure can be added depending on specificity of conducting book keeping in the certain branch.

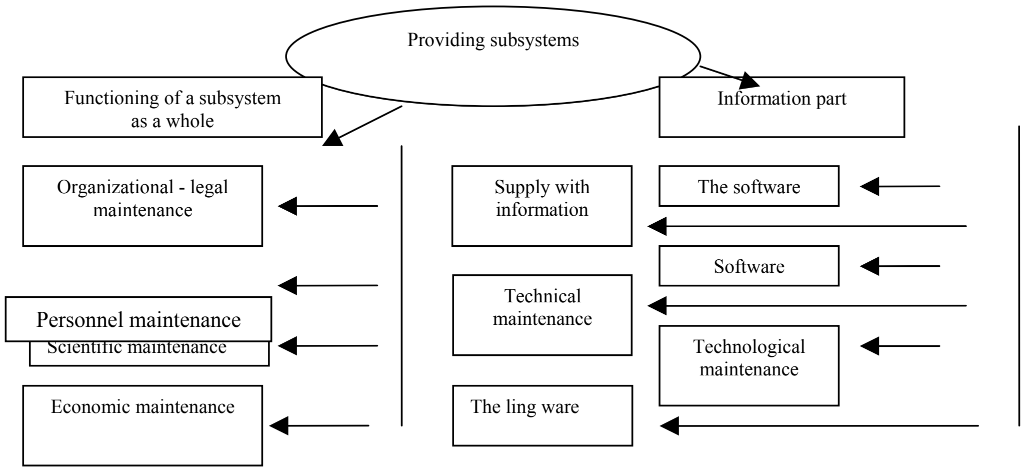


Fig. 5. Classification of providing subsystems

Subsystems of scientific and economic maintenance are called to carry out the problems connected to development of system effectiveness. Ergonomic maintenance represents set of methods and means for fast development of information technology, qualitative and correct work with information system. The subsystem of a supply with information is one of the major and includes the information on object. The subsystem of technical maintenance represents a complex of means, the engineering specifications on use ES. The subsystem of software will consist of mathematical methods, models and algorithms of processing of the information, used at creation of system. The subsystem of the software includes all variety of the typical and standard programs realizing the decision of problems on a computer for all functional subsystems. Technological maintenance – set of the design decisions determining technology of processing of the information at all technological stages of gathering and registration of the primary information, preparation and the control of files and databases, release of documents. The subsystem of the ling ware represents set of scientific and technical terms and the language means used with a view of simplification of dialogue of the personnel with computers. The structure of providing subsystems is defined at a design stage of information system and can be various depending on the purposes and problems of economic object. The information is one of the major resources of a society, as well as traditional material, financial resources so, process of processing of the information can be defined as technology. The information technology is a system of methods and ways of gathering, accumulation, storage, search, transfer and processing of the information on the basis of application of means of computer facilities. The purpose of information technology – manufacture of the information for its analysis the person.

4. THE PARAMETER OF COST EXPENSES ON TECHNOLOGICAL INFORMATION PROCESS

The parameter of cost expenses on j – technological process (C_j) of processing of the economic information represents the sum of expenses under following clauses: wages; amortization; materials; payment of machine time; conducting information base; an overhead charge [6]

This parameter pays off:

$$C_j = \sum_{i=1}^{n} C_{ij} \tag{1}$$

where C_{ij} – a parameter of cost expenses on I – operation j – technological process which structure includes the following components:

$$C_{ij} = C_{3/n} + C_{nr} + C_a + C_{MB} + S_m + C_{иб}, \tag{2}$$

where $C_{3/n}$ – expenses for wages of the operator which can be calculated:

$$C_{3/n} = t_{ij} r, \tag{3}$$

where t_{ij} – labour input of performance i – operations j – technological process; r_i – the tariff rate i-th operations; C_{nr} – expenses for an overhead charge considered as derivative size from expenses for wages:

$$C_{nr} = C_{3/n} \times K_{np}, \tag{4}$$

where K_{nr} – size factor of the overhead charge, accepted, as a rule, at a rate of 0.6–0.7 from size $C_{3/n}$, C_a – size of depreciation charges on the used technics:

$$C_a = t_{ij} a_i, \tag{5}$$

where a_i – norm of depreciation charges; C_{MB} – cost of machine time for input of the information in a computer, data processing and delivery information:

$$C_{MB} = t_{Mj} \times c, \tag{6}$$

where c – cost of machine hour; t_{Mj} – duration of performance mine of machine operation j – the technological process, including the following components:

$$t_M = t_1 + t_2 + t_3, \tag{7}$$

where t_1 – duration of performance operation input of the initial information in a computer.

The structure of procedures of transformation of the information and feature of their performance in many respects depend on the economic object conducting automated processing of the information. Usually economic information is exposed to all procedures of transformation. The sequence of their performance also happens various, thus some procedures can repeat.

5. THE MATRIX TOPOLOGICAL ORGANIZATION OF EVENTS AUDIT

In the offered way of prevention not authorized access main principles a substantiation procedure placing problems in system of a filtration events with the matrix topological organization events audit connected with the authorized and not authorized access have been used. The matrix filtration of audit event is represented in the form of superposition ring structures. Basic "part" of communications problems takes places in these rings, and remained not placed communications "keep within" between rings [7, 8].

Registered events of the module a filtration audit events can be presented in a kind column G. The filtration of data at audit is set by topological model in a kind column $H = \langle M, V \rangle$, where

$$M = \begin{vmatrix} m_{11} & m_{12} & \dots & m_{1k} \\ m_{21} & m_{22} & \dots & m_{2k} \\ \dots & \dots & \dots & \dots \\ m_{t1} & m_{t2} & \dots & m_{tk} \end{vmatrix} \tag{8}$$

where M – set of events different modules audit of protected information system from not authorized influence; $||M||t \times k,\ t = \overline{1, N},\ k = \overline{1, N},\ V$ – set of intermodular communications. Placing column G in system of audit H is set in the form of display

$$\beta : \begin{vmatrix} x_{i_{11}} & x_{i_{12}} & \dots & x_{i_{1k}} \\ x_{i_{21}} & x_{i_{22}} & \dots & x_{i_{2k}} \\ \dots & \dots & \dots & \dots \\ x_{i_{t1}} & x_{i_{t2}} & \dots & x_{i_{tk}} \end{vmatrix} \Rightarrow \begin{vmatrix} m_{11} & m_{12} & \dots & m_{1k} \\ m_{21} & m_{22} & \dots & m_{2k} \\ \dots & \dots & \dots & \dots \\ m_{t1} & m_{t2} & \dots & m_{tk} \end{vmatrix} \quad (9)$$

The placing problem is formulated as display search such, that

$$R^{\Sigma}_{\beta^*} = \sum_{m_{q,w}, m_{g,j} \in M} R(m_{q,w}, m_{g,j})^{\beta^*} = \min_{\beta \in \psi} \{R^{\Sigma}_{\beta}\}, \quad (10)$$

where $R(m_{q,w}, m_{g,j})^{\beta^*}$ – total intensity of interactions between elements a matrix of the module audit $m_{q,w}$ and $m_{g,j}$, corresponding to display β^*.

The quantity of the authorized events can be found under the formula and further can be used in algorithm of a filtration audit events for a finding not authorized event. Matrixes of a two-dimensional file also can be used as elements of audit events a matrix a three-dimensional file a filtration:

$$Y = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} A_{ijk} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} B_{ijk}, \quad (11)$$

where Y – the sum of authorized events A_{ijk}, B_{ijk} filtrations of audit information system.

Columns G is coherent, thus appointment of several subtasks to the same module of audit events is not supposed. All matrix system of audit events is represented in the form of superposition ring structures. In the course of top placing, incidental to arches with the greatest intensity are fixed on the nearest modules in a matrix in lines (rings), and arches with the least values of intensity are appointed to tops between lines (rings) of system a filtration of audit events. The choice of fixed arches is defined by the bottom estimation of intensity interaction modules audit.

6. SUMMARY

Let's consider features of performance of the basic procedures of transformation the information:

– Gathering the information is accompanied by its registration on the material carrier.
– Transfer of the information to a place of its accumulation, storage and processing in a case. Ways of transfer of the information can be various: with the help of the courier, transfer by mail, delivery with the help of vehicles, remote transfer on liaison channels.
– Record of the information on machine carriers by carry of the data from the primary document on machine carriers.
– Storage of the information is carried out as information files on machine carriers.
– Processing the economic information on a computer at which above the data logic and arithmetic operations are carried out.
– Delivery of results of processing on a seal, the screen of the display, depending on variants of further use of results.
– Duplicating results of processing of the economic information with the purpose of transfer to its various users.
– All this demands, that economists, being the basic users of computer information systems, owned bases of information technology, able to estimate actions of information systems, quality of processing, accuracy, completeness of the information for acceptance of administrative decisions.

Application of the control integrity confidential data and audit of occurring events in information system allows to lower quantity the attacks connected with not authorized access, to prevent financial losses at the expense of blocking channels information leakage. Mechanisms of the control integrity confidential data and audit of events can be applied in a complex to more effective protection of information system against not authorized access.

REFERENCES

[1] MOLASY Marian, MOLASY Mateusz, SHLEGEL A., SHLEGEL O., *Innovations in Management of Personnel Potential of an Enterprise*, SYSTEMS, Journal of Transdisciplinary Systems Science, Wrocław University of Technology, Poland, Vol. 10, No. 2, 2005.

[2] ГЕНЕРАЛОВ, Д., ШЛЕГЕЛЬ, О., *Угрозы и политика безопасности информационных ресурсов для мобильных устройств*, VI Международная научно-практическая конференция «Синергетика природных, технических и социально-экономических систем», Тольятти, Изд. ПВГУС, 2009.

[3] ГОСТ Р34.11-94, «*Информационная технология. Криптографическая защита информации. Функция хэширования*», Постановление Госстандарта России от 23.05.94, № 154.

[4] ШЛЕГЕЛЬ О., *Аудит безопасности информационных систем предприятия*, Материалы Международной научно-практической конференции Ч. 1, Саратов, Издательство «Научная книга», 2008, 355 с.

[5] ШЛЕГЕЛЬ О., *Интегрированные информационные технологии с учетом особенностей структуры, специализации организации*, Материалы научно-методического семинара «Научно-методическое обеспечение инновационного развития образовательной деятельности», Тольятти, 2008.

[6] BRUCE, SCHNEIER, *Applied cryptography: Protocols, Algorithms and Source Code in C*, John Wiley & Sons, 1995.

[7] SCHWADERER W., DAVID C., *Programmer's Guide to Netbios*, Howard W. Sams & Company, 1988.

[8] CLARK D., WILSON D., *A comparison of Commercial and Military Computer Security Policies*, Proc. of the 1987 IEEE Symposium on Security and Privacy, Oakland, CA, 1987.

OCL, contracts, UML, CASE tools,
code generation

Anna DEREZIŃSKA
Przemysław OŁTARZEWSKI*

CODE GENERATION OF CONTRACTS USING OCL TOOLS

The idea of contract-based programming can be realized at different stages of software development. Contracts can be inherited according to the rules defined for Eiffel, the language developed to carry out the Design by Contract™ paradigm. The Object Constraint Language (OCL) supports contracts assigned to models. Using OCL, invariants of classes, pre- and post-conditions of operations, as well as constraints in behavioral models can be specified. Various tools provide object-oriented modeling with UML and OCL, but only scarce of them can generate code of OCL contracts. Based on conducted experiments we evaluated their capabilities to transform OCL contracts into the corresponding code. The tools were compared in respect to different structures of OCL, contract inheritance, target languages, programming techniques realizing contracts, and system reactions in case of not satisfied contracts. Limitations in realization of contracts in the OCL tools were discussed. The reviewed tools were related to T.O.F.I.C. – a new tool that transforms UML class models, refined with the C#-aimed profile and specified with OCL expressions, into the appropriate C# code. In this approach, OCL contracts are established with the Microsoft Code Contracts library of .NET.

1. INTRODUCTION

High quality developed software should be consistent with its specification. Tight coupling of specification constraints with a code that implements them, combined with the code verification against satisfaction of those constraints, could improve the software quality. This idea can be realized using a concept of contracts. A contract is a kind of agreement that should be satisfied by all its participants. For example, a class invariant is a condition that should be true for all objects of the class in all their stable states. A pre-condition should be satisfied in a state before a required functionality,

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

e.g. a method/operation of a class, is executed. If a pre-condition was true, the satisfaction of the appropriate post-condition should be guarantied by the programming unit after the operation execution. Application of such ideas was proposed by Mayer in the Design by Contract [7] approach, and the Eiffel language supporting contracts [5]. Hence DbC™ is a trademark in the USA, the idea applied in different notations is also denoted as contract programming, contract-based development, or analogous.

Contract-based approaches can be realized at a code level. For this purpose different languages directly supporting contract concepts were developed, as Eiffel, Cobra, D (a variant of C++), Fortress (a modification of Fortran), SPARK (a clone of Ada), Oxygene (for .NET), Spec# (a variant of C#), etc. Various libraries were also proposed that extend languages and tools with contract capabilities, for example DBC for C/C++; Contract4J, jContractor for Java; JavaDbC, OVaL, Custos for AspectJ; SpringContracts in Spring Framework, Microsoft Code Contracts for C#.

Contracts describe constraints originating from different sources, such as business requirements, applied technologies, performance requirements, etc. They can be defined at various levels of a system development. Therefore, principles of design by contract can also be fulfilled at earlier stages, i.e. in models created during the software design. Software models are often expressed in the UML language [12], or other graphical notations consistent with the MOF meta-model. They can be refined with contracts written in OCL or other specification languages. Precise modeling can also be realized using graphical notations conformed to meta-model EMOF (Essential MOF) together with specification languages of the Epsilon family, etc. In Validation Framework for .NET contracts are specified in ACL (Another Constraint Language).

Object Constraint Language (OCL) [8, 13], primarily developed in IBM, belongs to one of OMG notations recommended for UML models. The language is used to express constraints enhancing object-oriented models in order to get more detailed and precise specification. OCL expressions are without side effects, i.e. they cannot make changes to the model. Using OCL we can describe class invariants, initial and derived conditions of attributes as well as pre-, post- and body conditions of operations. Apart from class models, OCL expressions can specify invariants of states in statemachines, and guard conditions in sequence, activity or state diagrams.

If contracts are introduced in an object-oriented design, regardless whether in an O-O programming language like Java, C#, or in an O-O modeling language like UML, we face an inheritance issue. Classes related by inheritance can be enhanced with contracts. The rules of the contract inheritance were proposed by Meyer in the context of the Eiffel language [5], but can be generalized for other notations. Basically, the invariant inheritance rule states that a derived class should satisfy all contracts of its base classes. In an operation of a derived class, pre-conditions can weaken that in the corresponding operation of the base class. In case of post-conditions there is the opposite rule, the post-conditions can be stronger or equal to the corresponding, derived ones. An expression *p* is denoted as stronger than *q* if *p* logically implies *q*.

Model-driven engineering paradigm assumes that models are gradually refined to create an executable code, or at least a code skeleton. If so, contracts specified at model level should be transformed into corresponding contracts in the target code. In this chapter, we review the tools that support code generation from contracts that are combined with models and specified in OCL.

Firstly, a variety of tools that allow creating an OCL specification was examined. The experiments with a benchmark UML model with constraints in OCL were carried out. The tools that support code generation from OCL were selected and compared. Among others, facilities of contract inheritance were evaluated. We faced these tools with the T.O.F.I.C. tool [4,10] that supports refinement of UML models towards C# specialized UML models. It also transforms refined models including contracts in OCL into C# code with contracts implemented using the Microsoft Code Contract library.

2. ANALYSIS OF TOOLS SUPPORTING CODE GENERTION FROM OCL

2.1. TOOL SELECTION

In the first step, a set of tools dealing with OCL was reviewed (Tab. 1). Some tools are based on existing components (Dresden OCL or Eclipse MDT OCL), although not entire component functionality was used. The tool origin, external component usage (if any), web site and the main tool features are summarized in Tab. 1. Code generation facilities are not given, as they are further discussed in greater detail.

Only few tools among surveyed in Tab. 1 support generation of a source code based on OCL expressions. Two tools provide this in specialized contexts. One of them is HOL-OCL that generates code for the security lever of an application. It uses an external model repository su4sml, Isabelle/HOL environment and the additional Proof General tool. The main goal is the formal reasoning in the area of OCL constraints associated with UML models. The second specialized application is KMF that supports definition of well-formedness rules at a meta-model level. Modeling tools consistent with a given meta-model are generated. KMF is not used for the generation of a source code for a software system. These both tools will be not discussed further.

For the comparison we selected five tools that can automatically transform contracts specified as OCL expressions in UML models into a source code at a target development platform. The compared tools are listed in the header of Tab. 2. The only commercial solution is Borland Together, which constitutes a complete CASE tool.

Table 1. Tools supporting OCL

Name, version, Company[author|project] [#Used external component] Web address	Basic features
Dresden OCL, 3.1.0, Reuseware (Tech. Univ. Dresden) http://reuseware.org/index.php/DresdenOCL	Comprehensive, widely used library. OCL inv., pre-, post-cond., OCl struct. Correctness rules at UML meta-model, Validation
ArgoUML, 0.30.2, Tigris.org #Dresden OCL http://argouml-downloads.tigris.org/	OCL inv., pre-, post- conditions
MagicDraw UML, 17.0, Magic, Inc. #Dresden OCL http://www.magicdraw.com/	OCL inv., pre-, post-conditions. Constraint syntax inconsistent with the standard
FUJABA, 0.8.0, Padeborn Univ. #Dresden OCL http://www.fujaba.de/projects/fujaba4eclipse.html	Currently moved to Eclipse. Probably OCL update not ready
Incremental OCL, v160906, J. Cabot #Dresden OCL http://jordicabot.com/research/IncrementalOCL/	Incremental compilation and verification of OCL constraints at meta-model of a model
EOS-OCL, 0.4, BM1Software (Tech Univ Dresden) http://www.bm1software.com/eos/	Component for quick calculating of OCL expressions in a UML model
USE, 3.0.0 RC2, Bremen University http://www.db.informatik.uni-bremen.de/projects/USE/	Textual language combining UML and OCL. OCL inv., pre-, post-cond. Validation. System simulation – snapshot verification
HOL-OCL, 0.9.0, A.D. Bruckner, B. Wolff. http://www.brucker.ch/projects/hol-ocl/	Interactive correctness proving of OCL spec. in the context of an UML model
KMF, 1.1.0, Kent University http://www.cs.kent.ac.uk/projects/kmf/	Library for meta-modeling. OCL used to define well-formedness rules. Not updated
OCLE, 2.0.4, Baabes-Bolaai University http://lci.cs.ubbcluj.ro/ocle/index.htm	OCL rules in UML meta-model. OCL inv., pre-, post-conditions in a model. OCL structures. Validation. Not updated
Oclarity, 2.2.0, EmPowerTec AG http://www.empowertec.de/products/oclarity/	OCL inv., pre-, post- conditions. Validation. Semantic correctness
Octopus, 2.2.0, Wermer, . Kleppe (Klassen Objecten) http://octopus.sourceforge.net/	OCL inv., pre- post-conditions at UML model. Not updated
OSLO, V1,Frauenhofer Inst. FOKUS http://oslo-project.berlios.de/	OCL at meta-model. Parsing and verification of const. in model. Probably not updated
Eclipse MDT OCL, 3.0.1, Eclipse http://wiki.eclipse.org/MDT/OCL	Widely used lib. OCL inv., pre-, post-cond. OCL structures. Well-formedness rules at meta-model. A model consistent with MOF. Verification at model and meta-model level
IBM Rational Software Architect, 7.5, IBM #Eclipse MDT OCL/ http://www-142.ibm.com/software/products/	OCL inv., pre-, post- conditions. Do not support all OCL structures from the library

Papyrus UML2 Modeler, Eclipse, #Eclipse MDT OCL/ http://www.papyrusuml.org/	OCL inv., pre-, post-conditions. Do not support all OCL structures from the library
TOPCASED, 4.2.1, TOPCASED #Eclipse MDT OCL/ http://www.topcased.org/	Missing validation of constraints in a model Supports for Meta-model, uses Kermeta
Kermeta, 1.4.0, Triskell #Eclipse MDT OCL/ http://kermeta.org/	Constraints at meta-model Meta-model development (MDE and Aspect Orient. D.)
Borland Together, 2008 R3, Borland #KMF, Eclipse MDT OCL http://techpubs.borland.com/together/	OCL inv., pre-, post- cond. at model. Parsing and verification of constraints
T.O.F.I.C, 1.1.7, P. Ołtarzewski, Warsaw Univ. of Technology #Eclipse MDT OCL http://mion.elka.pw.edu.pl/~poltarze/rsa/update/	Extension for IBM Rational Software Architect. OCL at model and C# profile. OCL inv., pre-, pos- cond. Parsing and verification
SQUAM Framework, 0.8.0, SQUAM (Advanced OCL Editor) http://squam.info/	Extending of OCL meta-model – defining operation library. Lib. used for defining contracts at model. Creating test cases
MIP OCL Parser, 3.1 Multilateral Interoperability Programme https://mipsite.lsec.dnd.ca/Pages/Default.aspxr	Parser for OCL verification and model validation with rules from meta-model. Working on models in Enterprise Architect format
ECO for Visual Studio, 5, CapableObjects http://www.bm1software.com/eos/installation.html	OCL Action Language for queries used in the context of data layer and interlayer interface

2.2. EXPERIMENTAL SETUP

OCL tools were analyzed based on their documentation and delivered examples. However, the main conclusions were drown from experiments concerning code generation from an exemplary UML class model with OCL constraints. An overview of the model that served as a benchmark is given in Fig. 1. Model details, such as attributes, some operations in classes, an enumeration and selected OCL constraints are omitted for the brevity reasons.

Several contracts using different OCL structures were defined in the context of classes *Account* and *Person*, and the *IAddressable* interface. There are four invariants of the *Person* class. The *setAddress* operation of the interface, and the *deposit* operation of *Account* have been specified with pre-, post-conditions. In the constraints, various OCL structures were applied, such as navigation, relations, collection operations, 'let' declarations, '@pre' postfix, etc. The benchmark does not check all possible OCL expressions, but cover the basic OCL features, as well as distinguishing ones for the assessment of the code generation capability.

