Biblioteka Informatyki
Szkół Wyższych

Information Systems Architecture and Technology

Contemporary Approaches to Design and Evaluation of Information Systems

Library of Informatics of University Level Schools

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

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

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

Wrocław University of Technology

Information Systems Architecture and Technology

Contemporary Approaches to Design and Evaluation of Information Systems

Editors
Leszek Borzemski
Adam Grzech
Jerzy Świątek
Zofia Wilimowska

Wrocław 2014

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.oficyna.pwr.wroc.pl;
e-mail: oficwyd@pwr.wroc.pl
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-855-6

CONTENTS

INTRODUCTION

CONTEMPORARY APPROACHES TO DESIGN AND EVALUATION OF INFORMATION SYSTEMS

Information system (IS) is an integrated set of components for collecting, storing, and processing data and for delivering information, knowledge, and digital products[1]. Recent advances to design and evaluation of information systems include diverse technical and non-technical approaches. This book would like to present these progresses in:

- Artificial Information in Modern Information Systems
- Computer-Human Interaction
- Knowledge Discovery and Data Mining
- Big Data Systems and Applications
- Virtual and Cloud Computing
- High Performance Computing
- Web Performance Evaluation
- Multiagent Technologies and Systems

This book consists of chapters presenting a selected research on current challenges in design and evaluation of modern information systems. It will help the readers to think about their own problems and come up with solutions presented in this book.

Chapter 1 presents few methods of selecting shares based on risk parameter as standard deviation, Value at Risk and MST. They were used to conduct a case study **based on Polish WIG-BANK index.**

Chapter 2 presents the results on Web performance experiment made using PlatetLab network infrastructure. PlanetLab is an open platform for developing, deploying and accessing planetary-scale services. Experiment was set up to run ten measuring agents once per hour to download files from twelve mirror servers of Linux

[1] "information system". *Encyclopædia Britannica. Encyclopædia Britannica Online.* Encyclopædia Britannica Inc., 2014. Web. 04 Sept. 2014. <http://www.britannica.com/EBchecked/topic/287895/information-system>.

Gentoo operating system. Three regions were studied: America, Europe and Israel, and Asia and Australia.

Chapter 3 discusses methodologies for grouping of autonomous systems on the Internet. Studies which were conducted classified nearly 38 thousand active autonomous systems.

Chapter 4 presents a comparison of computing power of GPUs (Graphics Processing Units) and CPUs (Central Processing Units) for chosen matrix operations to solve the Lyapunov equations. Parallel computing for solving the Lyapunov equation for large scale matrices is recommended due to the computational cost O(n3). The comparison is performed for the same algorithms implemented for both GPU and CPU environment.

Chapter 5 proposes a regular expression matching application, which can be configured to reflect a certain computation to data intensity ratio. The authors support its usefulness by showing execution times of their OpenCL implementation on selected CPU and GPU devices.

Chapter 6 discusses virtual environments consisting of large amounts of virtual machines requiring administrative monitoring on daily basis. The authors proposes a solution addressing this issue by providing network-less way to monitor virtual machines operating system with agent running inside.

Chapter 7 presents an approach to robotic middleware as a tool to facilitate the programming and testing of complex tasks for autonomous mobile robot with a very limited hardware resources.

Chapter 8 describes different approaches to code generation. The authors propose an approach to reuse a code originated from the previous project in a next development stage. The criteria were evaluated in experiments on UML models transformed into C# programs,

Chapter 9 shows examples of human behavior impact on data protection. The author determines a set of different sources of human negative impact related to such security areas as: general models, risk analysis, authentication with special emphasis on some innovative methods and tools.

Chapter 10 discusses the role of color and a graphical layout in interactive digital signage software.

Chapter 11 addresses automatic recommendation system and proposes a hybrid filtering method in, formed as a combination of collaborative filtering and content based filtering.

Chapter 12 deals with the problem of summary generation and presents a two-stage approach to knowledge integration and focuses on a translation of an output of the integration process into linguistic form.

Chapter 13 presents text document categorization method based on word frequent sequence mining. The proposed document classification method is evaluated by experiments carried out for the Reuters document collection.

Chapter 14 describes issues related to the themes of ontology, which will allow to understand the causes of problem of inconsistency of knowledge.

Chapter 15 presents an overview of data integration problem by providing a general definition to this problem, and classifying its different approaches that are proposed by the IT community.