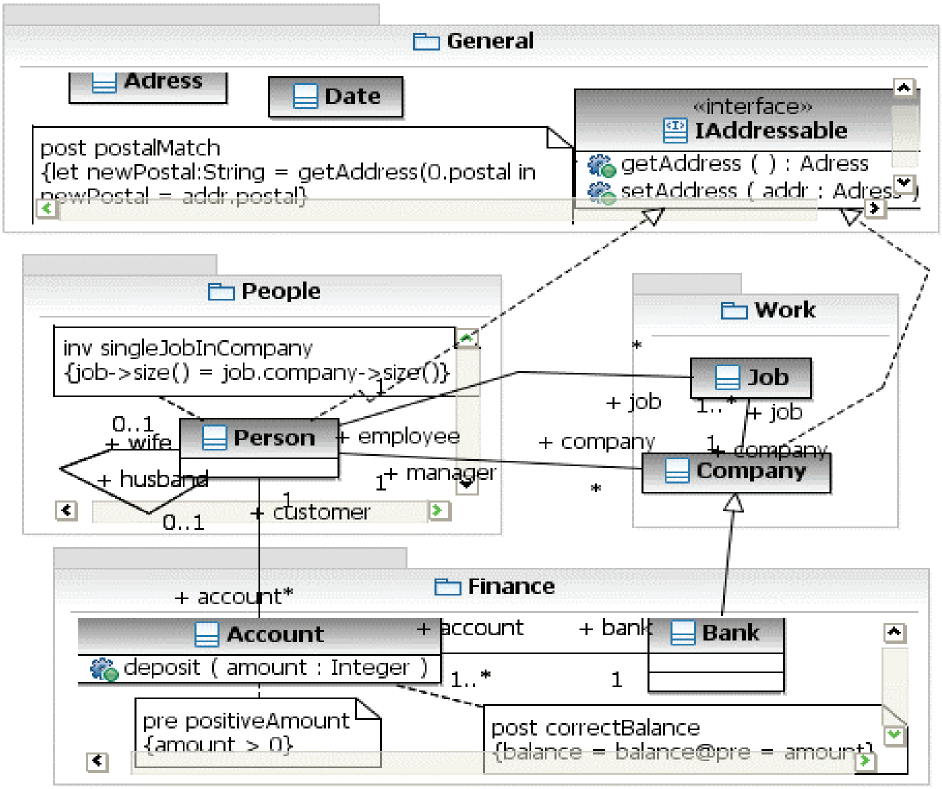


Fig. 1. Exemplary UML model with OCL constraints

Models for experiments were either created using a tool under consideration, as in the case of Borland Together, or built with an external modeling tool. Octopus and OCLE were tested with a model created in ArgoUML and exported to the XMI 1.2 format. DresdenerOCL was tested using a model created in and exported from the IBM Rational Software Architect. The same model was used in tests of T.O.F.I.C.

2.3. TOOL OVERVIEW

In Tab. 2 we summarized the preliminary features of the tools in respect to code generation from OCL contracts. Firstly, versions of supported UML, OCL and XMI standards are given. XMI files are helpful for interchange of models between different environments.

The next row defines the scope of modeling facilities. In all tools OCL constraints can be specified for given UML models ("OCL" description), but only in some tools UML models can be created and modified ("UML").

OCL constraints from which a source code can be generated are summed up in the "OCL generation" row. The following expressions of the OCL language were listed: *pre* – pre-conditions, *post* – post-conditions, *inv* – invariants, *def* – definitions of OCL helper operations for UML type, *init* – attribute initial values defined in OCL, *derive* – derived values of attributes, *body* – body expressions. In columns, characters "+" or "–" denote that a given OCL structure is considered during code generation, accordingly.

For all but one tool a code is also generated from a UML class model that creates the context for the OCL constraints ("UML generation" row). Later, a language of a target source code is given.

In the last row, the support for contract inheritance is compared, although it should be noted that the OCL specification does not define the inheritance rules. Derived classes inherit from contracts defined in a base class or an interface. In both cases inheritance of OCL pre-, post-conditions and invariants was checked. Character "+", "*"or "–" stands for a full, partial or no support for this kind of constraints, accordingly.

Table 2. Comparison of code generation from OCL contracts

	Dresdener OCL	OCLE	Octopus	Borland Together	T.O.F.I.C.
Version of UML	2.3	1.5	1.5	2.1	2.1
Version of OCL	2.2	2.0	2.0	2.0	2.0
Version of XMI	2.1	1.2	1.2	2.1	2.1
Modeling	OCL	OCL, UML	OCL	OCL, UML	OCL, UML
OCL generation: *pre, post, inv, def, init, derive, body*	+, +, +, +, +, +, +	+, +, +, +, +, +, +	+, +, +, +, +, +, +	+, +, -, –, –, –, +	+, +, +, –, –, –, –
UML generation	–	+	+	+	+
Target language	Java, AspectJ, SQL	Java	Java	Java	C#
Contract inheritance class: *pre, post, inv* interface: *pre, post, inv*	*, +, + *, +, +	–, –, –, –, –, –,	–, –, –, –, –, –,	–, –, –, –, –, –,	*, +, + *, +, +

Below, we briefly described how the contracts are implemented by the considered tools. The limitations of implementation of selected OCL constructs are discussed. Especially *@pre* primitive is not easy to be implemented. It can be used in OCL post-conditions to refer to a state of a property (i.e. an attribute value or an association-end specifying a collection) or of a query before an operation execution. We also point out actions that are undertaken when a contract is not satisfied in the target code.

In Dresdener OCL pre-, post-conditions and invariants are transformed into aspects of AspectJ. Using appropriate pointcuts of an aspect, the code of a contract is separated from a type that estblishes the contract context. Application of AspectJ allows a simple realization of contract inheritance, but the rule of weaken of pre-conditions is not satisfied. Primitive *@pre* can be used but its implementation is not fully consistent with the OCL specification. For a not satisfied contract a run-time exception will be generated.

In OCLE class invariants are aggregated in a *ConstraintChecker* class. A programmer is responsible for checking of invariants whether they can be not satisfied. An appropriate text message is output on the console. Pre- and post-conditions of an operation form methods in a local class which is defined in the body of the operation under concern. An instance of the class is created each time the operation is called, in order to check satisfying of the constraints. Contracts are not inherited. Contracts of interfaces, as well as *@pre*, and *null* value are not supported.

In Octopus invariants are transformed into separated methods of a class. Each such method can throw an exception. Invariants are checked in the *checklAllInvarints* method called by a programmer. When an exception is raised, an appropriate message is added to the error list. Pre- and post-conditions are placed in the operations as assertions. They are checked when the method is called. Similarly as for OCLE there is no support for contract inheritance, contracts of interfaces, *@pre* and *null* value.

In Borland Together no code was generated from invariants, probably due to an error in the environment. Code of pre- and post-conditions is built in a special, static, internal class. In an operation for which a contract was defined, assertions are placed before and after the operation body. The assertions call methods that check whether the contract is satisfied. Contracts of interfaces can be generated, but a programmer is responsible for their application in methods that implement the functionality of the interfaces. The *@pre* primitive is supported. A special method archives a state before the execution of a concerned operation; thus this solution is close to the OCL specification.

In T.O.F.I.C. OCL constraints are realized using contracts from the Microsoft Code Contracts library. Invariants are aggregated in one method *(ContractInvariantMethod*) of the class under concern. Run-time verification of invariants is automatically implemented by the library. Pre- and post-conditions are transformed into calls of the corresponding methods of the Code Contracts library. The calls are placed in a body of the context method of the constraints. Contracts of interfaces are supported by the special solution included in the library. For not satisfied constraints, run-time exceptions are generated but they are caught by the library due to its built-in policy. This behavior can be changed, when a programmer implements own class for handling of a contract invalidation during run-time. Contract inheritance for classes and interfaces is assured by the library. But contract inheritance does not allow changing pre-conditions when a method definition is changed. Due to limitations of the library, primitives *@pre* and *allInstances* are not supported.

3. RELATED WORK

A comparative study of tools handling OCL expressions was given in [11]. The tools were compared according to several criteria such as syntactic analysis and type checking of OCL expressions, coupling to UML models, additional facilities including code generation, dynamic validation of invariants and pre-, post-conditions. However, this work surveyed a state of art about ten years ago and at that time the OCL 2.0 version had not been available yet. Among compared tools only one, i.e. OCL Compiler from Dresden, supported code generation from OCL.

A comprehensive OCL benchmark was proposed in [6] in order to assess OCL engine accuracy, determinateness and efficiency. The benchmark was applied to a set of OCL tools, comparing mainly engines for creation and verification of OCL expressions, and do not focusing on the code generation. The authors concluded incompatibilities following from different interpretations of the OCL standard as well as faulty implementations of OCL features. However, the results were not related to particular tools.

There is a recognised need for an efficient OCL tool support covering various features, as stated in [1,3]. The authors also point out at the code generation facilities to different programming languages, but do not consider any particular solution dealing with contracts in the target code. A general overview of the main functionalities of the OCL tools is presented in [3]. Code generation capabilities are evaluated in a binary mode (exist or not), without any in-depth analysis.

Recently, a report of OCL tools based on the IDE4OCL feature model was published [2]. It covers an overview of seven selected tools discussing a wide range of various features and future plans. Therefore, it does not concentrate on more detailed examining the code generation capability. Among considered, the only tool supporting the model transformation feature was again the Dresden OCL.

A list of OCL tools can also be found in the OCL portal [9].

4. FINAL REMARKS

To sum up, the most of tools do not generate code from OCL expresions or transform them to a simple code without using available solutions of code contracts. Constrains are often checked in own, specialized classes. Except of two tools, inheritance of OCL contracts was not realized. Limitations of implementations can also concern contracts of interfaces, as well as post-conditions with the *@pre* primitive. The most comprehensive support of OCL constraints were achieved in the Dresden OCL tool.

Among the compared tools, some distinguishing properties contributed to solutions implemented in T.O.F.I.C. This was the only tool dealing with C# as the target language. In comparison to other tools, it allows precise modeling of a code structure with specialized stereotypes, and a partial verification of a refined model. OCL contracts, in forms of pre-, post-conditions and invariants, can be applied to various structures of the C# language reflected in a model, i.e. classes, interfaces, methods, constructors, operators, properties. The generated code benefits from the application of the contract library delivered in the standard development environment. Therefore, the solution is easy to use, and the contract inheritance of classes and interfaces is automatically implemented. On the other hand, using the library resulted in some drawbacks, such as no weaken in inheritance of pre-conditions and lack of implementation of *@pre*.

A promising approach for implementation of contracts could be utilization of aspect-oriented programming, similarly to AspectJ for Java applied in Dresden OCL. However, the current aspect-based solutions for C# are not powerful enough to satisfy the desired requirements.

REFERENCES

[1] CHIMIAK-OPOKA J., DEMUTH B., *A feature model for an IDE4OCL*, [in:] Proc. Inter. Workshop on OCL and Textual Modeling, 2010.
[2] CHIMIAK-OPOKA J., DEMUTH B. et al., *OCL Tools Report based on the IDE4OCL Feature Model*, [in:] Proc. Inter. Workshop on OCL and Textual Modeling, col. Tools Europe, 2011.
[3] CHIOREAN D., PETRASCU V., PETRASCU D., *How My Favorite Tool Supporting OCL Must Look Like*, In: Proc. of the 8th Inter. Work. on OCL Concepts and Tools (OCL'08) at MoDELS, 2008.
[4] DEREZIŃSKA A., OŁTARZEWSKI P., *Model-Driven Engineering Support for Building C# Applications*, [in:] Innovations in Computing Sciences and Software Engineering, Sobh, T., Elleithy, K. (Eds.), Springer, 2010, 449-454.
[5] Eiffel Software: An Eiffel Tutorial: http://docs.eiffel.com/bookmethod/eiffel-tutorial-et
[6] GOGOLLA M., KULMANN M., BUTTNER F., *A benchmark for OCL engine accuracy, determinateness, and efficiency*, MoDELS 2008, LNCS 5301, Springer, 2008, 446–4589.
[7] MEYER B., Object-Oriented Software Construction, 2nd Edition, Prentice Hall, 1997.
[8] Object Constraint Language (OCL), http://www.omg.org/spec/OCL/
[9] OCL portal http://st.inf.tu-dresden.de/oclportal/
[10] OŁTARZEWSKI P., *Software development using contracts by example of the T.O.F.I.C tool*, Master Thesis, Inst. of Computer Science, Warsaw University of Technology, 2011 (in Polish).
[11] TOVAL A., REQUENA V., FRNANDEZ J.L., *Emerging OCL tools*, Journal of Software and System Modeling, Vol. 2, No. 4, Springer, 2003, 248–261.
[12] Unified Modeling Language (UML), http://www.omg.org/spec/UML/
[13] WARMER J., KLEPPE A., *The Object Constraint Language: Getting Your Models Ready for MDA*, Addison-Wesley Professional, 2008.

software fault injection, hardware fault emulation
script language, scripted fault simulation

Piotr GAWKOWSKI
Konrad GROCHOWSKI*

INSCRIPT – A FAULT INJECTION SCRIPTING LANGUAGE FOR SYSTEM DEPENDABILITY EVALUATION

Fault injection systems are valuable tools for evaluation of target system dependability if the proper set of faults is injected into the system under tests (SUT). Among many advantages, one typical drawback of many fault injection systems is their limited functionality enclosed by the hardcoded fault models or specific target properties. New fault injection scenarios usually require tool modification. The chapter presents the novel scripting language that overcomes this issue, called InScript that is dedicated for software implemented fault injection systems. It allows to flexibly define the desired fault model and the after-injection behavior of the fault injection tool for optimal observation of the fault effects in the SUT. The proposed InScript language was successfully implemented in some fault injection tools and exemplary experiments are reported in the chapter.

1. INTRODUCTION

High dependability is required in many systems used not only in industry but also in many common digital processing devices (e.g. mobile phones, personal computers or multimedia players). So, an important issue is to analyze system susceptibility to faults and identify unsafe situations. For every digital processing system some fault threads exist. These threads have several origins. One of the major sources of faults is indisputably software. That is due to the complexity of implemented functionalities, interfaces, unclearly stated system specification, and, finally, programming errors of different nature within a code. The software engineering techniques and methodologies are targeted to minimize the above mentioned threats. However, it has to be in

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

mind that the software is assumed to be executed on the fully functional hardware. In many cases this assumption cannot be fulfilled as transient soft errors may occur. With the growing complexity of digital devices and power voltages and diameter scaling down, the probability of Single Event Upsets (SEU [1]) in memory elements within digital chips is becoming more and more relevant for system dependability. The system corrupted with SEU may behave unexpectedly, including the critical errors in produced results or safety violated outputs. To fulfill the dependability requirements (not only in terms of safety), a design should be examined with both: the classical software testing (i.e. with software engineering methodologies) and also towards its sensitivity/robustness to erroneous conditions or unstable software modules.

Recently, Software Implemented Fault Injection (SWIFI) gains a wide interest due to the possibility of checking fault susceptibility of the applications in the real system environment [2, 3, 11]. With injecting the proper set of faults into the system under tests (SUT) the one can test SUT's or its components' safety (e.g. observation of fail modes upon user or hardware errors), reliability (fault robustness, system resilience and stability), etc. In our Institute we have developed several flexible SWIFI tools and supporting systems with broad capabilities of testing scenarios and result collection [4–10]. For many years we have performed a lot of experiments with various applications. Moreover, we have developed various software techniques to detect and tolerate faults and compared them with other approaches [5–7, 9]. Having quite rich experience, we found out that among many advantages, one typical drawback of many SWIFI systems is their limited functionality enclosed by the hardcoded fault models or specific target properties. In order to develop a new fault injection scenario (e.g. to cover complex device failure mode), typically it is required to modify tool's source code.

The chapter presents the novel scripting language that overcomes this issue, called InScript that is dedicated for SWIFI systems. It allows to flexibly define the desired fault model and provides statements for manipulation of the target's context (e.g. reading/writing the target's memory and CPU/FPU registers). Moreover, the after-injection behavior of the SWIFI tool can be changed by the script, allowing optimal observation of the fault effects in the SUT (i.e. dedicated procedures registered to be triggered upon given target's events). So, InScript extends not only the flexibility of control over the injected faults but also the observation of the fault effects. A script can be parameterized, so the SWIFI user can tune script parameters in user-friendly dialogs. Thus, the script designer can leave some options to be selected by the experimenter. The proposed InScript language was successfully implemented in SWIFI tools dedicated to evaluation of applications and operating system dependability.

Section 2 presents the basic concept of the SWIFI tools described in relation to our fault injection systems. Section 3 introduces the proposed InScript language. The evaluation of its implementation is reported in Section 4. Then the chapter concludes in Section 5.

2. SOFTWARE IMPLEMENTED FAULT INJECTION CONCEPT

Software Implemented Fault Injectors (SWIFIs) in fact emulate errors – effects of faults. However, a *fault injection* term is well established in literature [2, 3, 11]. *Faults* are *injected* into the real target system (System Under Tests – SUT) during the execution of the workload – usually an Application Under Tests (AUT). Faults are injected through the disturbances to the states of some resources used by the AUT (e.g. changing bits in AUT's code memory, CPU registers). SWIFI is mostly used to emulate temporary faults in the SUT that dominate in modern digital systems (e.g. single/multiple event upsets [1,10]). However, one of the advantages of SWIFI is that it can also be used to emulate very different fault models, like intermittent, permanent, originated in software or hardware, and finally even very complex failure scenarios (e.g. subsystem malfunction). Two main advantages of the SWIFI approach make it very attractive: high controllability over the fault injection (in terms of instant, location, and type) and potentially quite good observability of its propagation and effects. Both are needed to provide a useful feedback to improve SUT/AUT design or implementation.

The SWIFI approach can be realized in different ways. For instance, GOOFI uses a management module with some plugin functionality to cover different targets (e.g. using JTAG interface to inject faults) ([2, 11] and references therein). In [8] we present InBochs fault injector that is based on the modified Bochs system emulator. Taking a look into different solutions and actual fault injection technique, it is clear that the most popular SWIFI implementations are based on the debugger approach [11].

Debugger based SWIFI tool processes some events occurring in the AUT (debugged application). While the fault injector processes an event, the AUT execution is paused in order to allow full access to the AUT context (i.e. CPU registers, memory content, threads management etc.) for the SWIFI system. After event processing (e.g. fault effects emulation) the fault injector resumes (through the debugging API) the AUT execution and waits for further events. Typical events in AUT among others are: loading/unloading a dynamically linked library, beginning of AUT execution, occurrence of software/hardware breakpoint, execution of single machine instruction (if single step mode is set) etc. These events are used by the fault injector to implement the debugging of the AUT that is the basis for fault injection capabilities. Other events are exceptions generated by the AUT as a result of the injected fault (it is very unusual if the application generates exceptions in normal execution), e.g. division by zero, access violation due to incorrect address in read or write operation, stack overflow. They are collected by the fault injector as they provide information upon fault effects and propagation.

Using debugging approach for fault injection has its advantages and drawbacks. As typical debuggers are running on the same machine as the target AUT and are using the system API to implement debugging, the AUT execution is slower as the events need to be processed in sequential manner. On the other hand, using system API guarantees that the scope of the disturbances will be contained directly only within the AUT. That simplifies the experiment setup. Moreover, sometimes it is required to limit the scope of disturbances to selected subset of modules or even selected procedures – that is covered by our fault injectors ([10] and references therein). Debugging API also makes the fault effect emulation easier to implement. Finally, fault effects can be tracked down with very low level of details (e.g. can be mapped into the source code level of the application as presented in [4]) or used for sophisticated analysis (data warehouse based system FEARS [9]). That is a valuable feedback for application developers (e.g. to create fault hardening procedures in the application).

In most of our experiments we use software implemented fault injection system FITS developed in our Institute [10]. Its concept is based on the Debugging API of the Win32 operating systems family. The fault injection experiment is composed of a set of tests (usually many thousands or millions) – each is a single execution of the AUT with a fault disturbance injected at the given triggering instant. FITS uses pseudorandom generator to select fault triggering moments, bit locations for disturbance etc. After the fault injection, the AUT execution is monitored – FITS collects all possible exceptions and messages (AUT can inform FITS that the error was detected and given actions taken with user messages [10]). The injected fault can provoke an unhanded exception within the AUT – in such case the execution is terminated by the operating system and the test is considered as *system exception*. Typically, exceptions relate to memory access violation, invalid or privilege opcode fetch, parity errors, arithmetic or stack overflow, etc. All those information are then logged to the result file and are helpful in further analysis of fault effects or error detection/tolerance effectiveness. Every test has given predefined time limit – in case of exceeding this limit, the test is aborted by FITS and considered as *timeout*. Injected fault can also have no observable impact on the execution of AUT, so, the AUT can terminate by itself in normal fashion. In such case, however, there are two possibilities: the result produced is acceptable and considered as *correct* or is *incorrect*. FITS provides statistics of test results (percentage of *correct*, *incorrect*, *system exception* and *timeout* type results) and on the application profiles.

FITS implements a closed set of fault models – the user has a choice and some configurability setting the fault model's parameters (if any). A new fault model requires to be implemented at the FITS's source code level and FITS to be recompiled. It definitely limits the possibilities of extending its functionality. As some new classes of faults and complex fault injection scenarios are required for different applications or experiment goals, it was important to make the system more flexible. Moreover, some new fault injection tools were designed meanwhile requiring recreation of the already defined fault models. To cover all these issues, the scripting language, called InScript, was proposed, developed and implemented. It is capable of defining the operations to be made by the fault injecting system in order to inject a given fault. Having such possibilities, the new fault models will not require source code changes any more but just preparation of another (reusable) script. Such scripts can be created not only by the fault injector developer but also by any fault injector user or just experiment developers. So, the goal was a new scripting language that would cover all specificities of the fault injection approach that would be intuitive and easy to handle for a user. The interpreter implementation should be effective while keeping the compatibility and ability to easy integration with the already existing and newly created fault injection systems.

3. INSCRIPT LANGUAGE

Thus far experience shows that users expect distinction of the experiment and fault model design. That results in the assumption that the script should still deliver any configuration options handful for experiment designer (e.g. available through the graphical user interface) while the fault model designer should be able to easily define the new fault model as well as define the parameters visible to the experiment designer. Additionally, the script execution overheads should be unnoticeable in terms of the whole experiment execution time – and in comparison with the analogical implementation at the fault injector's source code level.

As a base solution the FITS fault injection system was taken and evaluated into a prototype called SIN (*Scripted INjector*). The usage of the developed scripting language was added to the base functionality. So, the experiment designer can choose among different scripts (implementing different fault models and post injection effect analysis). Fault injection instant is still chosen the same way as it was in FITS. Generally, the one can choose between random distribution of fault injection instants in time domain (more frequent executed code areas are mostly related to the fault triggering instants) or in correlation to the static code image (fault distribution is equally probable in terms of execution of every executed instruction in code segments).

Fig.1 shows the exemplary script that presents the basic statements of the InScript. It implements the simplified fault model that modifies the memory content that is read by the fault triggering instruction of the AUT. As a result, the memory content (e.g. a variable value to be loaded to a CPU register for further operations) is corrupted and that corrupted state is going to be used by the fault triggering instruction (an instruction before the execution of which the fault is injected). The target data will be affected according to some parameters specified by the script user – not only by the script author. So, the implemented script language allows to declare some data as the one to be defined as specific for a given use case. Experiment designer will be asked (through the graphical user interface) after selection of a given script for specification of those parameters (if any). It extends the flexibility of the fault model definition though the scripting language.

```
DESCRIPTION "Exemplary script"
INPUT   /* Input parameters block */
{
  ENUM {
        SET,
        RESET,
        XOR,
        BRIDGE,
       } $type=SET;
  HEX $mask = 0x0;
} // INPUT

MAIN   /* the main block */
{
  // getting the address of the data read by the current instruction
  $address = getDataAddress();
  // getting the memory content from that address
  $value = readMemory($address);
  // value corruption
  switch($type)
  {
  case type_SET:
    $value |= $mask;
    break;
  case type_RESET:
    $value &= (~$mask);
    break;
  case type_XOR:
    $value ^= $mask;
    break;
  case type_BRIDGE:
    $value = bridge($mask, $value);
    break;
  default:
    Log("Unsupported type: " + $type);
    TERMINATE;
  }
  // saving the corrupted value
  writeMemory($address, $value);
} // MAIN
```

Fig. 1. Exemplary script

From the level of the InScript language, the fault model designer has a direct access to the CPU and FPU registers of the AUT as well as to its operating memory, some helping functions (e.g. translation of addresses), and arithmetic/logic operations. Thanks to that, the whole functionality of the original FITS system was quickly reconstructed and some new fault models created with new scripts located in dedicated subdirectory of the fault injector.

The first element of the presented exemplary script is a *description line*, which informs the user about the implemented fault model. This line is used by the fault injector to present the fault model description through the system's GUI.

The second element of the script is an *input block*. It defines some data a value of which can be specified by the experiment designer (not a script designer). In the script shown above, two variables are defined:

- enumerated variable called *type* with the default value equal to *SET* – here, it defines the disturbance type;
- hexadecimal variable called *mask* (32 bits wide variable presented at the GUI in hexadecimal format) – disturbing operation defined by the *type* is about to be applied on the target resource and the *mask* value;

All variables specified in the input block will show up in the graphical user interface as controls adequate to the specified data types (e.g. list of optional values, edit boxes). As shown in the example, the default values can also be specified (e.g. *$type=SET*). Script designer can also add *check blocks* inside parameters definition block, which provides input validation, both for user interface and experiment run.

The main part of the script is the *main block*. It consists of a set of operations to be executed by the fault injector upon reaching the fault triggering instant by the AUT. In example shown above several InScript built-in functions are used that in effect read the AUT's memory content that is going to be read by the fault triggering instruction in the AUT's context. First, the *getDataAddress* function resolves the address of the triggering instruction's source operand (from that address the instruction is about to read). Then the memory content from that address is extracted with *readMemory* function. That content (32 bit wide) is then disturbed with the fault mask according to the disturbance type (specified logical operation is done on that content and the specified mask value). A set of available InScript functions is listed in Tab. 1. As one can see, there is randomization function also available, so, the script can randomize for example the bit positions within the fault mask.

In InScript the variables are preceded with $ sign. However, the whole grammar is based on the C language. There are conditional (*if/else*, *switch*) and loop statements (*while*, *for*, *break/continue*) as well as logical and arithmetic operations – operators and their priorities are the same as in C. Beside the variables declared in the *input block*, the user can also refer to the CPU and FPU registers in the context of the AUT (e.g. to set a new register content: *$EAX=20;*).

The actual functionality of the script is defined in the *code blocks*. Each can be assigned to handle specific events in the AUT (e.g. specific exception handling). However, a single *main block* is required to be defined. It will be executed upon reaching the triggering instant by the AUT. Beside the fault injection (as described above in the example), the script can consist of several *event handling blocks*. They can be used for several proposes:

- to implement long lasting fault (to be active for several CPU machine cycles),
- to distinctively handle different kinds of possible events, like exceptions due to memory access violations or arithmetic exceptions etc.,
- to log the error propagation or execution path after the first fault injection.

Table 1. List of InScript built-in functions

Function name	Arguments:	Description
min	...	returns the minimal value from the set of arguments
max	...	returns the maximal value from the set of arguments
rand	1	returns the random value limited by the argument value
randomMask	1	returns the random value for the fault mask with the number of bits set to 1 equal to the argument value
readMemory	1	returns the 32-bit wide value read from the effective address (argument) within the context of the application under test
writeMemory	2	saves the 32-bit wide value (second arg.) under the effective address (in AUT's memory space) specified as the first arg.
countOnes	1	returns the number of 1 bits in the specified argument
bridge	2	returns a *bridged* value of the first argument with the second argument (all bit positions marked with 1 in the second argument are set to 0 or to 1 based on majority)
getDataAddress	0	returns the address of the data to be read by the current instruction that is going to be executed by the AUT
readFPUStack	1	returns the value read from the FPU stack (in the context of AUT) with the given offset (0÷7)
writeFPUStack	2	saves the value given as a second argument on the AUT's FPU stack at the offset given in the first argument
getBase	1	returns the AUT's base address of the memory segment (used in Windows targets prior Vista to obtain effective address)
testNo	0	returns number of current test (starting from 1)

Each event handling block (including the *main block*) can be dynamically reassigned to handle different events (e.g. single step execution, access violation) with dedicated statements (*CONNECT* and *DISCONNECT*, respectively). Initial *handler blocks* assignments, variable values etc. can be set in optional *init block*, which is executed before every test in the experiment. Additional *Log* statement can be used to write a note into the experiment result file of the fault injector from the script level. That can be used for creation of after-injection fault effects trace (for further analysis).

3.1. IMPLEMENTATION REMARKS

Adding script language to fault simulator introduces some extra overheads. It was crucial for the developed system to minimize those overheads which influence experiment duration. As a result, most of script connected processing is done in experiment preparation time: script code is parsed and interpreted once and stored in memory in form of executable object's composition. Language engine optimizes variable and function access time. Thus, extending the preprocessing time, script language can be optimized for quick execution during tests. Single experiment requires only one preprocessing but it consists of multiple tests (hundreds and thousands of them) and that makes the overhead almost at the unnoticeable level from the user perspective.

Script language engine is implemented in C++ and can easily be merged with different fault injectors. It delegates all of platform specific functionality into extendable interfaces. This way it is independent from fault engine details (like methods of accessing AUT's memory) and allows different platforms to provide some dedicated methods. Graphical user interface generation methods are delegated as well, so script configuration can become an integral part of target fault injector graphical layout.

4. FITS AND SIN COMPARISON

To determine complete time overheads of the newly created system, a set of tests was performed, comparing FITS and SIN experiments' execution times. To make comparison possible, it was required to implement FITS fault models using a new language. This proved the script to be at least as capable as FITS in fault modeling. Tests results can be seen in Table 2. Average time overhead over original implementation is about 1%. This is a reasonable cost for script fault modeling flexibility.

Table 2. SIN and FITS execution time comparison

Fault location	Exp. mean time FITS [hh:mm:ss]	Exp. mean time SIN [hh:mm:ss]	Tests per exp.	Time overhead [%]	Code lines count
Instruction data	03:37:43	03:37:54	3677	0,08%	60
CPU registers	01:02:28	01:03:26	998	1,55%	614
FPU registers	00:42:49	00:43:30	1171	1,62%	265
FPU stack	01:14:19	01:14:51	1171	0,73%	73

Results also show some correlation between script complexity (measured by line count) and execution time. This points out that some further optimization may lower overhead even more. Based on that results some changes are being implemented and tested in script engine, as it is a still developed and enhanced project.

5. CONCLUSIONS

The chapter presents the InScript language – a scripting language dedicated for fault injection systems. It was successfully integrated into two fault injectors: SIN – based on Win32 Debugging API, and InBochs – based on x86 system emulator. In both cases script language proved to be a powerful tool, providing great flexibility to parent systems, with minimal additional overheads. Using the same language in different fault injectors also enabled very easy migration of experiment configurations and fault modeling scripts, thus allowing to perform similar tests on different platforms.

The InScript is a relatively new project and still requires some end user input, but it has already proven to be worth further development and enhancements as it is a valuable inset to SWIFI tools.

REFERENCES

[1] BAUMANN R.C., *Radiation-induced soft errors in advanced semiconductor technologies*, IEEE Trans. Device Mater. Reliab., 5, 2005, 305–316.

[2] BENSO A., PRINETTO P., *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation*, Kluwer Academic Publishers, 2003.

[3] CABODI G., MURCIANO M., VIOLANTE M., *Boosting software fault injection for dependability analysis of real-time embedded applications*, ACM Trans. On Embedded Computing Systems. ACM, Vol. 10, No. 2, December 2010.

[4] DEREZIŃSKA A., SARBA K., GAWKOWSKI P., *Towards the analysis of the line and assembly instruction coverage*, [in:] Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, Vol. 6, 2008, 269–274.

[5] GAWKOWSKI P., ŁAWRYŃCZUK M., MARUSAK P., SOSNOWSKI J., TATJEWSKI P., *Fail-bounded implementations of the numerical model predictive control algorithms*, [in:] Control and Cybernetics, Vol. 39, No. 4, 2010, 1117–1134.

[6] GAWKOWSKI P., SOSNOWSKI J., RADKO B., *Analyzing the effectiveness of fault hardening procedures*, Proc. of the 11th IEEE Int'l On-Line Testing Symp., 2005, 14–19.

[7] GAWKOWSKI P., RUTKOWSKI T., SOSNOWSKI J., *Improving Fault Handling Software Techniques*, Proceedings IEEE 16th International On-Line Testing Symposium (I0LTS), 2010, 197–199.

[8] GAWKOWSKI P., GROCHOWSKI K., ŁAWRYŃCZUK M., MARUSAK P., SOSNOWSKI J., TATJEWSKI P., *Testing Fault Robustness of Model Predictive Control Algorithms*, H. Giese (Ed.): ISARCS 2010, LNCS 6150, Springer-Verlag Berlin Heidelberg, 2010, 109–124.

[9] GAWKOWSKI P., KUCZYŃSKA M. A., KOMOROWSKA A., *Fault Effects Analysis and Reporting System for Dependability Evaluation*, M. Szczuka et al. (Eds.), RSCTC 2010, LNAI 6086, Springer-Verlag, Berlin Heidelberg 2010, 524–533.

[10] MOSDORF M., GROCHOWSKI K., GAWKOWSKI P., SOSNOWSKI J., *Simulating faults in computer systems*, L. Borzemski, A. Grzech, J. Świątek, Z. Wilimowska (Eds.), *Information System Architecture and Technology, New Developments in Web-Age Information Systems*, Oficyna Wydawnicza Politechniki Wrocławskiej, 2010, 337–347.

[11] VOAS J.M., MCGRAW G., *Software fault injection: inoculating programs against errors*, John Wiley & Sons, Inc., New York, NY, USA, 1997.

PART IV

HIGH PERFORMANCE HYBRID ARCHITECTURES

OpenCL development
parallel programming

Jan KWIATKOWSKI*
Maciej PAWŁOWSKI*

TOOLKIT SUPPORTING OPENCL PROGRAMMING

In recent years powerful multi-core processors and programmable graphics processing units (GPUs) have become cheap and ubiquitous. It is common to see commercial gaming consoles or desktop PCs achieve peak single precision Gflop/s ratings in the range of hundreds, if not thousands, by means of employing various parallel architectures. This influx of high-performance yet relatively cheap, readily available compute devices fuels both the consumer market and the scientific supercomputing niche. It is desirable to exploit their high-performance and low unit cost by writing applications for scientific, engineering and commercial use. OpenCL is a framework for developing portable parallel applications for a wide variety of hardware devices (multi-core CPUs, programmable GPUs). Developing in native OpenCL is, however, difficult and requires writing a large amount of code even for simple programs. In the chapter a library that simplify programming in OpenCL is introduced. OpenCL applications can be written with 5–12 times less code by using this library.

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".

At the same time powerful multi-core processors and programmable graphics processing units (GPU's) have become cheap and ubiquitous. It is common to see commercial gaming consoles or desktop PCs achieve peak single precision Gflop/s ratings

* Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

in the range of hundreds, if not thousands, by means of employing various parallel architectures. This influx of high-performance yet relatively cheap, readily available compute devices fuels both the consumer market and the scientific supercomputing niche.

For instance, the IBM Cell processor architecture, implemented in QS20-QS22 Blade compute servers [8] but also in Sony PlayStation 3 gaming consoles, yields a theoretical single precision performance rating between about 180 Gflop/s and 460 Gflop/s, depending on the platform. The architecture has been successfully employed both in a consumer setting and in scientific supercomputing. A relatively novel addition to high performance parallel computing are programmable Graphics Processing Units (GPU). High-end GPUs boast peak single precision performance of over 2.5 Tflop/s per unit (NVIDIA GeForce GTX 590 and Radeon HD 6970, etc.).

OpenCL is a framework for developing portable parallel applications for a wide variety of hardware devices (multi-core CPUs, programmable GPU's). Developing in native OpenCL is, however, difficult and requires writing a large amount of code even for simple programs. In the chapter a library that simplify programming in OpenCL is introduced. The chapter is organized as follow. Section 2 introduces OpenCL, an open standard for programming parallel devices. Section 3 is an overview of CLAW a new library supporting development with OpenCL. Next section shows comparison CLAW against native OpenCL code. Finally, section 5 outlines the work and discusses the further works.

2. DESIGN OVERVIEW – OPEN CL

OpenCL is an open, royalty-free standard for programming parallel devices introduced in 2008 by the Khronos Group – an industry consortium responsible for other such standards, ex. OpenGL for graphics or OpenMAX/AL for audio processing. The objective of OpenCL is to allow writing portable code in a standardized environment. A piece of code written once in OpenCL should, in principle, run on many different kinds of hardware, ex. a multi-core CPU or programmable GPU, if the vendor of such device supplies a valid implementation [1, 2, 7]. This is similar in concept to OpenGL, where graphics-rendering code can be executed on any conformant GPU, as long as an implementation is supplied (usually bundled with the driver).

From the programmer's perspective, OpenCL consists of a language definition and set of header files defining a C interface. The first part, OpenCL C Programming Language, also known as device code is the language in which the actual compute functions (kernels) are written. The language is very similar in syntax to standard C, however it is extended with extra vector basic types (float4, short8, etc.), compound types and special keywords for specifying address space qualifiers (_ _global, _ _local, etc.), access space qualifiers and more. A detailed description of the language itself is beyond the scope of this work, the reader is referred to [4, 5] for further information. The second part of OpenCL visible to the programmer is the Application Programming Interface (API) which she uses to allocate buffers on the devices, copy data between the device and host (GPU and CPU), compile and launch the kernels written in device code and so on. This part can be viewed as a standard C library and is called host code. One of its several functions is to call a vendor-supplied compiler and build a file with OpenCL device code, but the extent of interaction between host and device code domains ends at this point. For all intents and purposes, the code domains can be viewed as separate and independent. The programmer must write kernels in device code and then write an application with host code to compile and use those kernels.

This work will focus on host code exclusively. OpenCL host API offers, by design, a fairly low-level abstraction of the device and requires a substantial amount of code written to perform even basic tasks, like moving a vector of floating point numbers to a GPU device and launching a kernel that will multiply them by two in parallel. While this approach has an advantage of simplicity from the perspective of hardware vendor that attempts to implement the standard, and a possibility to write OpenCL applications for platforms that do not support C++ compilers (ex. embedded devices), it has major disadvantages for programmers who wish to develop OpenCL applications for mainstream platforms (a Windows desktop with a programmable GPU). The following section presents some of the issues that make writing host code difficult.

2.1. COMMON PROBLEMS WITH OPENCL

The first problem is unfriendly C interface. OpenCL presents an interface where key concepts (like a kernel or event) are represented as object handles (actually, integers) and this completes the extend of object encapsulation. Functions are defined in global space and routinely require many parameters. In fact, there are functions which take 12 parameters. The fact that some of those parameters are pointers to type void is another issue. C++ programmers are taught to avoid typeless arrays for several reasons. C++ brings template vectors in the standard library which have a much cleaner semantic and are less prone to programmer errors. However, as OpenCL interface is in pure C, these vectors are not natively supported. Neither are standard strings, instead arrays of char are used through the code, which brings necessity to manage their lifespans, remember to allocate and free memory and so on.

The second problem is related to memory management. All OpenCL objects have pairs of clRetain* and clRelease* functions. So, for an OpenCL context there are clRetainContext(cl_context context) and clReleaseContext(cl_context context) functions. The retain function increments an internal reference counter and release decrements; once the counter reaches 0, the object is freed. This places the burden of managing OpenCL objects lifetimes on the user. The next aspect is managing device memory, that is allocating buffers and copying data. OpenCL defines functions for allocating, deallocating and enqueuing reads and writes. Each device has one or more command queues which receive sequences of commands (to copy data from a host address to a specified OpenCL buffer object) and synchronization points or events (barriers, flushes). The user is tasked with remembering what buffer objects have been allocated and freed, copying data around and synchronizing the commands properly. Additionally copying data between devices which are in different contexts or from different vendors requires manually allocating new buffers on target device, copying existing data to host, then from host to target device, finally freeing the old buffer on previous device. All while making sure no copy command is queued before a preceding copy command has finished. A failure can result in a normal error, undefined behavior or a silent error that results in reading/writing garbage data.

The next problem is causes the way in which errors are handling. Errors are reported in one of two ways - either through a return value or through in-out parameters. This inconsistency alone is a problem. However, even if we ignore this aspect, the code overhead required to check errors returned in this way is considerable. For every OpenCL function call there must be an if statement that compares the error code to CL_SUCCESS and acts accordingly on inequality. It is easy to forget to check an error code. Failure to do so results in unexpected behavior, as the application continues to run as if no errors occurred, yet some OpenCL object may have become corrupted or invalid.

Finally the last problem is related to dependences managing. OpenCL's command queues work in asynchronous mode by default, which allows to schedule a sequence of operations and continue to perform calculations on the host while the device works in background. This however requires that the user tracks dependencies between those operations. Each function that puts a command in a command queue takes an optional array of event objects and itself returns an event. The array of events is the "wait list", the list of events that must have completed before the scheduled command can begin executing. The returned event represents the event of completing the scheduled command. When scheduling a sequence of commands, the user should take events returned by calls to the various enquire functions and insert them into wait lists of subsequent commands. Additionally, when there are several devices and/or command queues in use, external synchronization points are needed. Injecting an event created in one command queue to a different command queue results in an error, the user must be concious of the various queues she uses and know when she needs to use external synchronization methods. Moreover, when the user schedules a copy from device to host memory, she must wait until the copy has finished before attempting to access the data. Otherwise, the data may become stale or corrupted. Similarly, failing to schedule a proper wait event to a kernel after scheduling a write to device may result in the kernel working on incomplete or uninitialized data.

3. LIBRARY SUPPORTING OPENCL DEVELOPMENT

To resolve some of the issues raised in Section 2 a library named CLAW is introduced. CLAW stands for CL API Wrapper and its main purpose is to minimize the amount of host code the developer has to write in order to execute his or hers kernels. CLAW does not affect device code [6].

CLAW is implemented as an object oriented C++ library utilizing modern design principles such as RAII (Resource Acquisition Is Initialization), exception-safety or safe memory management. The objective is to simplify developing OpenCL applications by reducing host code overhead, optimizing some common scenarios and protecting the programmer against typical mistakes.

CLAW consists of several classes which represent OpenCL concepts (Kernel, Device, Platform, etc.), one factory-like class to initialize other resources (Manager) and some auxiliary classes and structures. A Manager object is commonly a starting point in CLAW-enabled parts of the application. All classes which represent OpenCL concepts are derived from the Resource base class (figure 1). The important classes of CLAW are given below.

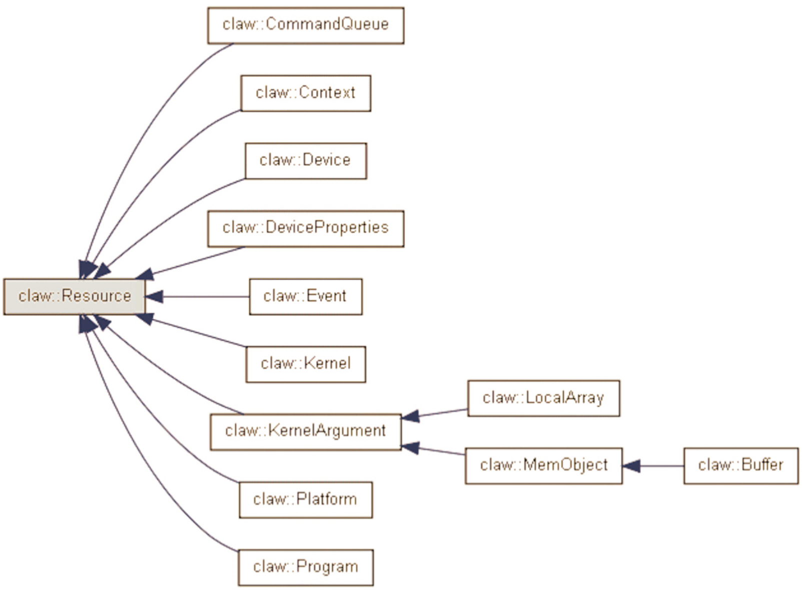


Fig. 1. CLAW resource inheritance diagram

Buffer represents an array of any POD type (float, int, the user's own struct) stored on the device. A Buffer can be accessed on the device as an array. There is no type checking between what kind of data the user uses while setting up a Buffer and what the kernel expects. This would require cross-language type safety (C++/CL) and is not implemented. Note that despite representing a device memory object, a Buffer is strongly connected to host memory and one shouldn't assume a Buffer represents an independent copy of some data. Instead, it must be assumed that the host memory from which the Buffer was initialized is partially shared between the host and device.

CommandQueue represents an OpenCL command queue. The user will rarely use it directly. This object is owned by CLAW internals.

Context represents a context in which one or several Devices from the same Platforms can exist. Wraps around OpenCL cl_context handle. This object is owned by CLAW internals.

Device represents a single OpenCL-capable device accessible on the system. It may be a CPU, a GPU or any other device. This object is owned by CLAW internals.

Event allows waiting for asynchronous calls, represents a synchronization point. Asynchronous methods, like Kernel::launch() or MemObject::retreiveData() with the parameter blocking set to false, return Event object pointers, through which the user can wait. Events can also be injected into Kernels' wait queues with Kernel::pushWaitEvent() to ensure they wait for the particular events to complete before launching. Since OpenCL 1.1, user events can be created. A user event can also be used as a synchronization point but the user may manually set its executionStatus once. Since OpenCL 1.1, event callbacks can be registered. This object is owned by the user.

Kernel represents an OpenCL kernel. A Kernel is extracted from a Program object and can be launched on various Devices if the associated Program was built for them. Kernels wait for and produce Event objects that represent dependencies.

KernelArguments represents an argument passed to a Kernel call. Basic Kernel Arguments can be used directly for POD items and vectors of thereof but doing so will make the kernel execution blocking.

LocalArray represents a local array argument to the kernel that dynamically allocates local memory on device per work group. Use this when the OpenCL kernel expects a pointer to _ _local variable.

Manager initializes Platforms, Contexts and Devices, required in any CLAW program. Manager is the starting point for all CLAW functionality. All Resources depend on the Manager. Manager is also a factory that allows creating Kernels, Programs and Buffers.

MemObject represents a memory object that can be instantiated on a Device, copied between Devices and filled with data. Additionally, it implements dependency tracking and synchronization logic. However, creating the underlying OpenCL memobject and enqueueing writes/reads is left to extenders. This allows extenders to exploit MemObject's behaviour but supply various OpenCL data types, like a buffer or image. A MemObject is not tied to a Device, Context or Platform. Its logic allow for moving the data between Devices, even across Platforms.

Platform represents a vendor-supplied platform that may support several devices. A Platform creates and stores Contexts.

Program represents a collection of Kernels build from common source. An OpenCL program may contain one or more kernel definitions and a number of static read-only variables. As OpenCL does not specify a linker, the entire program must be built from a single source, often a *.cl* file. All kernels defined within this file can be extracted from the program and called by the host. Device functions can be called by kernels and other device functions only, not by the host, and they are not exposed through Program objects. A Program must be loaded and then built. This is dome automatically by the Manager's creation methods. This object is owned by the user.