Chapter 16 discusses and evaluates various software multi-disk storage configuration in GNU/Linux operating system to support more effective management of system resources.

I would like to sincerely thank the contributors of each chapter. The book would not be possible to be edited without their great contributions. Hopefully these research works and related reviews give readers a comprehensive representation of recent developments in design and evaluation of information systems.

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

Wroclaw, September 2014

Leszek Borzemski

risk assessment, risk measures, financial risk, banking,
MST, value at risk, standard deviation

Anna KAMIŃSKA*
Anna KIŁYK*
Agnieszka PARKITNA*

STATISTICAL RISK ASSESSMENT OF WIG-BANK INDEX DURING YEARS 2008–2012

The purpose of this article is to indicate accurate method for selecting beneficial shares on stock market that promise the smallest loss, and achieve satisfaction on profits. Financial risk assessment is a high priority topic for investors. In article are shown the most popular methods and their advantages and disadvantages. This work attempts to use standard risk measures (standard deviation and VaR) and MST for the analysis of quoted securities. Case study is based on Polish WIG-BANK index.

1. FINANCIAL RISK ASSESSEMENT

1.1. CRITERIA AND IMPORTANCE OF RISK ASSESSEMENT METHODS

Number of factors can determine selection of the appropriate risk management method. Ostrowska in her own research indicates as primary factor the availability of information and its variation over time. When we consider the investment risk we must be aware about existing of investment conditions and the time horizon of the investment.
It is required, that the team, which is carrying out a risk assessment posses knowledge about methods and is skilled in their application, as well as have knowledge and experience- especially when estimating the scale and probability of events that have an impact on the risk (with the workload and cost of data efficiency). It became necessary to be awarded that there are tendencies of investors to make decisions in different

* Institute of Organization and Mangement, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław.

conditions and economic conditions that affects the assessment of risk. Entire process needs to be supported by computer programs [8].

Proper selection of risk assessment method doesn't guarantee its effectiveness, but it allows do make freer choice of investment with high income and low risk. Market as wide concept is like a living organism. It is variable and sensitive to the environment, and therefore the risk is a phenomenon which includes uncertainty factor, and only estimation of the chance of failure can make possible to prevent it [2].

1.2. PROPERTIES OF MEASURES OF FINANCIAL RISK ASSESSMENT

Term of "risk ratio" can be used whenever it provides necessary axioms. For companies consequences of business activity are mainly financial dimension. Therefore we assume, that X can represent:

– change of market or balance value, market position in absolute terms or relative,

– the result obtained on invested capital (profit or loss),

– cumulative losses for a portfolio of different types of risk.

The following axioms includes the assumption that the random variable is expressed in absolute value.

When we mark "V" as a set of random variables, which is taking real values defined on the probability space (Ω, F, P). In this case, risk measurement is determinate by relation between V and a non-negative real number:

$$\rho: V \rightarrow R^{+} \tag{1}$$

This assumption allows for easy submission and comparison of investment in terms of risk. It cannot be any function, because there have to be fulfilled additional conditions for ensure, that any will appear. These conditions are as follows:

1. Relevance – for each:

$$X \in V, \rho(X) \geq 0 \tag{2}$$

Risk is the measure of the level of deviation, therefore, is a requirement of unnegativity natural (when $\rho(X) = 0$ is the risk of a variable constant).

2. Subadditivity – for each

$$X_1 \; and \; X_2 \; \epsilon \; V, \rho(X_1 + X_2) \leq \rho(X_1) + \rho(X_2) \tag{3}$$

The sum of individual risks do not exceed the total level of risk (to merge the various types of risks not create additional risk)

3. Positive homogeneity – for every $\lambda \geq 0$, and each

$$X \; \epsilon \; V, \rho(\lambda X) = \lambda \, \rho(X) \tag{4}$$

This is partly a consequence of Subadditivity.

4. Invariance – for each $X \in V$ and for each $\alpha \in R$, there

$$\rho(X + \alpha) \leq \rho(X) \tag{5}$$

This is partially consequence, that index will not change if we add a constant α to a random variable.

5. Monotonicity – for each $X_1 \; and \; X_2 \in$ V, if only $X_1 \leq X_2$:

$$\rho(X_1) \leq \rho(X_2) \tag{6}$$

This means that when a change in the portfolio decreases systematically X_1, X_2 than the losses of the portfolio is the risk that first needs to be smaller.