Resources are a base class for all CLAW resources, like Kernels, Buffers or Platforms. All Resource are non-copyable, have facilities to log errors and messages via the Manager, need a Manager to live and have a readable name (may be empty) and typename

3.1. INTERNAL HOST MEMORY MANAGEMENT

CLAW allocates all Resources with the standard new() operator. Internally, dynamically allocated objects are always tracked by shared ptrs and standard vectors or strings are used instead of dynamically allocated arrays. The library is built in accordance with the RAII idiom. CLAW is very strict about not managing raw pointers lifetimes. Wrapping all allocations in shared ptrs, avoiding circular dependencies and properly defining ownerships greatly reduces the possibility of introducing memory leaks, double deletions and other memory related problems. Users are encouraged, but not forced, to follow RAII principles in their own applications using CLAW. The user should not attempt to create Resources with new(), the standard way of obtaining Resource objects is by calling appropriate creation methods. It's illegal to use a Resource after the destruction of its Manager. A Resource assumes its Manager is available for at least as long as the Resource lives. Attempting to artificially extend a Resource's life beyond the Manager's lifetime may result in unexpected errors.

3.2.ERROR HANDLING AND NOTIFICATION

CLAW uses a ClawException class to represent runtime exceptions. ClawException derives publically from std::exception. In addition to the standard what() method, which contains a human-readable error description, it supplies a getErrorCode() method, which returns a cl_int with the OpenCL error code, if applicable. Note that some errors are not results of failed OpenCL API calls and thus do not produce error codes. In such case, getErrorCode() will return CL_SUCCESS. When getErrorCode() returns CL_SUCCESS, the error was likely not raised by the OpenCL runtime, but by internal CLAW logic. CLAW is internally exception-safe in that no memory is ever leaked if an exception is thrown within a method. An exception thrown from within a constructor means the object became invalid and the user should not attempt to use it. In the current version, this behavior is recursive, so when, for example, a Platform's constructor creates a Context, and the Context's constructor throws, Platform should also be considered corrupted. This is likely to change in future versions. Exceptions thrown from other member functions do not corrupt the object.

3.3 DEPENDENCY MANAGEMENT

Since OpenCL kernels are, by default, launched asynchronously and potentially concurrently on several devices, a dependency tracking mechanism is required. OpenCL supplies a command queue/event based solution which requires the user to either manually specify which kernel launches should wait for which kernel completions or to insert barriers. CLAW automatically tracks all flow dependencies. This includes dependencies of copying data between host and device and launching subsequent kernels. Assuming the following sequence:

- Assign hostA memory to deviceBufferA buffer,
- Launch kernelOne(deviceBufferA),
- Launch kernelTwo(deviceBufferA),
- Copy data from deviceBufferA back to hostA.

CLAW will automatically allocate deviceBufferA on whichever device the kernel is being run on and begin copying contents of hostA to deviceBufferA, not allow kernelOne to launch before copying from hostA to deviceBufferA has finished, not allow kernelTwo to launch before kernelOne has finished – even if those kernels are launched on different devices from different platforms, not allow to start copying deviceBufferA back to hostA until kernelTwo finishes, block until data is finished copying to hostA before returning control to the application

CLAW uses lazy memory allocation copying. Memory will only be allocated and copied between host and device when needed Dependencies are properly tracked even across many devices, contexts and platforms. Currently, all tracked memory objects are treated as read-write.

4. AN EXAMPLE OF USING CLAW

Vector addition is a simple parallel algorithm: each element of input vector A is added to a corresponding element of input vector B and the result is stored in an element of output vector C. OpenCL kernel of NVIDIA's vector addition sample is shown below:

```
//OpenCL Kernel Function for element by element vector addition
_ _kernel void VectorAdd (_ _global const float* a,
    _ _global const float* b, _ _global float* c ,int iNumEle-
ments)
{   // get index into global data array
    int iGID = get_global_id(0);
    //bound check (equivalent to the limit on a 'for' loop for
    //standard/serial C code
    if ID >= iNumElements)
    {  return  };
    //add the vector element s
    c[iGID] = a [iGID] + b [iGID];
}
```

Fig. 2. Vector addition kernel from NVIDIA GPU Computing SDK v.3.0

The host code required to launch this kernel in CLAW is given in full below:

```
#include <iostream>
#include "claw.h"
using namespace claw;
int main (int argc, char* argv [ ])
{  try {
        int n = 10;
        std::vector<float> a(n), b(n), c(n);
        for (int i = 0; i < n; i++)
          {a[i] =1 ; b[i] = 2; c[i] = 0; }
        Manager m;
        KernelSP kernel =
            m.createKernel("VectorAdd", "VectorAdd.cl",
                            "-cl-fast-relaxed-math");
        kernel->setGlobalWorkSize(n);
        kernel->launch(a, b, c, n);
        for (int i = 0; i < n; i++)
        { std::cout << c[i] <<" ";}
   {catch (ClawException e)
      {std::cout << e.what( ); }
   return 0; }
```

Fig. 3. Vector addition host code in CLAW

When code that initializes data and prints out the results is removed, the entire application can be written with 20 lines. Example code from NVIDIA is too long to be shown here. The compacted version, with no comments, data initialization code and cross-checking but with oclLoadProgSource moved inline, takes 195 lines. This constitutes a roughly 10x improvement in code size. Interestingly, a significant fraction of the native code bloat was due to using C structures, malloc() and free(), handling strings through char arrays and so on. Only around 50 lines were tied directly to the OpenCL API. Many more were devoted to handling errors, processing strings, etc.

5. CONCLUSIONS

Example programs written with CLAW require significantly less code. The code overhead is noticeably smaller even when comparing to the official C++ bindings bundled with the OpenCL headers, although the difference is less striking. The differences are visible not only in the number of lines of code. Programs are also easier to write and maintain with CLAW due to automatic memory and dependency management and simpler interface. This effect is difficult to quantify and thus not measured, however it should be visible to programmers by comparing versions. A possible way to measure program complexity (as opposed to code verbosity/length) would be to count the number of explicit memory allocations and deallocations or if statements. Such measurements may be carried out in the future.

ACKNOWLEDGEMENTS

The research presented in this chapter has been done under international project "Large-scale calculations for heterogeneous architectures as a part of the COST IC0805.

REFERENCES

[1] CHAPMAN B., JOST G., van der PAS R., *Using OpenMP – Portable Shared Memory Parallel Programming*, The M.I.T Press, 2007.
[2] GSCHWIND M., HOFSTEE M.P., FLACHS B., HOPKINS M., WATANABE Y., YAMAZAKI T., *Synergistic processing in cell's multicore architecture*, IEEE Micro, 2006.
[3] IBM Redbooks, *Programming the Cell Broadband Engine Architecture: Examples and Best Practices*, Vervante, 2008.
[4] MUNSHI A., ed., *The OpenCL Specification*, Khronos Group, 2010.
[5] NVIDIA. NVIDIA OpenCL Best Practices Guide, 10 2009.
[6] PAWLOWSKI M., *Toolkit supporting OpenCl programming*, Master Thesis, Wrocław University of Technology, Wrocław 2011.
[7] THOMADAKIS M.E., *The architecture of the nehalem processor and nehalem-ep smp platforms*, Technical report, Texas A&M Supercomputer Facility, 2010.
[8] VOGT J.-S., LAND R., BOETTIGER H., KRNJAJIC Z., BAIER H. *Ibm bladecenter qs22: Design, performance, and utilization in hybrid computing systems*, IBM Journal of Research and Development, 53, 2009.

parallel processing, load balancing
Cell B.E. processor, logic-algebraic method

Mariusz FRAŚ*
Aleksander WALERIAŃCZYK*

EFFECTIVE PARALLEL COMPUTATIONS OF LOGIC-ALGEBRAIC ALGORITHMS ON CELL B.E. ARCHITECTURE

In order to perform parallel computations effectively, many issues must be taken under consideration. One of the most important is an architecture of computation system itself. The chapter concerns design of implementation of parallel algorithms in computation system based on multicore processor Cell B.E. In this section the problem of processing of parallel algorithms for logic-algebraic method on Cell B.E. processor is exploited. The specific structure of processor and its communication mechanisms impose a careful design of processing architecture. Presented solutions can be regarded as guidelines for design of parallel processing of any task with a similar pattern of computation. Proposed architecture was implemented in real system and some results of measurements of processing efficiency is finally presented.

1. INTRODUCTION

The Cell B.E. processor is a heterogeneous multi-core processor compatible with the Cell Broadband Engine Architecture. The unit expands the 64-bit PowerPC architecture. It was originally planned to be used mainly to handle media in the consumer electronics, such as game consoles, but over time it became clear that it is willing to be used also in the scientific community to perform intensive computing.

For a class of control, identification and recognition systems an object expert system with knowledge representation in the form of logical facts $F(\alpha)$ composed of sequences α of simple formulae (properties describing the object under consideration), and a logic-algebraic (LA) method are developed [3, 4]. The LA calculations are an example of difficult computational problem for Cell B.E. based systems.

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

For different problems the LA method defines the scheme of computations that is based on determining at a number of computational stages j (defined by partitioning of the set of facts on several subsets) an appropriate subsets $\overline{S}_{j-1}$ of sequences of logical values of the simple formulae $\overline{a}_{j-1}$ according to the formula:

$$\overline{S}_{j-1} = \left\{ \overline{a}_{j-1} \in S_{j-1} : \bigvee_{\overline{a}_j \in \overline{S}_j} \left[\overline{F}_j(\overline{a}_{j-1}, \overline{a}_j) = 1 \right] \right\} \tag{1}$$

where: $j = J, J-1, \ldots, 1$, J – the number of computational stages, $\overline{F}_j$ – subsequence of some facts (conjunction) containing appropriate subsequences $\overline{a}_{j-1}, \overline{a}_j$ of logical variables of values of simple formulae; S_{j-1} – the set of all possible sequences $\overline{a}_{j-1}$, $\overline{S}_J = S_J$, $\overline{a}_0$ – input subsequence of variables, $\overline{S}_0$ – the ultimate solution for the expert problem formulated on the basis of LA approach.

On each j-th stage the sequences $\overline{a}_{j-1}$ are computed for processing on the next stage. The LA computations can be performed in parallel [1]. To improve the speed of computations calculations can be organized by definition of so called basic task $Z_{j,i}(\overline{a}_{j-1}^{(i)}, \overline{a}_j)$ which is defined as follows: for specified $\overline{a}_{j-1}^{(i)} \in S_{j-1}$ the condition $\bigvee_{\overline{a}_j \in \overline{S}_j} \left[\overline{F}_j(\overline{a}_{j-1}^{(i)}, \overline{a}_j) = 1 \right]$ is tested for successive $\overline{a}_j \in \overline{S}_j$, and calculations terminate when the condition is true. Basic task is defined by specified sequence $\overline{a}_{j-1}^{(i)}$ which value (0-1 sequence) may be regarded as basic task identifier.

This scheme of computations has many advantages (among the others reduces the search space) but also represents difficult pattern of computations for parallel processing environment. Basic tasks have different completion time and data are processed in following manner:

- a part of input data (i.e. $\overline{a}_{j-1}$) can be processed independently,
- a part of input data (i.e. $\overline{a}_j$) are processed repeatedly (for every $\overline{a}_{j-1}^{(i)}$).

In case of large tasks (i.e. long sequences of $\overline{a}_j$) and small processors' cache it may lead to significant communication overhead caused by data transfer to/from external (main) memory. The system can be also unbalanced. Under such conditions the proper design of calculation must be done. The Cell B.E. based system is an example of the system where such problems exists. Next sections describe characteristics of Cell B.E. processor and presents proper design for mentioned scheme of computations.

2. CELL BE ARCHITECTURE

2.1. PROCESSOR ARCHITECTURE

The Cell B.E. processor [5, 6] has unique heterogeneous architecture (fig. 1). It consists of eight auxiliary SPE cores (Synergistic Processor Element), specialized in performing calculations of distinct character, and one main PPE core (PowerPC Processor Element). The division into different, functionally specialized, cores inside a single processor is purposeful. The main motivation for the separation of cores inside the chip was the increase in processing performance, which was achieved through a special design of the SPE cores, so that they are able to effectively execute compute-intensive tasks, and to dispense with this feature the main PPE core. In turn, the PPE core has the ability to quickly switch tasks, but does not perform computing as efficiently. These characteristics determine the specific way of building applications using the described unit.

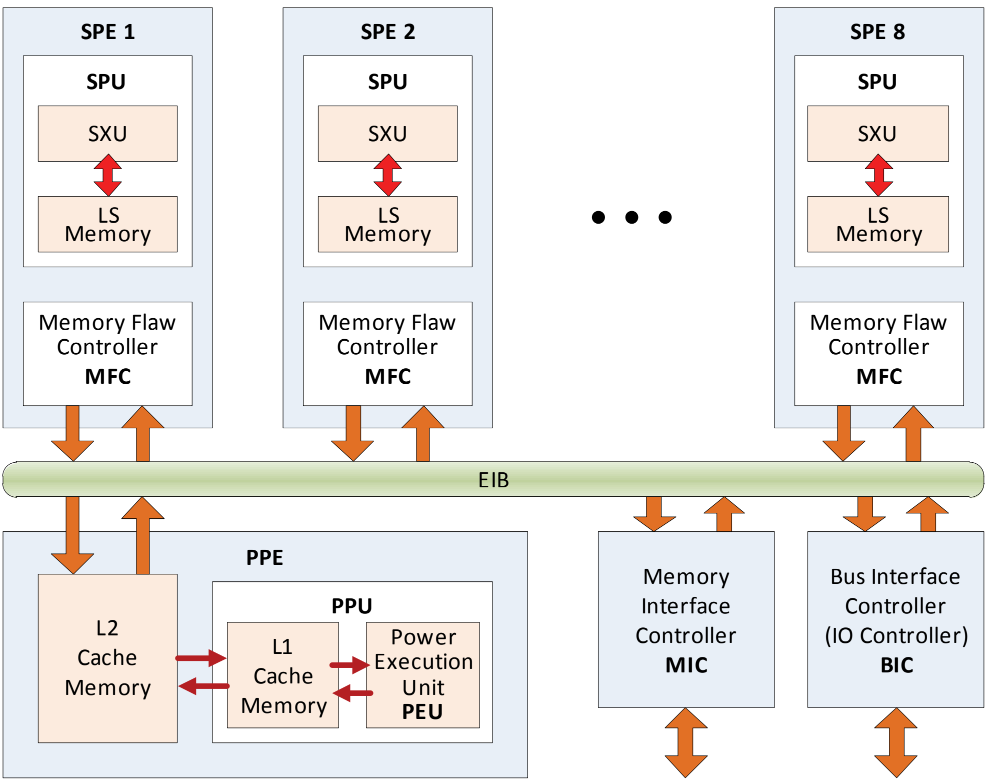


Fig. 1. Cell/BE processor architecture – main functional units

The SPE are independent processing units that allow to run individual programs or threads. Significant difference between the PPE and SPE cores is the way to access memory. Despite the fact that all cores have access to main memory, it should be emphasized that the PPE core performing the load and store instructions (causing the movement of data between main memory and registers) refers to the memory directly. In contrast, the SPE cores in order to move data between main memory and the registers must perform the DMA transfers. Load and store instructions executed by the SPE cores can refer only to the local storage LS.

PowerPC Processor Element is the main general purpose core of Cell B.E. processor. It is characterized by the possibility of both 32 and 64 bit processing and a reduced instruction set (RISC). PPE often performs overall control of the processing of the entire system. It is specialized to perform operating system code.

PPE core is composed of two major units: the PowerPC Processor Unit (PPU) and the memory subsystem PPSS (PowerPC Processor Storage Subsystem). In addition to storing data PPSS deals with maintaining consistency between memories. It is responsible for all memory access operations for both units SPU and PPU. PPSS subsystem fully supports SMP instructions. All elements (including general-purpose registers) are doubled, with the exception of those responsible for system resources. Through these features PPE may be viewed as a dual-multiprocessor.

Each SPU unit (Synergistic Processing Unit) is an independent processor with its own counter, optimized to perform computational programs. SPE cores (SXU – Synergic eXecution Unit) operate directly on their local memory LS (Local Store) with a capacity of 256 KB only. LS memory store both data and executable code for the SPU. There is a lack of any cache in SPU, what has a significant consequences in the design of parallel processing on Cell B.E. SPE performs data movement with the load and store instructions between their registers (called register file) and the LS memory only. However, LS memory can be loaded with required data using the DMA requests directed to a MFC (Memory Flow Controller), which supports data transfer between local memory and main memory.

2.2 COMMUNICATION ISSUES

The core processor communication subsystem is Element Interconnect Bus (EIB). EIB combines CPU internal systems and forms communication paths for commands and data sent by all the processor cores, memory controllers and I/O controllers. EIB provides consistent access to shared memory.

EIB consists of four 16-byte width communication rings. Each ring can perform up to three transfers simultaneously. The theoretical maximum bandwidth of the EIB is 2048 GB/s at clock frequency of 1.6 GHz. Communication subsystem theoretically supports simultaneous transfer for all cores of about 25 GB/s.

Each SPU core is equipped with his own MFC controller that serves as the interface connecting it through the EIB bus to main memory. The MFC basic function is to serve direct memory access transfer requests. In addition, the controller maintains consistency between main memory and the LS, and supports the communication inside the system that uses mailbox mechanisms and signals.

To perform transfers between memories efficiently, MFC is equipped with a DMA controller (DMAC). MFC has the ability to autonomously perform the DMA commands, which allows immediate continuation of the processing by the SPU core – in parallel with the ongoing DMA transfers. DMA transfer execution imposes a number of requirements on the organization of data transfer including: restrictions on the size of data transferred and the restrictions on alignment of data in source and destination memory. This all has a significant influence on the architecture of designed parallel processing applications.

An important feature of the Cell B.E. processor is the possibility of mapping the contents of LS memory to a specific area of main memory. The result is a very important benefit that allows to communicate the SPE cores directly with each other. The process of SPE-SPE communication is done by issuing by SPE cores write data commands to LS memory areas mapped to the areas of main memory. Recognizing the addresses of the transfer request between two LS memories, MFC controller in some measure changes the properties of executed transfer. This results in the transfer of the entire portion of transferred data (using the EIB bus), to the destination LS memory, with no phase of writing data to main memory. Next, the mechanisms that care about the consistency of memory are activated. Such behavior results in an almost instant appearance of the transferred data in the target LS memory. Having the addresses of such mapped areas, the SPE cores may perform transfers (using DMA) between their LS memories very effectively.

3. PARALLEL PROCESSING ARCHITECTURE

Effective processing of any task in parallel processing environment requires the proper design of three elements:

- development of a method solving the problem useful in a parallel environment,
- development of parallel computing algorithm,
- design of specific processing mechanisms appropriate for architecture of processing environment (Cell architecture in this case).

The method used for processing LA task is the one presented in introduction. The two proposed parallel computing algorithms are: the basic task allocation algorithm R [1, 2] and dynamic task allocation algorithm RD.

3.1. PARALLEL ALGORITHMS

The basic allocation strategy (algorithm R) is to assign to each processor the equal number of basic tasks $Z_{j,i}$ to be processed at given stage of computations by partitioning the sorted sequence of task identifiers w_i of basic tasks $Z_{j,i}$ on Q continuous subsequences [1, 2]:

$$\widetilde{Z}_{j,p} = \left\{ Z_{j,i} : w_i \in \left\langle w'_p, w''_p \right\rangle \right\} \tag{2}$$

where: $\widetilde{Z}_{j,p}$ – tasks executed on processor p at j-th stage of computations (the set of basic tasks $Z_{j,i}$), $p = 1, \ldots, Q$, w_i – identifier of basic task $Z_{j,i}$, w'_p, w''_p – boundary task identifiers for processor p, $w'_p = \left\lceil (p-1) \cdot \frac{\beta_j}{Q} \right\rceil$, $w''_p = \left\lceil p \cdot \frac{\beta_j}{Q} \right\rceil - 1$, β_j – the number of all subtasks to be processed at stage j, Q – the number of processors.

All processors compute their own set of data without communication with each other. This allocation strategy is very efficient and minimizes communication overhead, however does not guarantee balanced processors' load [1].

The second algorithm, dynamic task allocation algorithm RD, is proposed to balance workload of processors during the progress of computations. The idea of the algorithm is to extend algorithm R by an additional procedure of tasks migration according to the following rules:

- the initial partitioning of tasks is the same as for algorithm R,
- when any processor finishes his portion of computations it communicates with the others, and checks its progress of computations,
- all processors periodically checks if any processor requests the information, and if yes they suspend computations and send back the number of non-computed task,
- the idle processor chooses the processor with the biggest number of tasks to compute and takes a half tasks from the chosen processor,
- the idle processor signals all processors to continue calculation.

The sequential processing of basic tasks according to their identifiers permits to perform tasks migration with minimal communication cost. When given processor (SPE 1 in example in figure 2) becomes idle it signals the others to stop computations and modifies the range of tasks to process on one chosen processor (modifies one identifier of next task to process). The migration process overhead is affected only by synchronization of reading and writing minimal amount of data.

3.2. DESIGN OF DATA MANAGEMENT

The unique design of parallel processing apart from specific nature of algorithm, usually also refers to inter-processor communication mechanism. Here, the migration process and specific features of basic tasks must be taken under consideration.

The efficient tasks migration can be assured by fast writing data directly between processors (here SPUs) using the following schema used in implemented application:

- running by PPE core computational process in so called SPE context (mechanism available in Cell B.E. processors – SPE context is schedulable for running on SPE core) with LS memory mapping option,
- during context initialization obtaining by PPE core MS (Main System) memory addresses that are mapped as storage area for LS memory of every SPE core,
- distribution of obtained addresses to all SPE cores,

Using LS memory mapping, knowing all addresses, and using established a common format of processing control data (especially variables representing tasks identifiers including current/next basic task to compute) the tasks migration is achieved by simple read/write operation from/to proper variables as shown in figure 2. Using such a scheme tasks migration procedure is very efficient.

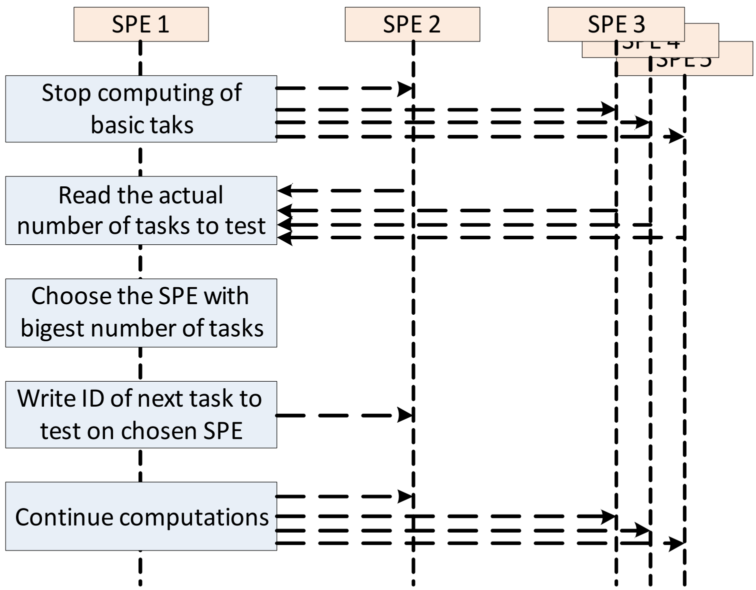


Fig. 2. Tasks migration procedure

The calculation of basic task relies on multiple testing of the value of fact for different input data. The scheme of data processing is shown in figure 3.

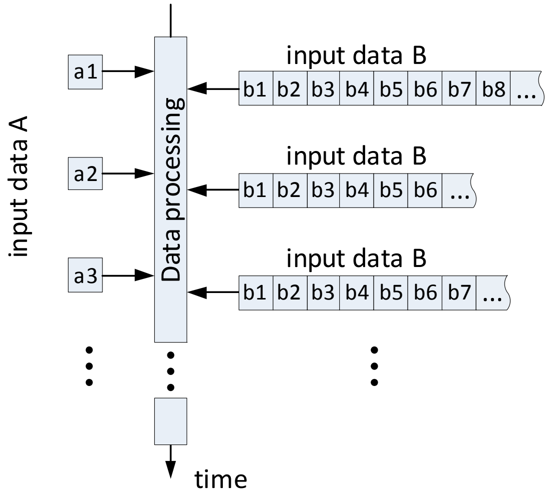


Fig. 3. The scheme of data processing for basic task

For subsequent portions of input data A calculations are performed repeatedly for subsequent portions of input data B. The problem arises when the amount of data B is large compared to the amount of processor memory. Then processing must include data transfer from external memory what may decrease the speed of computations. This is the case for SPE cores and LA computations.

In order to decrease the impact of multiple data transfers on speed of computations we propose modified double data buffering with background data transfer. The idea of data processing is shown in figure 4. First, Data Buffer 1 is loaded with data from external memory (MS memory in our case). This buffer maintained until the end of calculations. Next, data from Buffer 1 are processed while Data Buffer 2 is loaded. Finally, data from buffers 2 and 3 are alternately processed and loaded with subsequent input data B in parallel, for processing for all portions of input data A (i.e. basic tasks in our case). The procedure is explicit by below code:

```
block_no = 1;
load(block_no,1);
while(exist_basic_tasks_to_process)
{
  do in parallel
  {
    calculate(1);
    load(block_no+1,2);
  }
  while(not_stop_condition)
```

```
    {
      block_no = block_no+1;
      do in parallel
      {
        calculate(block_no mod 2) + 2);
        load(block_no+1,(block_no+1) mod 2) + 2);
      }
    }
}
```

The instruction **load**(block_no,*x*) means loading buffer *x* with external block data with number block_no, **calculate**(*x*) means LA calculation on data in buffer *x*, *stop_condition* is stop condition for calculating given basic task.

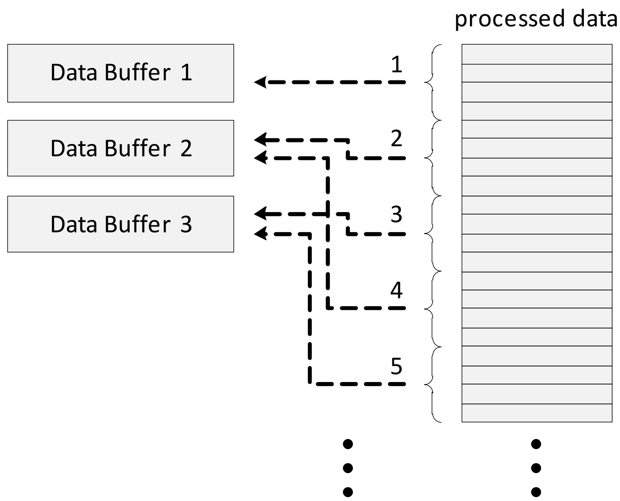


Fig. 4. Modified double buffering of processed data

Since data transfer is performed by MFC controller in the background, such organization of processing makes very low overhead for data transfer.

4. COMPUTATIONAL EXPERIMENTS

The presented design of parallel processing of LA algorithms was implemented in real Cell B.E. based system – IBM BladeCenter QS21. The effectiveness of proposed solutions was evaluated for randomly generated LA tasks producing workload according to specified number of facts *LF*, number of simple formulae *LFE*, number of computational stages *J*, and several parameters that define probability of occurrence of logical operators in facts. The experiments were performed for R and RD algorithms. This allowed to evaluate the effectiveness of procedure of tasks migration and the effectiveness of entire procedure, including the buffering mechanism. The presented results of experiments are for the tasks of the following profile: *LF* = 50, *LFE* = 78, *J* = 5. The average number of basic tasks to compute on every stage was about 8 thousand.

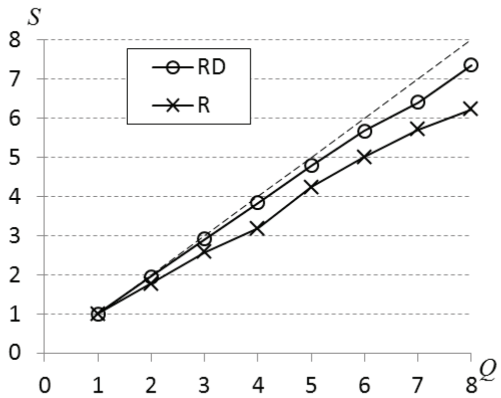


Fig. 5. The speedup of computations *S* versus the number of SPU cores *Q*

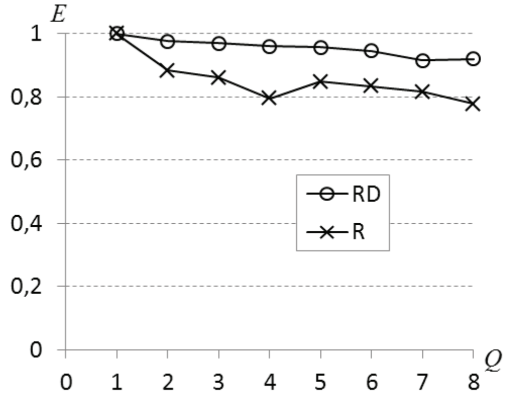


Fig. 6. The effectiveness *E* versus the number of SPU cores *Q*

The figure 5 shows the speedup of computations *S* versus the number of SPE cores for algorithm R and RD (the ideal speedup is also shown by a dotted line). The figure 6 shows the effectiveness *E* versus the number of SPE cores for both algorithms. The results show that tasks migration procedure is effective for tested number of processors, and RD algorithm works very well. The proposed modified double buffering is a mechanism that mot only reduces non-calculation periods but also apparently seems not to entail congestion of data transfers.

5. CONCLUSIONS

Presented work show that logic-algebraic algorithms can be successfully applied in real multiprocessor environment. The specific architecture of Cell B.E. processor requires, as any real computation system, suitable design of processing architecture. The proposed solutions for computations in Cell B.E. based systems apply not only for logic-algebraic algorithms but also for any problem that involves load balancing and the pattern of data processing is such that requires multiple repetitive calculations on large data, that must be loaded many times from main memory.

In work [1] simulation experiments performed for the systems with theoretical architecture of inter-processor communication showed that effectiveness of algorithm RD distinctly decreases for larger number of processing units. The presented scheme of processing for Cell B.E. processor doesn't show such decrease. However it should be verified in the aggregated system with larger number of SPU cores.

ACKNOWLEDGEMENTS

The research presented in this chapter has been done under international project "Large-scale calculations for heterogeneous architectures" as a part of the COST IC0805.

REFERENCES

[1] BORZEMSKI L., FRAŚ M., *Execution of Parallel Logic-Algebraic Algorithms in Expert System SYMLOG*, Journal of Applied Computer Science, Vol. 10, No. 1, 2002.
[2] BORZEMSKI L., FRAŚ M., *Parallel computations for logic-algebraic based expert systems*, Lecture Notes on Artificial Intelligence, Vol. 4692, 2007.
[3] BUBNICKI Z., *Logic-Algebraic Method for Knowledge Based Control Systems*, Modelling, Identification and Control, Acta Press, Zurich 1995.
[4] BUBNICKI Z., *Logic-algebraic method for a class of knowledge based systems*, [in:] Pichler F., Moreno-Diaz R. (Eds.), *Computer Aided Systems Theory*, LNCS, Vol. 1333. Springer-Verlag, Berlin 1997.
[5] IBM Corp., Sony C. E. Inc., Toshiba Corp., *Cell Broadband Engine Architecture*, NY, 2007.
[6] IBM Redbooks. *Programming the Cell Broadband Engine Architecture: Examples and Best Practices*, Vervante 2008.

complex instruction set computing, reduced instruction set computing
floating point arithmetic, Web-based decision support systems

Michał DANIELAK*, Grzegorz KOTOWSKI*,
Leszek BORZEMSKI*

COMPARISON OF CISC AND RISC ARCHITECTURES' PERFORMANCE IN WEB-BASED DECISION SUPPORT SYSTEMS

This chapter contains the comparison of CISC and RISC architectures in Web-based decision support systems. Not only does its first part elucidates those two architectures, but it also illuminates operating systems that can be used on the latter. The second part of this chapter introduces four computational programs solving problems that may be found in above-mentioned systems: the aggregation of autonomous systems, the prediction of a time needed to download resources, elementary arithmetic operations and sorting algorithms. The final part embodies obtained results and introduces an absolute quality indicator.

1. INTRODUCTION

The rivalry between the architecture of reduced instruction set computing (RISC) and its predecessor – complex instruction set computing (CISC) – lasts continuously since the eighties. Currently, those two technologies significantly differ from those of 80s and a thin boundary that used to divide them commenced to gradually disappear. What is more, proven solutions of the former have appeared in the latter and vice versa.

According to [9], all published comparisons of RISC and CISC architectures are classified into two categories:

- quantitative comparison, namely those that compare program sizes and execution speeds,
- qualitative comparison, which examine issues of high level language support and use of very large scale integration – the process of creating integrated circuits by combining thousands of transistors into a single chip.

---

* Institute of Informatics, Wroclaw University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, e-mail: {michal.danielak, grzegorz.kotowski, leszek.borzemski}@pwr.wroc.pl

This chapter is a quantitative comparison whose main task is to suggest the most sufficient architecture for efficient data processing in Web-based decision support systems, especially those which enable efficient retrieval of data from the Internet by monitoring network traffic and predicting its load [6], [7]. Accordingly, this part evaluates computers' architectures by means of specific computational problems that usually are not solved in such architectures' comparisons.

According to [9], every comparison of RISC and CISC is associated with the following problems:

- pair of RISC and CISC machines that is directly comparable in terms of: levels of technology, complexity of compilers or even mere costs does not exist, and
- no definitive set of test programs that unambiguously evaluates the performance of examined machines occurs.

Nevertheless, the latter problem is inessential, because this chapter merely presents the set of evaluative programs in order to help in selecting the appropriate architecture for the described earlier Web-based system.

Later in this section, the general characteristics of CISC and RISC architectures and the brief description of operating systems such as IBM AIX and Red Hat Enterprise Linux appear. Section 2 not only contains the general description of experiments and research environment, but it also introduces four computational problems and synthetic quality indicator. Subsequent sections of this chapter discuss obtained results and comprise the final remarks.

1.1. CISC AND RISC

According to Steve Brazier and Hrvoje Cekolj's definition, the term 'CISC' (complex instruction set computer or computing) refers to computers designed with a full most frequently used instructions, the computer would get more tasks done in a shorter

Table 1.1. Short comparison of RISC and CISC architectures [2]

CISC	RISC
Primary goal is to complete a task in as few lines of assembly as possible	Primary goal is to speedup individual instruction
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" Incorporated in instructions	Register to register: "LOAD" and "STORE" are Independent instructions
Small code size	Large code sizes
variable length Instructions	Equal length instructions which make pipelining possible

set of computer instructions, intended to provide needed capabilities in the most efficient way. However, it was later discovered that, by reducing the full set to only the amount of time. Since this was called reduced instruction set computing (RISC), there was now a need to have something to call full-set instruction computers – thus, the term CISC. Table 1.1 presents the short comparison of those two architectures.

1.2. IBM AIX AND RED HAT ENTERPRISE LINUX

While a few years ago Linux systems, including Red Hat Enterprise Linux (RHEL), were very popular and omnipresent, operating system developed and sold by IBM – IBM Advanced Interactive eXecutive (AIX) were not; however, even though AIX can be used only on specialized equipment such as POWER architecture, it has evolved significantly in the past few years. Currently, AIX can still only be used on dedicated hardware, but IBM's product line has grown considerably to such an extent that it is supported by diversity of servers: from blades through small rack servers (2U size) up to the huge servers that can fill practically all the space in a computer rack.

RHEL operating system is a commercial Linux distribution created by Red Hat, Inc. Not only can this system successfully run on computers with IA-32, IA-64, POWER processors, but it is often used as an operating system for many embedded devices, mobile PCs and servers. Detailed comparison of the functionality of AIX and RHEL can be found in [3], [4].

2. EXPERIMENTS

2.1. GENERAL DESCRIPTION OF EXPERIMENTS

The further part of this chapter describes the evaluating researches of RISC and CISC architectures. Tests were conducted in order to choose a proper computer architecture for Web-based decision support systems, especially those described in [6] and [7]. To this end, this part distinguishes the most important tasks, whose execution times had a significant impact on Web-based systems' performance. First, it was crucial to solve the combinatorial autonomous systems' (AS) aggregation problem [6]. Second, calculating definite integrals was necessary to predict (by means of aggregated data) the times needed to obtain resources from the certain source [7]. Finally, performing basic arithmetic operations, namely sorting the sets of numbers or matrix multiplication was essential.

2.2. RESEARCH ENVIRONMENT

All experiments presented in this chapter were conducted in the Department of Distributed Computer Systems at Wroclaw University of Technology.

The tests were conducted on three servers equipped with RISC processors, marked successively as: JS22-1-AIX, JS22-1-RHEL, RHEL-JS22 and one CISC-based processor IBM Blade HS21 server, marked as: HS21-CentOS. The detailed hardware description of above-mentioned servers is shown in Table 2.1

Table 2.1. Hardware configuration of the evaluated machines

Component \ Server	JS22-1-AIX	JS22-1-RHEL	JS22-2-RHEL	HS21-CentOS
Processor (CPU)	4.0 GHz POWER6	4.0 GHz POWER6	4.0 GHz POWER6	3.0 GHZ CoreQuad
Number of CPUs	2	2	2	2
Number of cores in CPU	2	2	2	4
CPU type	64-bitowy	64-bitowy	64-bitowy	64-bitowy
Memory	4GB	4GB	4GB	16GB
Internal memory disk	73,4GB SAS 10 000 rpm	73,4GB SAS 10 000 rpm	n/a	73,4 SAS 10 000 rpm

The configuration of server JS22-2-RHEL differs from the one of JS22-1-RHEL only in terms of lack of internal disk storage. Hence, operating system installed on the latter server is run by means of the local network from the external disk array. This configuration allowed to examine whether the lack of internal disk storage and thus the need to install an operating system on the external disk array, affects the efficiency with numerical calculations in Web-based decision supporting systems.

2.3. COMPUTATIONAL PROBLEMS

As noted at the outset, experiments were conducted from the point of architecture selection, which was to be recommended for Web-based decision support systems, especially concerning those described in [6] and [7]. First computational problem (CP1) was taken directly from [6]. It consists in the aggregation of autonomous systems (AS) into the larger groups – Meta-ASes – that have similar characteristics. This NP-hard problem has been formulated as a set minimizing task. The algorithm used in this chapter solves this task by means of generating all possible solutions and then choosing the one that contains the smallest number of subsets.

Monte Carlo Method (MC) is used for mathematical modeling of complex processes, such as evaluating multidimensional definite integrals with complicated boundary conditions or stochastic optimization. Using the MC method is justified in cases in which the rate of acquiring the result is more important than its accuracy [5]. The in-depth description of MC can be found in [8].

The second computational problem (CP2) uses equation (1) to obtain definite integral of the function (2) on the interval $\lfloor x_p, x_k \rfloor$:

$$\int_{x_p}^{x_k} f(x)dx \approx \lambda \frac{\sum_{i=1}^{n} f(x_{random})}{n} \tag{1}$$

$$f(x) \approx \sqrt{x} + 3x \tag{2}$$

The subsequent computational problem (CP3) is a classical matrix multiplication, called Cauchy's multiplication. This chapter copes with CP3 by means of a naive algorithm with the $O(n^3)$ complexity. For this reason, OpenMP – a multi-platform application programming interface for multiprocessor systems with shared memory – was used for the sake of CP3. For more information on OpenMP, see [1].

The final computational problem (CP4) consist in sorting η-number floating-point number set by means of radix sort algorithm (RSA). RSA in most cases is used for sorting integers and its computational complexity equals $O(n \log n)$. However, given the bit representation of floating point number, this chapters implements modification of RSA proposed in [10] and treat it as if it were a fixed-point number.

2.4. SYNTHETIC INDICATOR OF QUALITY

This section introduces following notation:

$t_{i,j}$ – the average time of three tests for conducting the *j*-th computational problem on *i*-th server,

- w_j – weighting factor for the *i*-th computational problem,
- J – the number of computational problems,
- K_i – *i*-th computer, wherein:
  - K_1 – JS22-1-AIX,
  - K_2 – JS22-1-RHEL,
  - K_3 – JS22-2-RHEL,
  - K_4 – HS21-CentOS.

Table 2.2 shows the adopted weighting factors for different computational problems. In the previously described web-based system, aggregation will be performed only once, therefore the weighting factor for CP1 is $w_1 = 0.2$. The weighting factor for CP2 is $w_2 = 0.3$, because integrals calculations are necessary in predicting the times needed to obtain resources; the weighting factor for CP3 is $w_3 = 0.4$, because this computational problem copes with examining the times of performing basic arithmetical operations. Finally, taking into account that sorting algorithms will be executed very rarely the weighting factor for the last computational problem is $w_4 = 0.1$.

Table 2.2. Adopted weighting factors for individual programs

Problem	Problem's symbol	Problem's weight
Minimum set cover problem	CP1	$w_1 = 0.2$
Monte Carlo integration	CP2	$w_2 = 0.3$
Matrix multiplication	CP3	$w_3 = 0.4$
Radix sort algorithm	CP4	$w_4 = 0.1$

The quality indicators are calculated for individual instances of problems that are characterized in Table 2.3. For instance, the time $t_{i,3}$ means the time of multiplying two matrices with dimensions $N \times N = 4\,000 \times 4\,000$ on the i-th computer.

Table 2.3. Characteristics of assumptions for computational problems, in order to calculate quality indicators

Problem	Assumptions
CP1	the number of ASes under aggregation $card\,|A| = 22$
CP2	the interval of integration: $[x_p, x_k] = [0, 2010]$
CP3	the size of a matrix: $N = 4\,000$
CP4	the number of elements of a sorted set: $\eta = 40\,000\,000$

Let

$$Q^* = Q_1 = \sum_{j=1}^{J} w_j t_{1,j} \tag{3}$$

be the quality indicator for the reference server K_I.

Substituting the results into the formula (3) one can calculate the value of the reference index:

$$Q^* = \sum_{j=1}^{N} w_j t_{1,j} = 41\,685.394$$

The absolute indicator of quality for the *i*-th PC can be calculated as:

$$Q_i = \sum_{j=1}^{N} w_j t_{i,j} - Q^* \tag{4}$$

The use of a quality indicator Q_i lets you answer the question about how many seconds considered computer – K_i – is better or worse than the reference one – K_1 – in solving computational problems proposed in this chapter. Attention focuses on two cases:

– $Q_i > 0$, which means that K_i is Q_i seconds inferior to the reference server,

– $Q_i < 0$, which means that K_i is Q_i seconds surpasses the reference server.

For the reference server K_1 $Q_{i=1} = 0$.

3. RESULTS

Figure 3.1 shows the relationship between CP1's solution time and the number of ASes subjected to aggregation. For the last measurement (wherein the number of Ases = 22), it took virtually 147 days for JS22-AIX server to generate the desired MetaASes, whereas its CISC counterpart – HS21 – coped with this problem in barely 126 days. One should realise that the difference is immense and evaluation survey speaks for CISC solutions even for so small a size of the problem

Notwithstanding obtained results, it should be mentioned that the implementation of the algorithm for the solution of CP1 used only fixed-point operations, hence present in the POWER6 processor floating-point accelerator is not used at all.

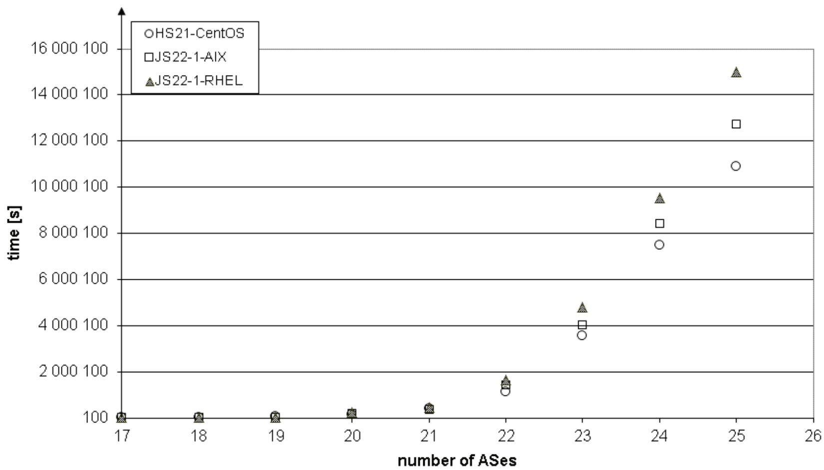


Fig. 3.1. The relationship between the time of CP1's calculations and the number of AS

Figure 3.2 shows the relationship between time of CP2's calculations and the number of haphazard points for evaluated RISC and CISC servers. Better results were obtained for the CISC members. One should be aware that computer program, solving CP2 is performed in a single thread, because dividing it into multi-thread application, and then synchronizing all the threads is so difficult a task that the results obtained by such an application are worse than those of a single-threaded.

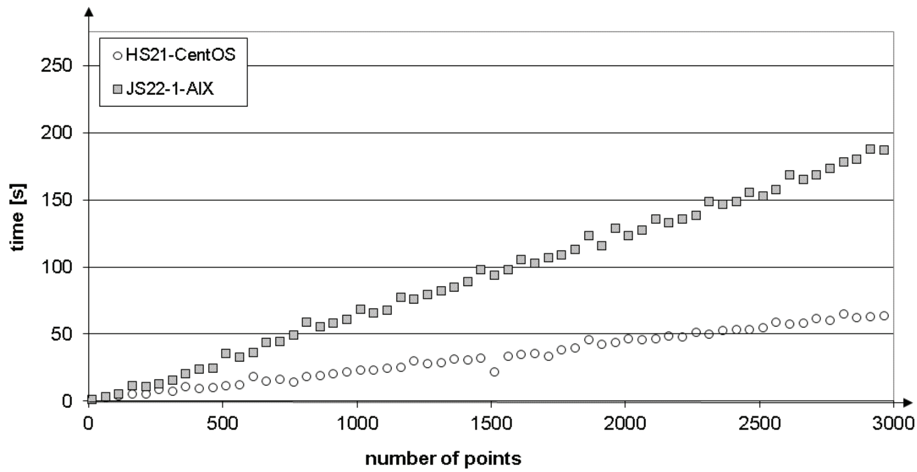


Fig. 3.2. The relationship between the time of CP2's calculations and the number of haphazard points for the selected CISC and RISC servers

However, the figure illustrating the relationship between execution time and the number of haphazard CP2 points for the evaluated RISC servers was not presented due to negligible differences in obtained results.