6. Invariance because of the translation for each $X \in V$, and for each $\alpha \in R$ occurs

$$\rho(X + \alpha(1 + \gamma)) \leq \rho(X) - \alpha \tag{7}$$

where γ is the risk-free rate of interest. If we add a deterministic profit for the random variable$X$, then the risk will be reduced byα.

7. Responsiveness – $\rho(X) > 0$ if $X \leq 0$ and $X \neq 0$ - ensures that the risk measure identifies a random portfolio as risky.

If risk measure complies axioms 2, 3, 5 and 6 is called a coherent risk ratio. An additional feature is the protrusion (resulting from axioms 2 and 3). A measure of convex stochastic dominance of II row, that means that diversification reduces risk, which determinate its significance for the optimization of the investment portfolio.

Numerical development of risk optimization problem requires that local minima correspond to the global, and the property are convex functions. Each function subadditive and positively homogeneous is convex, which means that each risk measure is convex risk measure (but not vice versa). Coherent Measures can be used by supervisors in the context of the minimum capital needed to cover losses from risk.

Presented approach can be extended to generalized coherent risk measure where specified risk measure $\rho: V \rightarrow R^+$, the set H is as follows:

$$H\rho \equiv \{X \in V: \rho(X) = 0\} \tag{8}$$

It is the indication of the next axiom:

8. The shortest distance for each $X \in V$ are $X^* \in H$, so that it is a point of the shortest distance X and the set $H\rho$ of $0 \leq \alpha \leq \rho(X)$ we have

$$\rho(X + \alpha\mu) = \rho(X) - \alpha \tag{9}$$

where u is a unit vector, such that

$$u = \frac{(X * (-X))}{\|X * (-X)\|} \tag{10}$$

Generalized measure of risk satisfies axioms 2, 3, 5, 7, and 8 key difference here is in the axiom 8, where assets are added to the portfolio $\alpha\mu$ not have to be the capital, but a possible risky investment.

Axiom, which should still indicate that:

9. limiting the expected value $\rho(X) \geq E(X)$ for each non-permanent X and $\rho(X) = E(X)$ for each fixed X.

The measure, which satisfies the conditions (2, 3, 6 and 9) is called the expected limited value. When the conditions are fulfilled or axiom 5 is called a measure of the measure of limited coherent expected value.

Mentioned axioms are not the only, but the most popular. Nevertheless, the fulfillment of all of them is very difficult, especially subaddictiveness [1], [5], [11].

2. METHODS OF RISK ASSESSMENT

The phenomenon of risk can be measured in many ways. To select appropriate measurement method it is necessary to systematize the knowledge of the available methods of measurement and assessment of risk. The following table shows the most common classifications of risk measurement methods.

Table 1. Risk assessment methods

Autor	Clasification
E. Ostrowska, 2002; D. Skorupka, 2012	1) Correcting the efficiency of the investment project: Adjustment of the discount rate, Equivalent assurance 2) Calculation of sensitivity: Break-even investment, Return on capital, 3) Probabilistic statistics: A formula with variables independent or dependent, Decision tree 4) Simulation: The parameters of the simulation, A series of simulations, 5) Operational Research. Strategy games: Formula maksymin,Minimax formula.
K. Marcinek and others, 2010	1) Indirect methods: Sensitivity analysis, Scenario analysis, Statistical analysis (probabilistic methods) , Standard deviation and coefficient of variation, Simulation analysis as a method for estimating the expected value and standard deviation, 2) Direct methods: Limit the period of repayment, Equivalent assurance and Discount rate of risk 3) The methods of operations research: Methods Network (critical path), The analytical hierarchy process (AHP), Method ELECTRE, Linear and Dynamic programming.

K. Kuziak, 2011	The breakdown by type of risk in the company: 1) measures of market risk: Quantile distribution of the risk factor, Valuation model (the classic models of financial econometrics- ARIMA and GARCH models of stochastic differential equations) 2) Operational risk and operational tasks: Top-down and Bottom-up approach, 3) Credit risk: The probability of default, Loss in case of default, The value of the contract threatened breach of conditions, CreditVaR, Correlation and concentration of the portfolio, Raiting credit (such as Credit Risk Plus model).