Figure 3.3 shows the relationship between the time needed to perform two square matrix multiplication and their size for selected servers. This experiment was conducted for the gradually increasing sizes of matrixes (N). One should note that choice of the operating was essential in CP3's calculations. Strictly speaking, calculations conducted on the same server are performed faster on AIX than on RHEL. Furthermore, due to the use of floating-point numbers, RISC computers needed less time to obtain the correct results. Once again, the JS22-1-RHEL managed worse than JS22-1-AIX.

Figure 3.4 shows the relationship between the RSA sorting time for various floating-point number sets. Although floating-point numbers were used in this computational problem, it appeared that due to the use of the aforementioned RSA's modification – treating floating-point numbers as if they were fixed-point – the program resulted in lack of the use of the accelerator for the floating-point operations. It appeared, therefore, that the mere use of floating point-numbers in the program is not enough to observe the significant growth in the rate of calculations (as in CP3).

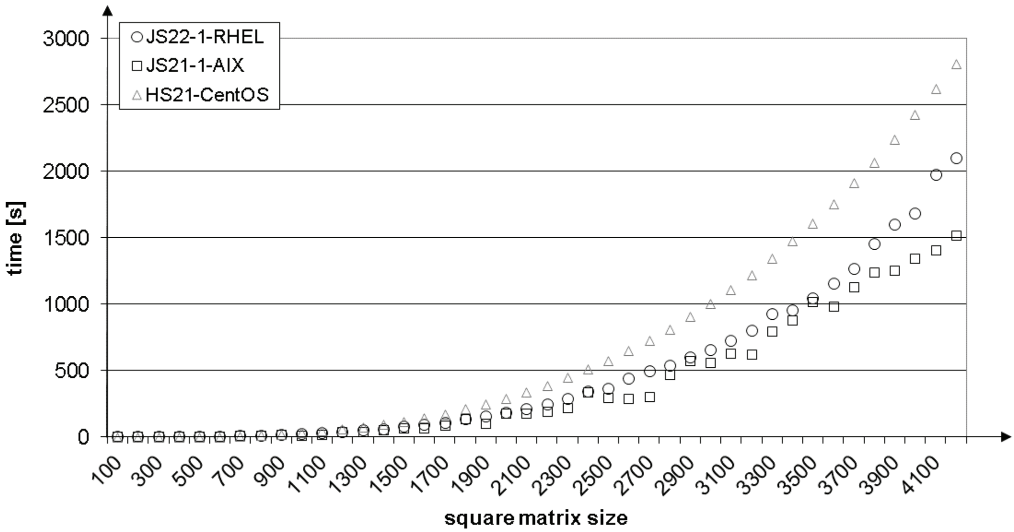


Fig. 3.3. The relationship between the time of matrix multiplication and their size for the evaluated servers

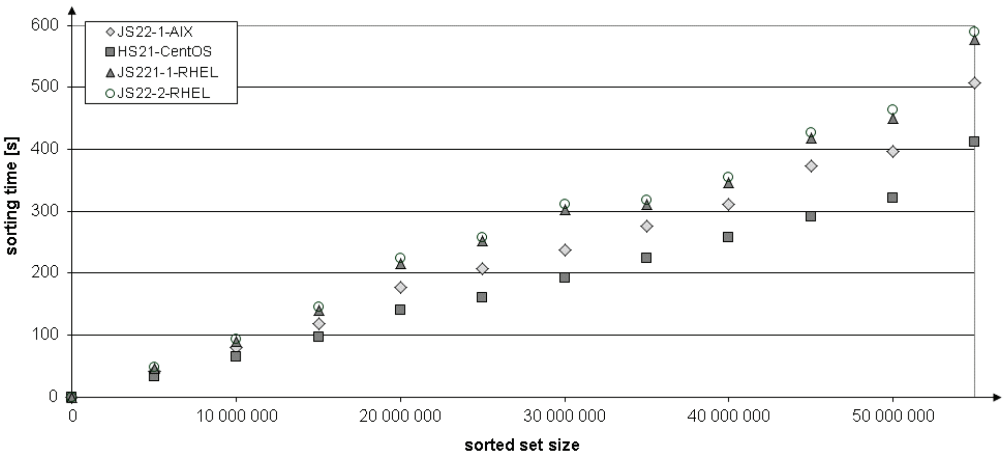


Fig. 3.4. The relationship between the RSA sorting time for various floating point number sets

Table 3.1 presents the times of computational problems solving (calculated in accordance with the assumptions presented in Table 2.3) for every evaluated machines. By using the formula (3), the absolute quality indicator was calculated for every server and then presented in the last column.

Table 3.1. Times of solving particular computational problem and quality indicators for every tested server

$t_{i,j}$ / K_i	$t_{i,1}$ [s]	$t_{i,2}$ [s]	$t_{i,3}$ [s]	$t_{i,4}$ [s]	Q_i [s]
K_1	123.586	310.174	205,402.645	1,341.930	0.000
K_2	129.803	346.414	242,470.245	1,685.060	7,556.261
K_3	136.000	355.069	247,332.481	1,739.189	8,553.084
K_4	47.074	256.440	247,332.481	2,429.940	–9,239.227

Based on the results presented in Table 3.1, it can be concluded that computer K4 obtained the most efficient results. It also turned out that the mere choice of an operating system for RISC computers is important, namely JS22-1-AIX server can solve all computational problems by more than two and a half hours (9 239.227 seconds) faster than JS22-1-RHEL, and the reason is probably that, unlike Linux, AIX was designed and optimized solely for the RISC architecture.

Less significant was the lack of internal disk storage, and thus the need to install the operating system on an external disk array. The differences in performance between the server without internal storage and the one with it were often negligible.

4. FINAL REMARKS

Based on the obtained results, it can be stated that in creating the previously mentioned Web-based system, the hybrid solution is proposed:

– CISC architecture in order to perform the aggregation of ASes (CP1),
– RISC architecture in order to predict times needed to download resources (CP2, CP3 and CP4).

Although creating the hybrid Web-based system is possible (CP1 is performed independently of the others computational problems), it is very expensive. Therefore, in the case wherein the creator of such a system would be forced to choose only one architecture, he or she should opt for the CISC one, for which the absolute quality indicator was about 2.5 hour better than that of RISC.

The further research in this area may concern the qualitative comparison of computer clusters based on CISC and RISC processors for Web-based decision supporting systems.

ACKNOWLEDGEMENTS

The research presented in this chapter has been done under international project *Large-scale calculations for heterogeneous architectures* as a part of the COST IC0805.

REFERENCES

[1] CHAPMAN B., JOST G., van der PAS R., *Using Open MP: Portable Shared Memory Parallel Programming*, 2007.

[2] CHEN C., NOVICK G., SHIMANO K., *The Intellectual Excitement of Computer Science: RISC architecture, http://www-cs-faculty.stanford.edu/~eroberts/courses/soco/projects/2000-01/risc/*

[3] GALVIN P., Pete's all things Sun: comparing Solaris to Red Hat Enterprise and AIX, 2010.

[4] GALVIN P., Pete's all things Sun: comparing Solaris to Red Hat Enterprise and AIX – Virtualization Features, 2011.

[5] GOULD H., TOBOCHNIK J., CHRISTIAN W., *Numerical Integration and Monte Carlo Methods*, 2001.

[6] KOTOWSKI G., GĄSIOR D., *Application of Non-Probabilistic Descriptions of Uncertainty for Evaluation of Approximate Performance Parameters in Autonomous Systems Aggregation*, [in:] Information Systems Architecture and Technology Advances in Web-Age Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, 2009, 59–63.

[7] KOTOWSKI G., GĄSIOR D., *Adaptive fuzzy predictor for Web server resource obtaining time anticipation*, [in:] Information Systems Architecture and Technology New developments in Web-Age Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, 2010, 77–87.

[8] *Metoda Monte Carlo*, http://pl.wikipedia.org/wiki/Metoda_Monte_Carlo.

[9] STALLINGS W., *Computer Organization and Architecture: Designing for Performance*, Pretinc Hall Inc., a Simon Schuster Company, 2009.

[10] TERDIMAN P., *Radix sort revisited*, 2000.

computer system survivability, risk analysis, security, stochastic activity network

Eimantas GARŠVA, Nerijus PAULAUSKAS,
Lech GULBINOVIČ, Arvydas STANKEVIČIUS*

COMPUTER SYSTEM SURVIVABILITY EVALUATION BASED ON RISK ANALYSIS

Computer system survivability simulation model was composed and computer system survivability simulation was performed in this research. Model parameters were taken from legal regulation and risk analysis. Requirements to the system recovery time and accessibility are set by regulation, based on the computer system category. The simulation was done by using stochastic activity networks. Simulation results show that the modelled computer system security mostly depends on the incident occurrence probability, on the strength of protection mechanisms, while the occurring incident severity has the least effect on the protected computer system.

1. INTRODUCTION

The knowledge about the system and threats occurring to it is vital, in order to evaluate computer systems security. This information can be revealed by the risk analysis. Survivability is a common, numeric characteristic of system ability to survive the incident. It is used for system comparison, and security mechanism evaluation. During this research computer system survivability simulation model was composed. Model parameters were taken from computer system security regulation and risk analysis. Modelling results are presented and analysed. The model is different from the previous model [1, 2, 3], because of a slightly different computer system concept.

Computer system security regulation is discussed in next section. Model parameters taken from risk analysis are described in section 3. Composed computer system survivability simulation model is provided in section 4. Section 5 analyses computer system survivability characteristics. Results are presented in section 6. In Section 7, we offer concluding remarks and share directions of future work.

---

* Vilnius Gediminas Technical University.

2. COMPUTER SYSTEM SECURITY REGULATION

Government information systems are best regulated and will be addressed in this research. Information systems are categorised based on its vitality to the state [4]. Then, the requirements to the system recovery time and accessibility are set [5]. There are four different categories. The requirements for 1st and 2nd system categories are very high and system recovery time should not be longer when 15 min for first category and one hour for second category. The information accessibility is set as 99% for 1st and 96% for 2nd system category. The requirements for 3rd and 4th system categories are set only for working hours and working days. We chose to model the system which is qualified as to be in third category, because most governmental systems are managed by different institutions in Lithuania and they are not directly regulated by the law. Third category information system must be recovered in 8 hours and must be accessible 90% at working hours. Security mechanisms are described for each category. Adapting these regulations to the modelled environment we consider that there are 36 different mechanisms, one distinct mechanism can be used to protect from one or more threats: 28 mechanisms protect from the threats to confidentiality (C), 26 – to integrity (I) and 21 – to availability (A).

The regulation sets the information system complexity requirements: 4th category must have 2 or more information system subsystems (modules), 3rd category must have no less than 3 modules, 2nd category no less than 5 and 1st category no less than 7 modules. We made an assumption that modelled information system despite being in third category is complex and has 5 modules m.

3. USAGE OF RISK ANALYSIS INFORMATION

Risk analysis in this research addresses one aspect of all information security – the computer network risks rising from the outer perimeter of the computer system. During the risk analysis assets of the hypothetical third category computer system were identified, threats were outlined and implemented security mechanisms revised. There is different amount of security mechanisms implemented to protect different modules in the modelled computer system.

Risk analysis showed module compromise detection interval Δt_d, the importance of different modules described as module weight $w(m)$ and the rate of different computer system module usage, so module usage probabilities $P_M(m)$ were determined. Then the probabilities of incidents targeted to confidentiality ($P_{Cm}(j)$), integrity ($P_{Im}(j)$) and availability ($P_{Am}(j)$) for different modules according to incident severities $P_m(j)$ were determined. Modelled computer system characteristics revealed by the risk analysis are shown in the Table 1.

Risk analysis showed that particular computer system has not all protection mechanisms in place, 0.917 of all security mechanisms required by the third category information systems were in place. Because of possibility for one security mechanism to protect from more than one threat, the number of security mechanisms protecting the modelled system N_a is this: $N_{a\mathrm{C}} = 25.4$, $N_{a\mathrm{I}} = 23.8$, $N_{a\mathrm{A}} = 19.6$.

Table 1. Modelled computer system characteristics

	m														
	1			2			3			4			5		
$P_M(m)$	0.4			0.2			0.2			0.1			0.1		
Δt_d, h	0.1			0.5			1			2			4		
$w(m)$	0.5			0.3			0.1			0.05			0.05		
	$P_m(j)$														
j	C	I	A	C	I	A	C	I	A	C	I	A	C	I	A
3	.04	.04	.05	.04	.04	.05	.04	.04	.05	.18	.15	.10	.18	.15	.10
2	.05	.05	.06	.05	.05	.06	.05	.05	.06	.23	.20	.12	.23	.20	.12
1	.06	.06	.08	.06	.06	.08	.06	.06	.08	.29	.25	.15	.29	.25	.15

Note: .04 = 0.04.

4. SIMULATION MODEL

The computer system is a distributed computer network with boundaries defined, which is facing the computer incident i after the time interval $\Delta t_{inc,i}$ and withstands it or one or more modules of the system are compromised on time $t_{c,i}$. The degradation of the system is detected after some time $\Delta t_{d,i}$ and then the system state is restored after the interval of time $\Delta t_{r,i}$. Computer system is modelled during the time interval Δt_{all}, which is long enough for all the events to appear (Fig. 1).

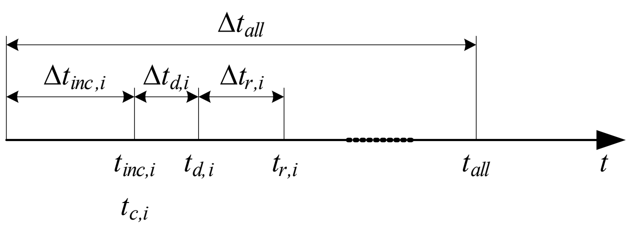


Fig. 1. Computer system security events

Incident occurrence in this model is a stochastic process, not a genetic algorithm which is often used to model malware propagation [6, 7]. Incidents are grouped according to the threat and can have different severity levels j, first one is most severe $j = 1$. Incidents occur by Poisson law [8, 9] and target the specific module m of the modelled computer system considering its rate of use. All the system modules are protected by the security mechanisms N_a defined by the risk analysis. Computer system modules have different importance which is represented by its weigh $w(m)$. Structure of computer system model is presented in Fig. 2. Model parameter values were determined by risk analysis or set by regulation; corresponding parts are marked on the models structure.

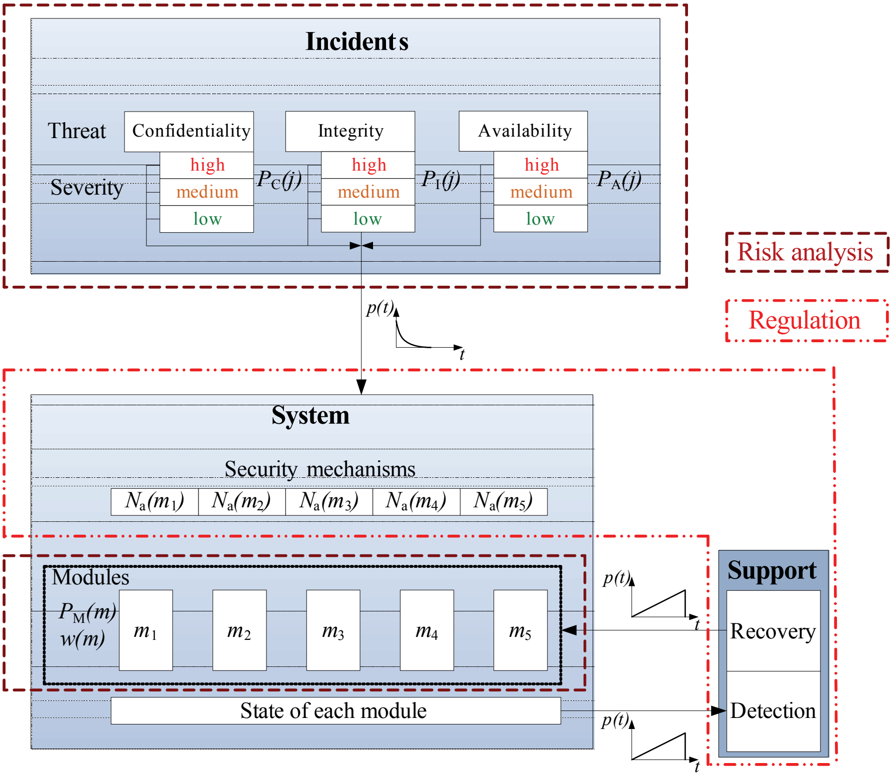
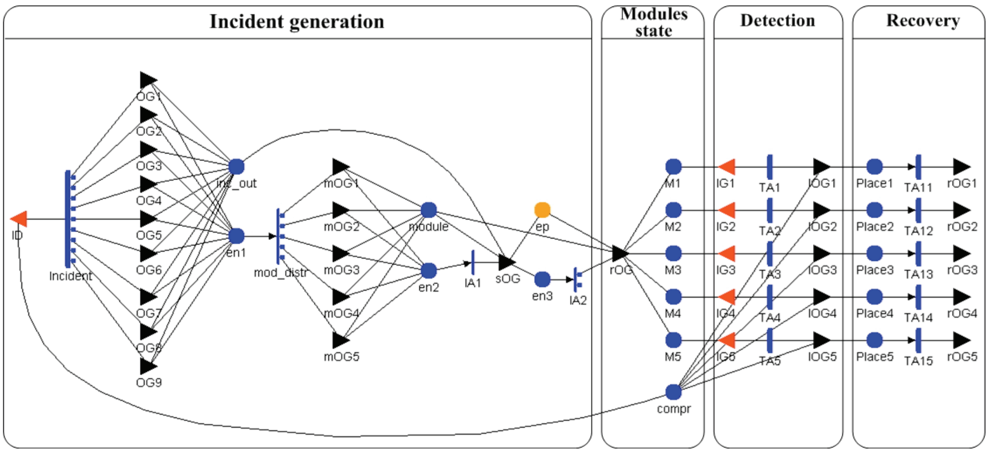


Fig. 2. Computer system model

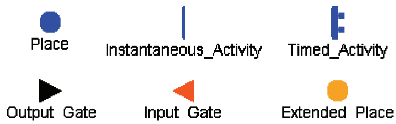
After the incident computer systems module is compromised or not, by that affecting the state of the whole system. The computer system can be in five states: normal (b_1), when no modules are compromised; one module compromised (b_2); when one or more modules are compromised and the system state change is detected and the system goes to the recovery state (b_3); if the incident is severe then more than half of the system modules can be compromised (b_4); or even all of them can be damaged (b_5).

It's more likely that the systems degradation will be detected faster and the problem will be addressed in the shorter interval than the one set by the law, that's why triangular distribution is used.

Computer system simulation model is composed using Stochastic Activity Network (SAN) formalism, which is quite similar to the stochastic Petri nets. Simulation model organised using Möbius tool by its design repeats the block diagram of the model (Fig 3a).



a)



b)

Fig. 3. Computer system survivability simulation model by using SAN: a) the model, b) SAN primitives.

SANs consist of such primitives: places, activities, input gates, and output gates (Fig. 3. b) [8]. Places contain tokens, which indicate the "value" or "state" of a place. Timed activities transfer tokens from one place to another in a set timely manner. Instantaneous activities specify zero-timed events. Cases (denoted by small circles on activities) are used to specify probabilistic choices. Input gates are used to define enabling predicates and completion functions. Output gates are used to define completion functions. The times between enabling and firing of activities can be distributed according to a variety of probability distributions, and the parameters of the distribution can be a function of the state.

SAN model consists of four main parts: incident generation, modules state, incident detection and system recovery. Timed activity *Incident* is used to generate incidents. We made an assumption that incidents are independent and their occurrence is distributed by Poisson law. Assumption that system is attacked three times per day was made based on risk analysis. The cases of timed activity *Incident* represent incident severity for different threats (to confidentiality, integrity and availability) estimated by probabilities acquired from risk analysis. Depending on which case was chosen after timed activity *incident* completes, one of the output gates OG1–OG9 write to *inc_out* place a number which represents incident threat and severity level. The instantaneous activity *mod_distr* is enabled by writing 1 to place *en1*. This activity is used to define module usage probabilities $P_M(m)$. Depending on probability $P_M(m)$ output gates mOG1–mOG5 modify *module* place and save affected module number. Then instantaneous activity IA1 is enabled by writing 1 to place *en2*. Output gate sOG according to generated threat type, incident severity level and targeted module, writes corresponding value to extended place *ep*. This value represents compromise probability of targeted module and is used to define probability distribution of IA2 activity two cases. If first case was chosen when the output gate rOG depending on targeted module writes 1 to corresponding place (M1–M2), which represents that this module is compromised. Otherwise the incident was unsuccessful. Input gates IG1–IG5 are used to define timed activities TA1–TA5 enabling predicates. Timed activities representing detection (TA1–TA5) and recovery (TA11–TA15) use triangular distributions, because the incident detection probability rises when time after system compromise is getting longer. Triangular distribution usage for recovery activities is justified based on the same assumptions.

5. COMPUTER SYSTEM SURVIVABILITY

Computer system survivability S is universal and quantitative characteristic showing its ability to provide the services it is intended to in the hostile environment, which influences the level of the provided service [9, 10].

When service or the system survives in the maximal functional state b_1 during the system usage time Δt_{all} then such characteristic can be called maximal survivability $S_{\max}$:

$$S_{\max} = \frac{\Delta t_{b1}}{\Delta t_{\text{all}}} \tag{1}$$

If system survives in the functional state, which represent half of its functionality described by functional state b_4 during the system usage time (Δt_{all}) then such characteristic can be called midrange survivability S_{mid}:

$$S_{\text{mid}} = \frac{\Delta t_{b4}}{\Delta t_{\text{all}}} \tag{2}$$

Different services or modules providing these services represent different importance to the mission of the system, this must be considered. Survivability of the system *S* can be described as:

$$S = \sum_{m} w(m)S(m)\,,\ 0 \le S(m) \le 1\,,\ \sum w(m) = 1\,,\ 0 \le w(m) \le 1 \tag{3}$$

where *S*(*m*) is the survivability of computer system module *m*, and *w*(*m*) is the weight of the module.

6. RESULTS

Incident severity influence on the modelled computer system was evaluated using defined incident sets from the mildest one to the most severe (Table 2).

Table 2. Incident severity sets

		{*j*}				
		1	2	3	4	5
j	1	0	0	0	0.5	1
	2	0	0.5	1	0.5	0
	3	1	0.5	0	0	0

The interval of the system staying in the compromised state rises when the severity of the incidents rise (Fig. 4a) and the computer system chance staying in the normal state is less when the incidents are more severe (Fig. 4b). Affect of the incidents targeting different threats differ because of the computer system module defending mechanism speciality and depend on the computer system, but the effect of more severe incidents is similar (Fig. 4c).

If incidents occur more frequently then the modelled computer system is more likely to be compromised (Fig. 5a). Different computer system modules are protected by the distinct security mechanism and because of popularity of the module, the chance of being compromised does differ. Though the more the protection mechanisms are stronger, the more it is likely that the system will stay in the normal state (Fig. 5b).

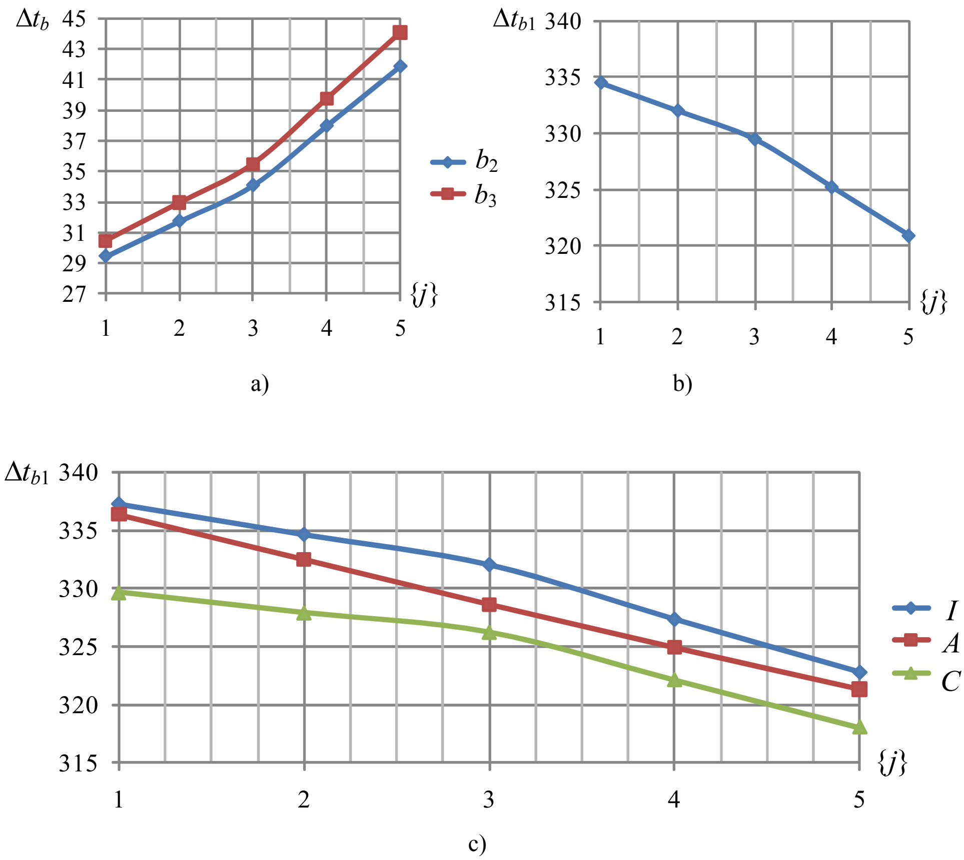


Fig. 4. Incident Severity Influence on System States:
a) b_2 and b_3 b) b_1 c) b_1 according to the threat

Survivability is the quantitative computer system security characteristic that is addressed in this research (Fig. 6). Maximal survivability S_{max} is the probability that the computer system will stay in the normal state after the incident. The probability that the half of the computer system modules will be functioning represented by S_{mid} is most possible. Computer system survivability S which has the average values best represents the influence of the computer system security mechanisms to the modelled computer system.

Different security mechanism sets $N_m(n)$, where n is the number of security mechanism set, determined to be most relevant to the modelled system by risk analysis were used. Security mechanism sets are represented by the ratio between the number of used security mechanisms and all possible third category system security mechanisms.

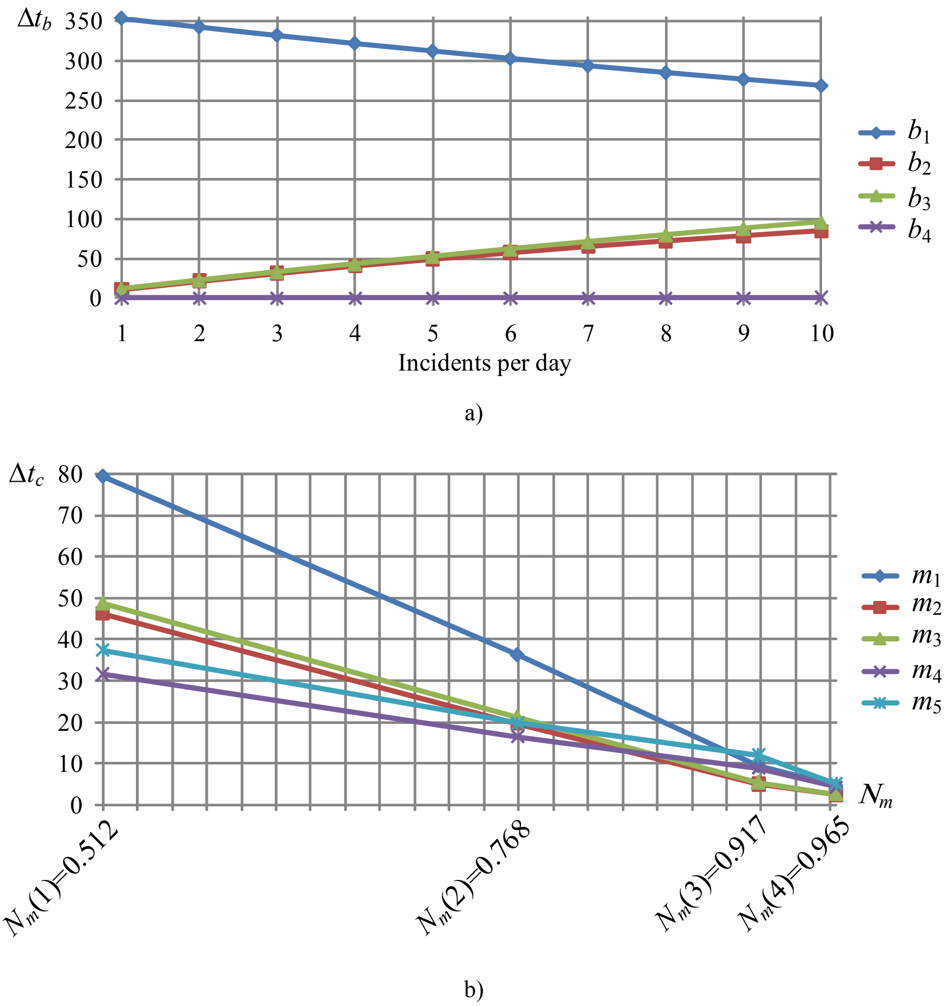


Fig. 5. a) Incident Occurrence Interval Influence on System States;
b) Protection Mechanism Set Influence on Normal System State according to the Module

The more security mechanisms are implemented, the higher survivability characteristics will be. Survivability graphs of the real information system would provide information which is critical to make a decision to install some security measure or not based on its costs and the expected affect on system survivability. Survivability characteristic of the modelled computer systems is quite linear because the exact influence and the cost of the protection mechanism is not evaluated, more detail risk analysis could provide more specific results with more significant security mechanism implementation thresholds.

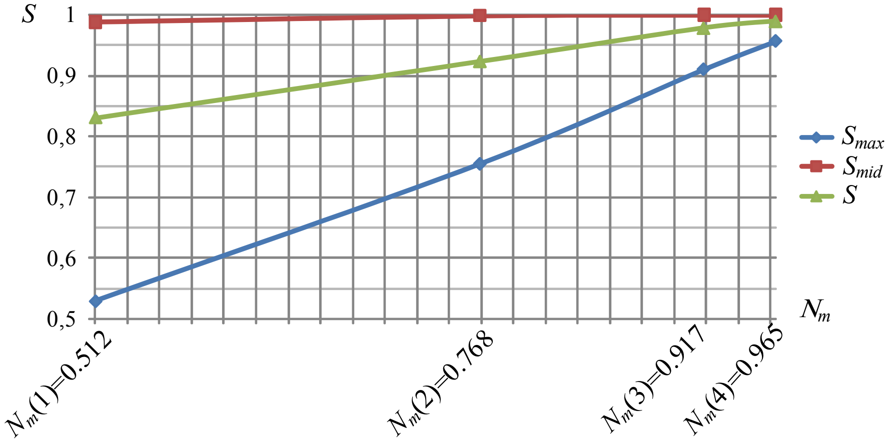


Fig. 6. Survivability characteristics

7. CONCLUSIONS

1. Computer system survivability simulation model was designed and implemented. The model allows evaluating computer system survivability according to incident severity, occurrence frequency, threat category (CIA) and protection mechanism strength.
2. The modelled computer system security mostly depends on the incident occurrence probability, on the strength of protection mechanisms, while the occurring incident severity has the least effect on the protected computer system.
3. Computer system security implementation regulation provides more sufficient information to the risk analysis and the targeted security level is more likely to be achieved.
4. Future plans: to evaluate and compare different category information systems, using more granular protection mechanism sets, using statistical data to test the simulation results.

REFERENCES

[1] GARŠVA E., *Computer system survivability modelling*, Electronics and Electrical Engineering, Kaunas: Technologija, 2006, Vol. 1(65), 48–51.
[2] GARŠVA E., *Computer System Survivability Modelling by Using Stochastic Activity Network*, SAFECOMP'06, 2006, Vol. 4166, 71–84.

[3] PAULAUSKAS N., GARSVA E., *Attacker Skill Level Distribution Estimation in the System Mean Time-to-Compromise*, 1st International Conference on Information Technology, Gdańsk, May 19–21, 2008, 463–466.
[4] *Clasification Guidelines for Goverment Institution and Office Information Systems according to the Processed Electronic Information*, 2007-07-11, No. 1V-247.
[5] *Requirements to Technical Security of Goverment Institution and Office Information Systems*, 2008-10-27, No. 1V-384.
[6] GORANIN N., CENYS A., *Genetic algorithm based internet worm propagation strategy modeline*, Information Technology and Control Vol. 37(2), 2008, 133–140.
[7] GORANIN N., CENYS A., Malware propagation modeling by the means of genetic algorithms, Electronics and Electrical Engineering, Kaunas: Technologija, Vol. 6 (86), 2008, 23–26.
[8] SANDERS W.H., *Möbius: model-based environment for validation of system reliability, availability, security, and performance*, Möbius Manual Version 2.3.1, 2010, https://www.mobius.illinois.edu/manual/MobiusManual.pdf
[9] MOORE A.P., ELLISON R.J., LINGER R.C., *Attack Modeling for Information Security and Survivability*, 2001, www.cert.org/archive/pdf/01tn001.pdf
[10] MOITRA S.D., KONDA S.L., *A Simulation Model for Managing Survivability of Networked Information Systems*, 2000, www.cert.org/research/00tr020.pdf

environment, generation,
test automation

Łukasz POBEREŻNIK*

AUTOMATIC GENERATION AND CONFIGURATION OF TEST ENVIRONMENTS

Modern software is developed to be used on various platforms, architectures, middleware and to work with multiple peer systems. It means that there are many possible configurations of a single application and its environment that are valid. This creates a challenge for test automation that has to cover as many installation variants as possible. Most of the currently used test automation tools assume that environment for application is ready before tests start. The goal of this research is to create tools that would generate more or less automatically different environment configurations, deploy and run tests on them. This work presets a model of such system based on the concept of exchangeable, hierarchical resources that are described using specially created rule system. In order to make test cases work in different environments a multi- layer structure of test scripts was introduced that separates general test steps from their platform dependent implementation. A prototype was constructed to verify the proposed concept and algorithms. Results showed that it is possible to create such tools that can generate all valid configurations and flexibly control test environment coverage by implementing the paradigm: write once, run on many.

1. INTRODUCTION AND PROBLEM DESCRIPTION

Many software products, especially those off-the-shelf solutions, must be able to work in different environments and configurations in order to fit into customer's existing infrastructure. This requires application to support different operating systems, architectures, application servers, databases and other middleware. It is common problem that one combination of elements mentioned before will work, but another will cause problems. In order to claim support for different configurations, manufacturer must test as many possible software setups as possible.

Lets take a simple use case: an application working on two operating systems, with three database servers and two application servers. It will give about 2*3*2 = 12

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

different configurations to test. If we add another variable: 32-bit or 64-bit architecture, it will double possible configurations to at most 24. Adding new configurable element to environment tends to increase the number of possible setups exponentially. Not all configurations may be possible to create (for example some middleware may not be available for all operating systems), but it still is significant number of variants to test.

For creating test environments three methods seem to be most frequently used:

- manual installation and reconfiguration of environment components as a pre-test activity, outside testing process (see Rational Unified Process [1]),
- virtual machine images (see [3]) – allows easy environment cleanup but takes plenty of disk space, computing power and expensive software and hardware,
- Integration of the environment preparation inside test scripts. It increases testing time and makes test cases environment dependent.

In order to reduce costs of such multi-configuration testing the following strategy is commonly used:

1. Select at least two most popular configurations,
2. during the development phase, test the software using only those configurations,
3. when the number of newly discovered defects will fall below certain level, run tests on other configurations.

The problem is between step 2 and 3: it is not trivial to increase the number of environment configurations. This usually requires one to obtain new machines, configure them and rewrite tests to support the new configurations.

All cases above lack flexibility and scalability. It is not possible to easily increase the number of tested configurations. A lot of disk space and computing power or modification of test cases is required.

The solution should have following properties:

- use only one test case and then execute it on several different environments,
- automatically prepare environments based on software requirements and test cases that we want to run,
- be able to scale number of configurations from 1 to all possible,
- able to handle preparation of about one hundred different configurations,
- environment model should not limit possible use cases or limit them as little as possible.

2. PROPOSITION OF SOLUTION

In this work we will focus on generating environment configurations. Their construction and deployment are subject of further research.

Software systems are designed to fulfill certain requirements stated by the clients. Usually software producers tend to have the same function set in each environment their application is working on.

Test cases that check single requirement usually have similar steps in each environment. However, details of each step are different – path, script name, parameters, configuration file, etc.

In order to design a test case structure that will fulfill requirements stated above we need to abstract the general test steps from system level implementation.

2.1. STRUCTURE OF TEST CASES

Let's divide test case into two separate layers:

- test scenario,
- test fragments.

Test fragments are the basic building blocks of test automation. They are environment aware, so it means they behave differently according to the environment they are working on. Their role is to abstract implementation details from the layer above. Test fragments can contain testing logic or utility functions (environment preparation, clean up, logs collection etc.).

Test scenario is implementation of a test case that uses test fragments to perform operations on system. It contains only general steps that are environment independent. It is used only to control the execution and data flow between test fragments.

Test scenario/test fragment concept described above was the result of previous work described in [2].

2.2. ENVIRONMENT MODEL

In order to automatically generate and deploy configuration we need to have a model of environment. This model should be as general as possible so I would not impose any constraints on test cases.

Lets define *resource* as a part of test automation environment. It can be anything that is required to perform tests that is not an algorithm. Each resource has:

- unique identifier,
- type,
- set of properties,
- relations to other resources.

Properties are key-value pairs. Key is a string and value can be also a string, integer, floating point number, boolean, enumerable choice, range of numbers. They can be set either by a configuration generator – to tell deployment engine how resource should be installed, or by deployment engine itself – as the effect of collecting information from installer.

Properties can be accessed by scenarios and fragments to gather information required to perform test activities.

Resource type can use inheritance similarly to classes known from object oriented languages. Type determines the list of obligatory properties that must be available in each resource of that type.

Each *relation* consists three elements – resource A, resource B and relation type. Typical relation types are: *connected to*, *installed on*. Those relations are needed to build installation plan. *Connected to* means that two resources can be installed separately and then configured to work together. *Installed on* means that one resource must be ready before second one can be installed. Example of software system with relationships marked between resources is shown on Figure 1.

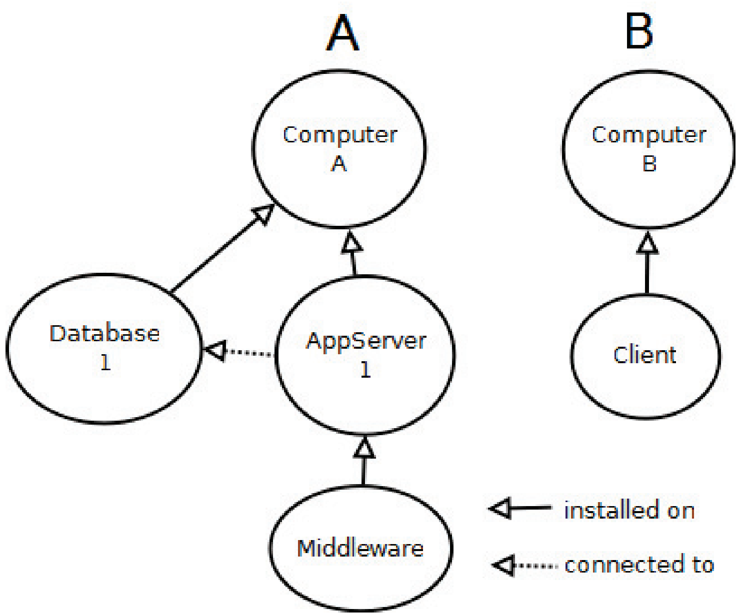


Fig. 1. Example of system with marked relations between dependent resources.
Tree A represents environment for server application. Tree B is environment for client application

Unique identifier is a string available only when resource is *materialized*. By materialization we mean that resource has become an entity that can be accessed using properties and relations. Not materialized resource is also called resource description. Some of the properties may become obligatory when resource is materialized - especially those ones that allow test fragments to communicate with the resource. Example of such property can be IP address.

Resources can be implemented as classes and objects in object oriented programming languages.

Each scenario defines a set of resources that it requires to perform tests. Those resources are represented by variables. Each variable has its name and type that corresponds to resource type.

Variables must represent **all resources** required by scenario. However some resources need other resources to work as defined by relations. Those resources are represented by indirect variables. Indirect variable name is constructed by adding a dot and relation name to base variable. For example scenario needs database *DB* and database needs operating system (defined by relation called *os*). Database has relation called *os*. Then indirect variable name is *DB.os*.

There entities are called *rules*. Rules are logical expressions that use resource properties and relations. They are used to define which environment configurations are possible. Each rule is a logical AND or OR of one or more terms. Brackets are also allowed to group subexpressions.

Term is a triplet of property reference, operator and value. Property reference is a variable name (direct or indirect) followed by dot and property name. Operator can be equal or not equal sign. Value can be a constant string or another property reference.

Below is a simplified grammar of such rule system in EBNF:

```
rule:= (<matcher> ":")? <expression>
matcher:=(<id_string> "role" | "is" <q_string>)*
expression:= <term> {"and" | "or" <term> }*
term:= (<property> ("=" | "<>" | "<" | ">" | "<=" | ">=") (
<q_string> | <property>)) | "(" <term> ")" | "not" <expression>
property:= <id_string> ("." <id_string>) *
id_string:= "A"-"Z" | "a"-"z" | "0"-"9
q_string:= "\"" (~["\""])+ "\""
```

Each test scenario defines a list of required resources that are needed to perform testing procedure. Those resources are expressed as a set of rules (described by grammar above):

```
SRV1.os.type=Linux and SRV1.memory>512MB and
DB1.type=MSSQL and DB1.version>=2005
```

Rules above mean that scenario needs Linux server with more than 512MB RAM and MS SQL database at least version 2005.

We can distinguish three types of rules used to generate configurations:

- test scenario rules (described above),
- general validation rules,
- constraining rules.

General validation rules are used to define general environment limitations specifying relations between resources that are not possible or values for properties that are out of range. Because they may refer to any kind of resource there must be a way to match them with resources required by scenario. Validation rules are preceded by a matching expression that allows them to associate with proper variables defined in scenario. Colon separates matching expression and main part of the rule.

For example they can specify that MS SQL database cannot be used on Linux:

```
RES is database : RES.db-type=MSSQL and RES.os.type<>Linux
```

Constraining rules are additional rules used to limit configurations only to those that are most important, for example used by the majority of customers. Those rules are applied to scenario rules and use the same set of variables.

General concept of generating environments is presented in Figure 2.

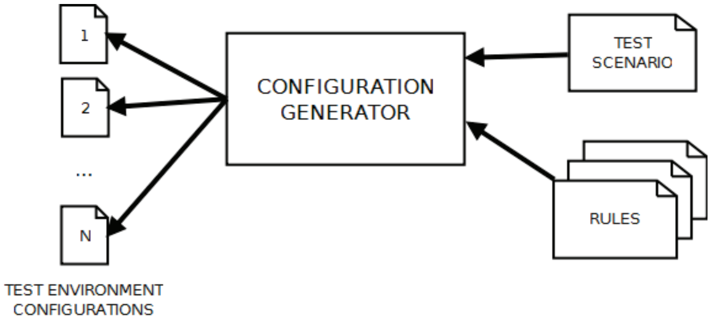


Fig. 2. Information collected from test scenario and set of external rules are used to generate different environment variations

2.3. GENERATING ENVIRONMENT CONFIGURATIONS

To generate all possible variants of configuration we need to use Cartesian product operator on each set that contains every possible value of properties. Then we need to use selection operator to filter out impossible and unwanted combinations.

Those two operators can be found in every SQL database. The general idea is to put resources into separate tables and then cross join them together and filter using rules converted to logical expression that will be put in WHERE clause.

Configuration generation algorithm can be presented in following steps:

- look into scenario and find all required variables, add them to variables set,
- for each of the variables in the set determine indirect variables and add them to the set,

- repeat previous step until it won't be possible to add new variables,
- parse scenario rules, construct syntax tree,
- scan syntax tree and get possible values of all properties described in all rules and scenarios,
- place those values in separate tables,
- for each resource and property required by scenario create a column in SELECT statement and connect them using JOIN expression,
- transform syntax tree into SQL logical expression that will be put into WHERE statement.

To demonstrate how algorithm works, assume that example tested software works on two operating systems: Windows, Linux. It can be installed on two application servers: JBoss and WebSphere and can use MySQL, MS SQL or Derby databases. This will require three tables in configuration database: OP_SYS, APP_SRV and DB. Example of such tables is shown in Table 1.

Table 1. Tables that contain all possible values of resource properties.
Note that there are more types of resources that are supported by tested system

OP SYS	APP SRV	DB
Linux	Boss	Oracle
Windows	WebSphere	MSSQL
Solaris	WebLogic	Db2
	Geronimo	MySQL

Test scenario will require operating database and application server. This can be defined by following rule:

```
 ( DB.os.type in (linux,windows) and
( DB.type in (mysql,derby,mssql) and
( APP_SRV.type in (jboss,websphere)
```

We also need a constraining rule because MS SQL works only on windows:

```
X is database : not (X.type=mssql and not X.os.type=windows)
```

Selection statement will need to join all the tables:

```
SELECT S.NAME as OS, A.NAME as APP_SRV, D.NAME as DB FROM
OP_SYS S, APP_SRV A, DB D ...
```

Scenario rules will be transformed into the following SQL expression:

```
... WHERE S.NAME in ('windows','linux')
AND A.NAME in ('jboss','websphere')
AND D.NAME in ('mysql','mssql','derby') ...
```

There must also be a constraing rule:

```
… AND NOT (D.NAME='mssql' AND NOT (S.NAME='windows'))
```

Output from SQL query above should look as in Table 2.

Using SQL database has also one more advantage: similar query generated from rules may be used to query table with existing configurations that will act as a cache. If configuration already exists, it will be reused (for example virtual machine image will be restored).