Source: Own research based on [5], [6], [8], [9].

Methods pointed out in table, regardless of the classification have advantages and disadvantages.

The most commonly encountered method is **a sensitivity analysis**. It allows to define these variables, on which the Net Present Value (NPV) is the most sensitive. It helps to identify the factors that contribute to the collapse of the project. However, it requires that each key variable suppose to be isolated, and it does not allow for the accumulation of the effects it produces two or more factors simultaneously. **Standard deviation**, the likelihood and rate of change for high reliability design, give the results of the likelihood of obtaining a certain level of income during the period. It may happen, however, that these results are not entirely true. **Simulation methods** allow managers to check the different combinations of events, and to investigate the relation between factors affecting cash flow, which gives a better understanding of the whole project. **Equivalent of certainty** is almost foolproof way, because it gives the most probable value of the cash flows. Sometimes, however it's hard to determine the level of reliability of the method, because it can understate the value of NPV. **Method of risk-adjusted discount rate** is based on correcting ingested discount rate for the calculation of faithfully reflect the risk of the project. However, there is a risk that the beta of a new project, in practice, may be inadequate to the average for the group of companies in a particular industry or market [4] [10].

All these methods are methods of risk assessment. When they are used properly, they will help to minimize the cost of risk. Such costs are divided into two groups. First, the costs arising from the occurrence of the risks that arise at the time of the negative deviations of risk factors, and that in the absence of security is directly reflected in the results of operations of the company. The second group are the costs of uncertainty, such as the cost of hedging against risk and cost inefficiencies caused by the exposure. Often in economic reality seeks to too large hedge future economic conditions. These costs show up in improperly invested enterprise resources in a manner which does not allow for optimal utilization [7] [9].

3. ANALYSIS OF WIG-BANK

The sample is index WIG-BANKI in Polish stock market. As the name suggests it contains a company qualified for Bank sector. They are: PKOBP, PEKAO, BZWBK, MBANK, HANDLOWY, INGBSK, GETINOBLE, MILLENNIUM, ALIOR, GETIN, UNICREDIT, BANKBPH and BOS. Number of banks often change although it fluctuates around 14. WIG-BANK is a sub-index and that means that this factor is a total return index and for its calculation takes into account both the price of underlying shares and the income from dividends and rights issues. It contains the same packages as in the WIG index portfolio. The base date of the index is December 31, 1998, with the value at the date of 1279.56 points. Methodology sub-index is the same as the WIG index.

Selection of WIG-BANKI index was decided by its thriving from September 2013. This phenomenon is important because of fact, that banks is strong on the Stock Exchange, due to their high share and stock market indices. There is no growth in this sector without changes in WIG20 and WIG. Chart covering the period from 01.12.2012 to 01.12.2013 is presented below.

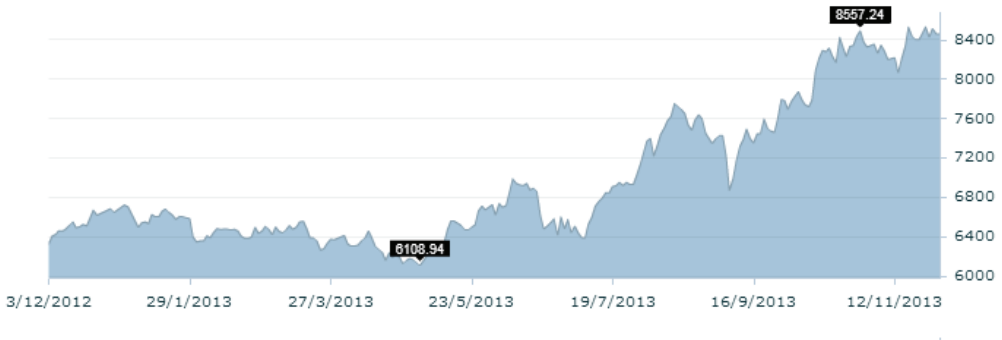

Fig. 1. Quotations WIG-BANKI for the period from 01.12.2012 to 01.12.2013
Source: http://www.gpw.pl/ dated 05.12.2013

Analysis of the index will be carried out using an algorithm Minimal Spanning Tree (MST).