Table 2. Output from SQL query that contains all valid combinations of environment that was described in the text

OS	APP_SRV	DB
Windows	Websphere	MSSQL
Windows	JBoss	MSSQL
Windows	WebSphere	MySQL
Windows	JBoss	MySQL
Windows	WebSphere	Derby
Windows	JBoss	Derby
Linux	WebSphere	MySQL
Linux	JBoss	MySQL
Linux	WebSphere	Derby
Linux	JBoss	Derby

3. EXAMPLES OF APPLICATIONS

Application is designed to work on many Unix and Windows systems. However customers mostly use it on Windows 2003 and Red Hat Linux systems. Testers can create a rule that will create test environments for that systems:

```
( R1.os.type=Linux and R1.os.subtype=redhat )
or ( R1.os.type=Windows and R1.os.subtype=2003 )
```

Limited number of test environments will reduce testing time and required resources and allow tests to be run in daily manner.

At the end of development cycle in order to find all possible defects this rule can be modified to include other supported operating systems:

```
( R1.os.type=Linux or R1.os.type=Windows or
R1.os.type=Solaris )
```

A little effort put into changing this rule will result with many new configurations to test on.

Customers often want to know how application will handle the load depending on available resources. They want to know for example what is the minimal amount of memory required to handle 100 users working simultaneously.

There can be a rule created that will construct the same testing environment but with different amount of available memory (using virtualization):

```
( R1.memory in (1Gb,2Gb,3Gb,4Gb) )
```

The same test can be run on each of the generated environments and comparative performance chart can be created from collected data.

4. IMPLEMENTATION

To test the above presented model a prototype in Java was created. This language was chosen because it is object oriented, portable, has strict syntax and has built-in reflection mechanisms and annotations.

Resources are modeled as objects. Fields inside class are properties. Relations are implemented by adding annotation *Relation* to the property field. Scenario is implemented as plain Java class with properties annotated with *ResourceProperty* that indicate direct variables. Use of plain Java objects (POJO) and annotations simplifies construction of test code.

After environment configurations are generated and deployed, test execution engine injects materialized resources into instances of scenario class. Rules are processed using JavaCC based parser that builds object model of a rule and then converts it to SQL expression. To generate configurations combination of Derby SQL database is used. It does not handle large loads but it is sufficient for prototyping.

5. SUMMARY

Research showed that the proposed approach can be used to create test environments in scenario-based test automation. The proposed solution allows handling a large number of valid environment configurations for various software systems. It is also able to flexibly control a number of covered configurations by using rules. Tester can easily change the number of configurations from 1 to the maximum possible, still keeping them valid for the application under tests. General test environment model does not limit resources to only software and hardware items.

Theh presented solution is an important step in our research because it fills the gap between test scenarios and environment deployment tools, so the test engineer can focus on developing test automation code, not environment setup.

In the next phase, the research will focus on execution of deployment plans. This will allow to reduce human resources needed to prepare test environment. The goal is to allow testers to focus on tests scope and details rather than on tedious and repetitive test execution. We plan to leverage the use of virtualization (for example VMware Lab Manager [4] and Cloud Computing) to automate and optimize environment preparation with limited hardware resources.

REFERENCES

[1] KRUCHTEN P., *The Rational Unified Process: An Introduction*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3 edition, 2003.

[2] POBEREŻNIK Ł., ZIELIŃSKA A., BYRSKI A., *New concepts of software test automation in heterogeneus production environments*. In Information Systems Architecture and Technology – New Developments in Web-Age Information Systems, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2010, p. 285.

[3] SEETHARAMAN S., MURTHY K., *Test optimization using software virtualization*, IEEE Software, 23, September 2006, 66–69.

[4] VMware Inc., *Accelerating test management through self-service provisioning*, Technical Whitepaper, http://www.vmware.com/Files/pdf/solutions/vmware-devtest-accelerate-test-management-wp-en.pdf

filesystem snapshot,
on-line data backup

Mateusz SMOLIŃSKI*

FILESYSTEM STRUCTURES INFLUENCE ON SNAPSHOT STORAGE SPACE USAGE

Snapshot is a technique of preserving consistent in time state of selected object. This technique is often used to create archive contains consistent in time files backup from filesystem, that continuously on-line updating data. Snapshot creation it is not burdensome operation for operating system because requires to save data block state before modifying it, so unchanged blocks are never duplicated. According to copy on write rule data stored in a single block can be saved in snapshot storage space only once. Modern filesystems offer snapshots functionality, then snapshot storage space is supported by allocated blocks from filesystem structures. Older filesystem types also could be snapshoted using snapshot created on block device level, but not all volume managers support this functionality. Use of snapshot storage space depends on filesystem metadata structures and executed operations on stored files and directories, however snapshot storage space should not be overflow. When snapshot storage space is fulfilled, snapshot does not provide consistent in time filesystem state and is useless. This disadvantage can occur only when snapshot storage space size is smaller then filesystem storage space size, in practice it is the most frequent case. Presented research work and analysis of filesystem structures modification impact on the use of storage space for snapshot done on the block device level enable to classify popular filesystem types according to vulnerability to overflow snapshot storage space. The obtained results allow users using snapshots to choose an appropriate filesystem in storage space preparation process and also estimate the required size of storage space used by the snapshot before file backup creation.

1. INTRODUCTION

1.1. DATA BACKUP

Modern operating system offers tools to automate data backup creation process. System administrator could use them according to defined rules to provide regular data backup securing accidental data loss. Selection of the backup frequency, type,

* Technical University of Lodz.

format and external location are main features of backup policy. The effects of these settings have been widely described in the literature i.e. selecting the backup type as a logical or physical affects system resource utilization during data backup creation and recovery [4,6].

The primary parameter for preserving the history of data changes is the frequency of backup creation. In the case of an irrevocable loss of data such as system failure or user error data state can be restored at the creation time of the last backup. Apart from the selection of backup frequency there is a need to maintain a consistent data state at the time of backup creation. A simple solution to the problem is to prevent changes to the data during backup creation i.e. off-line data for other tasks or accessed in read only mode. In production systems, this solution is often impossible to use because of the continuous on-line access to data in read-write mode. In this situation it is essential to use the functionality of snapshots.

1.2. MAIN PRINCIPLE OF SNAPSHOT ACTIVITY

In general, snapshot is a state of a object in particular point in time. Snapshot technique can be used to object located in volatile memory or non-volatile storage space [6, 7]. In the following parts will be considered snapshots of filesystem localized on single block device (i.e. hard disk partition).

Snapshots use COW (Copy On Write) rule, which is main principle of snapshot activity. This rule requires the preservation of state for each addressed data fragment before making any its changes. State preservation for each data fragment can be performed by making copy once before first change in this data fragment occurs. All next data changes in this fragment do not require another data copy from this addressed fragment, that ensure efficient use of storage space.

For block device such a hard disk a basic allocation unit is a sector, so fragment of its storage space could be represented by block, which is a group of sectors. Also in filesystem basic allocation unit is a block, which size is a multiple sectors, where usually multiplicity is a power of two. Saving block state has impact on performance in first write operations realized by block device after snapshot creation, especially if many snapshots were created for this device or related filesystem.

To store the changed blocks for snapshot a separate buffer is used, that contains dedicated snapshot storage space. Due to the application of the COW rule buffer size can be less then original storage space size, for which the snapshot is made. Buffer size depends on the number of blocks changed since the time of snapshot creation. Effective storage space use is a advantage, in practice required space for snapshot size often is between 20–40% of original data storage space size. If the snapshot buffer becomes full it is useless, because snapshot no longer guarantee consistent data state in time. When snapshot was created to create a backup it is important to check after backup creation whether snapshot buffer has not been overflow. Situation never occurs if the snapshot buffer size equals origin data storage space size.

1.3. SNAPSHOT TYPES

Snapshot can be created on filesystem or block device (volume) level. Modern filesystems offer snapshot functionality base on its structures, examples are ZFS (Zettabyte FileSystem), BTRFS (B-tree FileSystem) or popular NTFS (New Technology FileSystem) which require VSS (Volume Shadow copy Service) [1–3, 5, 10]. Use NTFS in the latest Windows operating systems offers the ability to recover previous versions of files and directories without administrator privileges. This functionality is only available for snapshots performed at the level of block devices because of the unfamiliar a particular filesystem structures.

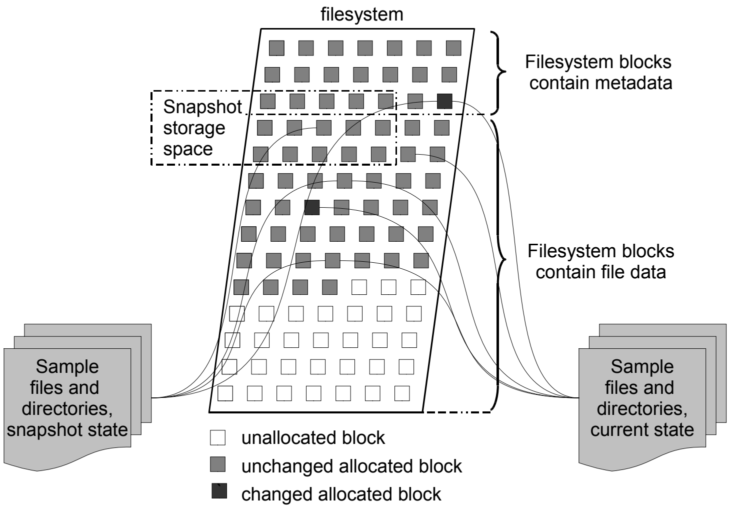


Fig. 1. Storage of filesystem with snapshot functionality

In some situations snapshot on filesystem level efficiently use storage space i.e. creation of file storing data. This is due to application of the COW rule only to the allocated earlier filesystem blocks. Then block copy can not be performed, instead this new state is saved in newly allocated filesystem block. In addition snapshot storage space can be dynamically changed depending on the actual needs, only if unallocated filesystem blocks are available.

Block device level snapshot offers among others LVM (Linux Volume Manager) for logical volumes, which use permanent sector mapping tables supported by DM (Device Mapper) [9]. Snapshots for block devices are more flexible because of the variety of applications i.e. can be used to create the frozen state of storage space for operating system. Above all allow to create a snapshot for legacy filesystem, which normally do not offer this functionality. In contrast to snapshots performed on the filesystem level snapshots that are created on the level of block devices has fixed storage space size. In the case of LVM, it is even possible to restore the state saved in the snapshot to its original block device. Snapshot at the block device level without additional solutions are not able to guarantee filesystem consistency at the time of the snapshot creation. With this type of snapshot it is not possible to distinguish between blocks that contain data from files and filesystem metadata.

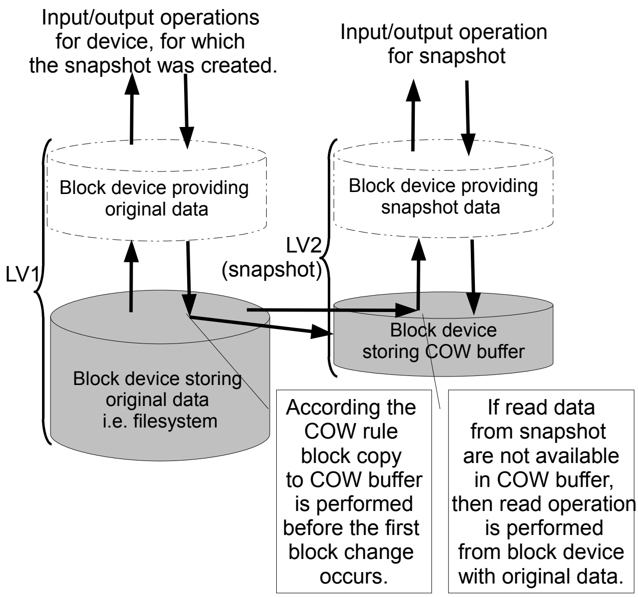


Fig. 2. Snapshot on block device level

Both types of snapshot can offer standard read-only mode or read-write mode (known as clone functionality), then all changes are saved in the snapshot storage space. Write operations in snapshot does not alter directly the original data storage space.

2. FILESYSTEMS CLASSIFICATION

2.1. DESCRIPTION OF RESEARCH WORKS

A research were performed using only block device snapshots approaches. In order to examine the impact of changes in the structure of each filesystem to use the snapshot buffer simple test method was developed. The research were performed in the 64-bit Linux operating system from Fedora 15 distribution (kernel version 2.6.38.8-32) for legacy filesystems: Ext3 and Ext4 (e2fsprogs version 1.41.14), XFS (xfsprogs version 3.1.4), JFS (jfsutils version 1.1.13), ReiserFS (reiserfs–utils 3.6.21), FAT32 (dosfstools version 3.0.11). All tested filesystems are commonly used in production environments, but none of them offers snapshot functionality. All filesystems was created with 4KiB block size on individual 20GiB logical volume managed by LVM (version 2.02.84). Also block device level snapshots chunk size was selected to 4KiB, each snapshot was separated logical volume managed by LVM. Sizes of filesystem block and snapshot chunk are equal to ensure the highest possible accuracy in testing the impact of filesystem structures for the use of storage space snapshot. Selected block size fit also to average file size given in some research [8].

Test are performed using fs_mark software tool from sourceforge (version 3.3). A single test required the creation of a fixed number of files without any data. The name of each file was 40 characters long. Files are grouped in a single level directory structure with fixed number of directories. In each testing case after files and directories creation, the number of blocks that were saved in the snapshot buffer was inspected. The lower number of blocks saved in the snapshot buffer means less impact of changes in the filesystem structures of the buffer occupancy.

2.2. THE RESULTS OF RESEARCH WORKS

In each filesystem 20 tests were performed, respectively for 1000, 750, 500 and 250 files in directories, which number were respectively selected for 1000, 750, 500, 250 and filesystem root directory only case. In first test in single filesystem a total of one million files was created without any contained data. Although each tested filesystem had the same block device storage space size, the number of allocation units were not identical. First of all, this is due to differences in the physical layer design in each filesystem. Specific differences are evident in the filesystem structures. Some filesystems use indexes to accelerate search selected by name file in directory. Also directory entries format is different in tested filesystems. Similarly, data storage structures of the available free blocks are significantly different in various filesystems. Some tested filesystems supports extends, that are sequence of blocks. Additional filesystem functionality like journaling also has impact on obtained results. Filesystem journal typically assures filesystem structures consistency, however this requires additional write operations determining which blocks in filesystem are currently being modified. Each tested journaling filesystem had the standard journal settings. Also collection of additional information releted with files and directories are different in various filesystems. The stored metadata includes among others ownership, access permissions matrix, last access timestamp and file attributs.

FAT32 filesystem does not store many additional data releated with file, therefore changes in its structures are limted to fixed minimal block number (Tab. 1). This filesystem does not have journal and use linear block allocation policy, that has impact on directories structures sizes in performed test.

Table 1. Snapshot storage space use for FAT32 filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	362456 blocks	241560 blocks	161072 blocks	80616 blocks
750 directories	241520 blocks	181176 blocks	120824 blocks	60464 blocks
500 directories	161040 blocks	120784 blocks	80568 blocks	40328 blocks
250 directories	80544 blocks	60416 blocks	40304 blocks	20176 blocks
Root directory only	360 blocks	280 blocks	200 blocks	120 blocks

Characteristics of Ext3 and Ext4 filesystem structures does not significantly differ (Tab. 2,3). Metadata of stored files include timestamps, user and group identifiers and list of access permissions. File data are stored in dedicated i-node structure, number of i-nodes is determined in filesystem creation process. Ext3/4 filesystem has also dedicated structures for blocks and i-nodes allocation. Additionally filesystem resource units are grouped to provide better performance. Allocation policy ensures to localise file data in the same group, where directory resources are stored. Both tested versions of Ext filesystem use journaling, directory indexes and hierarchical block allocation policy for every file or directory.

Table 2. Snapshot storage space use for Ext3 filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	1903728 blocks	624192 blocks	462952 blocks	300448 blocks
750 directories	621760 blocks	501872 blocks	381176 blocks	259368 blocks
500 directories	459904 blocks	379624 blocks	299992 blocks	218008 blocks
250 directories	296928 blocks	256680 blocks	217720 blocks	152968 blocks
Root directory only	1432 blocks	1128 blocks	760 blocks	456 blocks

Fundamental difference between Ext3 and Ext4 filesystem is block addresing capacity, because Ext4 filesystem can even address 1EiB storage space.

Table 3. Snapshot storage space use for Ext4 filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	1733184 blocks	766192 blocks	589512 blocks	430808 blocks
750 directories	752584 blocks	640712 blocks	508416 blocks	389264 blocks
500 directories	589704 blocks	515208 blocks	427280 blocks	274056 blocks
250 directories	427528 blocks	389648 blocks	305688 blocks	86096 blocks
Root directory only	1416 blocks	1112 blocks	856 blocks	616 blocks

XFS filesystem use journaling and 64-bit block addressing. Stored file and directory metadata include owner identifier, POSIX permissions and access timestamps. File nodes are created dynamically and XFS structures provide efficient storage utilization according to filesystem functionalities (Tab. 4).

Table 4. Snapshot storage space use for XFS filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	965968 blocks	766568 blocks	589592 blocks	430824 blocks
750 directories	752480 blocks	640480 blocks	508392 blocks	389296 blocks
500 directories	589936 blocks	515304 blocks	427136 blocks	297424 blocks
250 directories	427120 blocks	390152 blocks	299688 blocks	86096 blocks
Root directory only	1448 blocks	1128 blocks	872 blocks	632 blocks

JFS was designed as journaling filesystem with directory indexes, it stores additional file data like timestamps, permissions and ownership. This filesystem uses allocation groups, that divide aggregate space into separate chunks. Allocation policy is to store related blocks and i-nodes in the same resource group. In JFS i-node structure can be created dynamically. JFS structures were prepared to high input/output operation performance and are fine grained, that feature is presented in research results (Tab. 5).

Table 5. Snapshot storage space use for JFS filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	1300560 blocks	526680 blocks	336880 blocks	187120 blocks
750 directories	515792 blocks	400248 blocks	257776 blocks	145544 blocks
500 directories	350744 blocks	273680 blocks	178776 blocks	103960 blocks
250 directories	185832 blocks	147256 blocks	99808 blocks	62368 blocks
Root directory only	5152 blocks	3880 blocks	2472 blocks	1104 blocks

ReiserFS is journaling filesystem which minimizes internal fragmentation. Other characteristic feature is filesystem global index. ReiserFS like other tested journaling filesystems allows to store file metadata including access timestamps and permissions with file/directory ownership. This filesystem has complex structures and high performance in file and directory operations. Additionally its structures are compacted, therefore do does not affect a significant number of blocks saved in snapshot buffer (Tab. 6).

Table 6. Snapshot storage space use for ReiserFS filesystem

	1000 files in directory	750 files in directory	500 files in directory	250 files in directory
1000 directories	354912 blocks	998560 blocks	690872 blocks	391096 blocks
750 directories	990272 blocks	765488 blocks	534640 blocks	309824 blocks
500 directories	682240 blocks	532312 blocks	378392 blocks	230048 blocks
250 directories	374064 blocks	299112 blocks	227960 blocks	149216 blocks
Root directory only	80 blocks	80 blocks	80 blocks	80 blocks

2.3. ANALYSES OF RESEARCH RESULTS

Presented research results show the impact of particular filesystem structures changes to the snapshot buffer utilization. Results analyses show, that number of

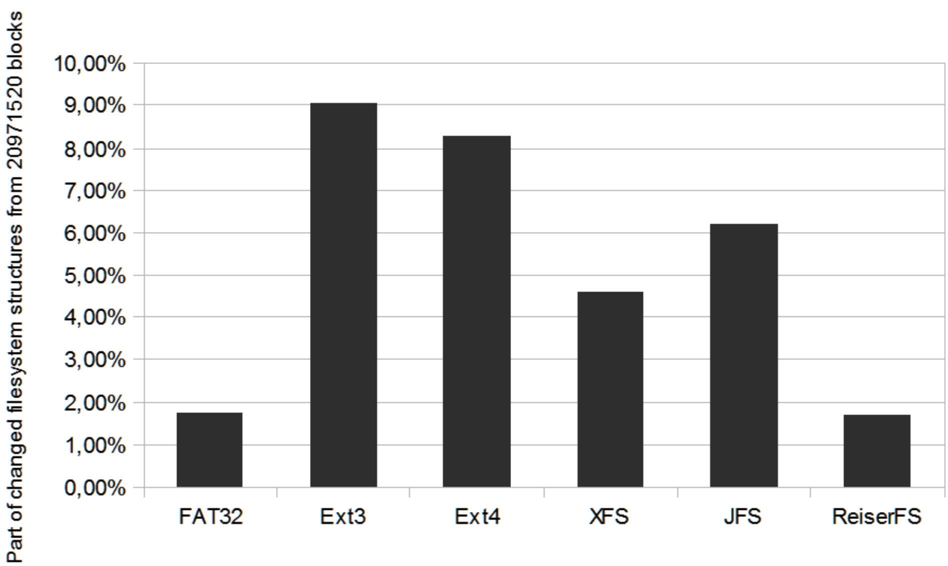


Fig. 3. Comparison of filesystem structures influence on snapshot buffer utilization

modified blocks does not always increase according to number of created directories and files. In order to classify filesystems one milion created files case was chossen, filesystem structures coparison presents figure 3.

Futher analyses of research results show that file and directory create operation change filesystem metadata and has influence on snapshot storage space usage, but also compacted filesystem structures have impact on snapshot buffer utilization. For legacy filesystems in single test multiple differences are reported in snapshot storage space usage. For example results of test with creation of one milion files in many directories show that snapshot buffer for Ext3 filesystem is used five times more than in ReiserFS.

3. SUMMARY

3.1. CONCLUSIONS

Analyses of obtained research results allow to conclude that the structures of different legacy filesystems have a significant impact on the utilization of storage space snapshot. Legacy filesystems have multiple differences in usage of snapshot buffer block resources in the same test scenario. If application of filesystem supporting snapshots internally is not possible and volume manager offers snapshot on block device level, then FAT32, XFS or ReiserFS is recommended for on-line backup purposes. The structures of these filesystems reduce the risk of overfilling the snapshot buffer before the backup completion.

3.2. FUTURE WORKS

Further research will be concern on legacy filesystems and storage space usage of snapshots performed at block device level including different naming schemes, various files operations and multiple levels of directory hierarchy.

REFERENCES

[1] COWART R., KNITTEL B., *Microsoft Windows 7 in Depth*, Que, 2009.

[2] HELLWING C., *XFS the big storage file system for Linux*, Proceedings of the USENIX Conference, 2009.

[3] HEGER D., *Work dependent performance evaluation of the btrfs and ZFS filesystems*, Computer Measurement Group International Conference Proceedings, 2009.

[4] HUTCHINSON N.C., *Logical vs. physical file system backup*, Proceedings of the 3rd Symposium on Operating System Design and Implementation, 1999.

[5] KARA J., *Ext4, btrfs, and the others*, In Up Times – Proceedings of Linux-Kongress and OpenSolaris Developer Conference, 2009, 99–111.
[6] NELSON S., *Pro data backup and recovery*, Apress 2011.
[7] TANNENBAUM A., *Operating systems*, 3rd edition, Prentice Hall, 2008.
[8] TANNENBAUM A., HERDER J., BOS H., *File size distribution on UNIX systems: then and now*, ACM SIGOPS Operating Systems Review, Vol. 40 Iss. 1, 2006.
[9] TEIGLAND D., MAUELSHAGEN H., *Volume Managers in Linux*, Proceedings of the USENIX Conference, 2001, 185–198.
[10] WATANABE S., *Solaris 10 ZFS Essentials*, Prentice Hall, 2009.

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 978-83-7493-630-9