3.1. RESEARCH

The aim of first method is to investigate the relationship and possible changes for individual stocks listed in the analyzed index. The basic element of this test method is standardized and distance analysis of aligned vectors twists on a pre-determined time window [3], indicating the smallest distance between successive pairs of values. By analyzing the return vectors to investigate relationships between the analyzed values, we can better manage investment portfolio. In a situation where the stock exchange is

stable such test is able to identify a set of assets that exhibit similar behavior. The problem, however, comes at a time when the listed values start to behave erratically. It is difficult to predict the future value of trading in such a situation it is better to choose stocks of companies with the weakest relationships, and thus the highest correlations. Such behavior will potentially reduce the possible loss:

$$d_{ab} = \sqrt{2(1 - C_{ab})} \tag{11}$$

where:

a, b – more banks,

d_{ab} – distance between another pair of values,

C_{ab} – coefficient of correlation between a pair of test values.

The study carried out by this method will be used two time intervals, characterized by large fluctuations in prices in the studied intervals, and in addition (which will be presented in the drawings) is possible due to the construction of two different graphs MST:

- 01.10.2008–01.11.2009,
- 01.06.2011–01.06.2012.

Starting with an analysis of the earliest period (01.10.2008–01.11.2009) it can be seen that the graph for WIG-BANKI characterized by a large spread. It should be noted also that in the presented tree only two company have more than two neighbors (GETIN and MBANK), which may indicate a strongly correlated relations companies included in the index.

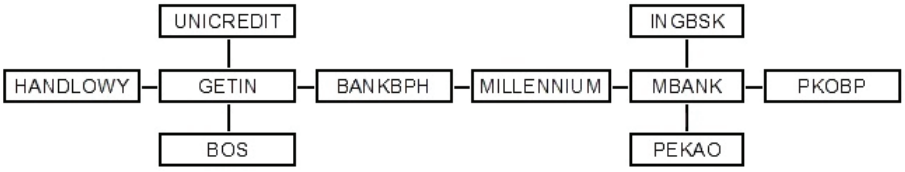

Fig. 2. MST WIG-BANKS for the period: 01.10.2008 - 01.11.2009.
Source: Own research

A similar situation can be observed by analyzing the next time interval (01.06.2011–01.06.2012), for which the tree relationship is more compressed and is characterized by one large cluster. Although both graphs present periods distant from each other can be observed further term relationships between some of the banks, even MILLENNIUM and BANKBPH, GETIN and UNICREDIT or INGBSK and MBANK.

Next step of the analysis will be obtained compared to the distance in the graph MST with standard risk measures which are the standard deviation and VaR (Figure 1 and Figure 2). Although both measures have some drawbacks, it is worth noting that

Fig. 3. MST WIG-BANKI for the period: 01.06.2011–01.06.2012.
Source: own research

they are recognized as international standards for risk assessment, which further argues for their use when testing the relationship between banks belonging to the WIG-BANKI. While in this study, the standard deviation has its classic form, and says the size of the existing fluctuation, it used VaR will not examine the value of capital at risk, and the percentage of capital at risk. For this purpose, will be used variations-covariance approach modified to the following formula:

$$VaR_N = k\sigma_N - \mu_N \tag{12}$$

where:

k = 1.65 The value of the confidence level of 95%;

N – test bank

b – analyzed the risk of the bank,

n – the return of the bank recorded.

Additionally authors used return on the share price of banks, in order to get a rid of the noise that would interfere unnecessarily analyzed signal (time series).

Table 2. Summary of risk measures for the studied banks and their distance from the neighbors for the period: 2008 to 2009

Bank 1			Distance MST	Bank 2		
Standard deviation	VaR	Bank		Bank	Standard deviation	VaR
3,58	5,81	GETIN	0,297	HANDLOWY	2,95	4,83
3,58	5,81	GETIN	0,327	UNICREDIT	4,00	6,52
4,57	7,37	MBANK	0,361	MILLENNIUM	4,41	7,07
4,57	7,37	MBANK	0,367	PEKAO	4,24	6,87
4,57	7,37	MBANK	0,370	INGBSK	3,29	5,47
3,58	5,81	GETIN	0,402	BANKBPH	3,95	6,44
4,41	7,07	MILLENNIUM	0,408	BANKBPH	3,95	6,44
4,57	7,37	MBANK	0,491	PKOBP	3,51	5,65
3,58	5,81	GETIN	0,879	BOS	3,26	5,30

Source: Own research

Comparing the results for the first test period shows, that the greatest correlation, and hence the shortest distance MST are primarily central indexes (GETIN, MBANK). In addition to the first pair of values (GETIN, HANDLOWY), the remaining five consecutive pairs have a large ratio of the standard deviation and VaR. In addition, these couples have a low rate of MST distances (less than or around 0,400) mainly reflecting a high correlation coefficient. As already mentioned exception of the first pair of values (GETIN, HANDLOWY) have the lowest rate of MST distances at the same time the low value of the standard deviation of the Handlowy Bank. The emergence of such a situation leads us to believe that it is possible existence of additional relationships between these two values, which can be immeasurable (holding a share of the second bank).

Unlike banks present in the other analyzed time period (2011–2012), in which the shortest distance in the graph MST have a company with the largest standard deviation and VaR. However, in this case there is an exception (MBANK, HANDLOWY) pair of values, which in spite of the high correlation coefficient (a small distance values MST) has a relatively low standard deviation and low VaR values.

Table 3. Summary of risk measures for the studied banks and their distance from the neighbors for the period: 2011 to 2012

Bank 1			**Distance MST**	**Bank 2**		
Standard deviation .	**VaR**	**Bank**		**Bank**	**Standard deviation**	**VaR**
8,60	15,16	UNICREDIT	0,295	GETIN	10,65	18,70
6,88	12,47	NOVAKBM	0,373	GETIN	10,65	18,70
6,88	12,47	NOVAKBM	0,382	BOS	3,03	5,39
6,88	12,47	NOVAKBM	0,409	HANDLOWY	2,60	4,44
2,55	4,42	BANKPBH	0,420	MILLENNIUM	2,80	4,85
6,88	12,47	NOVAKBM	0,445	BGZ	2,77	4,55
2,56	4,35	MBANK	0,453	HANDLOWY	2,60	4,44
6,88	12,47	NOVAKBM	0,468	PKOBP	2,17	3,73
2,55	4,42	BANKPBH	0,468	PKOBP	2,17	3,73
2,56	4,35	MBANK	0,482	PEKAO	2,50	4,23
2,56	4,35	MBANK	0,582	INGBANK	2,04	3,43

Source: Own research

4. SUMMARY

As it could be observed in studies in most cases low distance value pairs MST banks corresponded to high-risk values that were measured standard deviation and

VaR. This suggests that the method of MST may be a good method to study the investment risk. On the other hand, it should be noted that due to the method of MST studies are also subjected to immeasurable elements of the relationship between pairs of values, such as joint investments, stock ownership of the second asset, etc. For this reason, it can be assumed that using this method, along with a thorough analysis of the companies (e.g. using fundamental analysis) you can attempt to create a diversified investment portfolio.

REFERENCES

[1] ARTZNER P., DELBAEN F., EBER J.-M., HEATH D., *Coherent Measures of Risk,* Mathematical Finance, 9, 1998, 203–228.
[2] HULL J.C. *Zarządzanie ryzykiem instytucji finansowych*, Wydawnictwa Profesjonalne PWR, Warszawa 2011.
[3] KIŁYK A., WILIMOWSKA Z., *Minimal Spanning tree of the Foreign Exchange Market (FOREX), ISTA 2011.*
[4] KOSTENOK I.V., FIL E.V., *Risk measuring methods: modern approaches*, Economic Herald of the Donbas, no 4 (26), 2011.
[5] KUZIAK K., *Pomiar ryzyka przedsiębiorstwa: modele pomiaru i ich ryzyko,* Wydawnictwo Uniwersytetu Ekonomicznego, Wrocław 2011.
[6] MARCINEK K. [et al.], *Ryzyko w finansowej ocenie projektów inwestycyjnych: wybrane zagadnienia,* Wydawnictwo Uniwersytetu Ekonomicznego, Katowice 2010.
[7] MICHALSKI D., KRYSTA B., *Nowoczesne narzędzia kontroli zarządzania, Wybrane zagadnienia: od rachunku odpowiedzialności i zysku ekonomicznego po controlling ryzyka,* Instytut Doskonalenia Wiedzy o Rynku Energii, Warszawa 2006.
[8] OSTROWSKA E., *Ryzyko projektów inwestycyjnych*, Polskie Wydawnictwo Ekonomiczne, Warszawa 2002.
[9] SKORUPKA D., KUCHTA D., GÓRSKI M., *Zarządzanie ryzykiem w projekcie*, Wyższa Szkoła Oficerska Wojsk Lądowych im. generała Tadeusz Kościuszki, Wrocław 2012.
[10] SKORUPKA D. (ed.), GOLONKA A., KAMIŃSKA A. *Risk management and stress in the project* Scientific problems in Management*,* General Kościuszko Military Academy of Land Forces in Wrocław, Wrocław 2013:, 97–108.
[11] TRZPIOT G. (red. nauk.), *Wielowymiarowe metody statystyczne w analizie ryzyka inwestycyjnego*, Polskie Wydawnictwo Ekonomiczne, Warszawa 2010.

performance testing,
PlanetLab,
web servers

Anna KAMIŃSKA-CHUCHMAŁA*,
Michał SCHATT*

RESEARCH AND MEASUREMENTS OF WEB SERVERS PERFORMANCE IN INTERNET NETWORK USING PLANETLAB ENVIRONMENT

This paper is written about important and well known topic as is measuring performance of Web servers. The studies on performance of Web servers and whole Internet network are under way for many years. Tests were performed on different levels, in example one measures were performed only in local area networks, others were performed between nodes on different continents. These research were carried out on nodes which belonging to the PlanetLab. This is an open platform for developing, deploying and accessing planetary-scale services. Nodes used to perform research were working under MyPLC software, which is a specially prepared Linux operating system. Ten measuring agents once per hour were downloading file from twelve mirror servers of Linux Gentoo operating system saving parameters of downloading: speed and time of downloading, time of response from server. The studies lasted four weeks and brings a lot of data to perform analysis. To collect downloading data standard software was used to trace packets in the Internet, and downloading files from servers. The results were divided for analysis into three regions according to the geographic location of agents: agents from America, agents form Europe and Israel, agents form Asia and Australia. The final conclusions from experiment ending the paper.

1. INTRODUCTION

Infrastructure of World Wide Web could be seen as a large, artificial ecosystem. The endpoints computers are like a little cells of body and veins deliver to them everything that they need. In the Internet environment the little veins are connections from local Internet Service Provider (ISP). ISPs are connected to the other, bigger

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

Internet Providers, these companies are delivering Internet for countries and regions. The global computer net is very complicated, it consists of many big companies controlling, developing, and maintaining Internet infrastructure. New connections between backbone nodes are setting all the time. Those are optical fiber, which can transmit data on long distances with high speed. New services of Internet such as video and audio streaming, e-learning and sending Voice over IP (VoIP) need high speed structure to ensure proper quality of services. Moreover Internet is more and more popular, so new users are beginning to use internet what adds to the net more load.

Nowadays Internet became important part of our life. Performance of web servers is crucial for working the whole Internet. Web servers are hosting websites, databases and web applications. If performance of web server is below some kind of threshold, user resigns to use this services. Important thing to the IT companies is to have servers in best available condition, because this will bring them new clients. Dissatisfied clients will be gone to the other companies. Effective measurements of web servers performance will allow us to know better structure of the Internet, packet routing traces, load distribution of the net. All this actions will let us predict some performance issues and help to prevent lack of access to the server resources.

2. RELATED WORK

The measurements of web servers performance could be divided into two methods: active and passive measurements. There is a lot of publications about both methods. In related work we present some of finished researches based on active methods.

In article [2] authors focused on performance of data transmission in Internet. Experiments were performed using MWING measurement system. Agents of this system were located in four cities: Los Angeles, Gliwice, Gdansk and Wroclaw. Research was started in 2008 and lasted a year.

Authors of article [3] were checking performance of three types of Internet network. Using active and passive methods researchers were testing performance and searching bottlenecks of those networks.

Researchers from Greece in article [6] wrote about new framework based on SMNP (Simple Managing Network Protocol) which allows to monitor and perform active measurements on DiffServ networks. Measure is based on “SAM” (SMNP for Active Measurement) agent running on routers in tested networks.

3. PLANETLAB ENVIRONMENT

PlanetLab is a worldwide research net of connected servers which support developing new Internet services. From 2003 almost 1000 universities, and industry companies were using PlanetLab to develop, deploy and access planetary-scale services.

Now PlanetLab have 1180 nodes in 563 places (please see figure 1.).

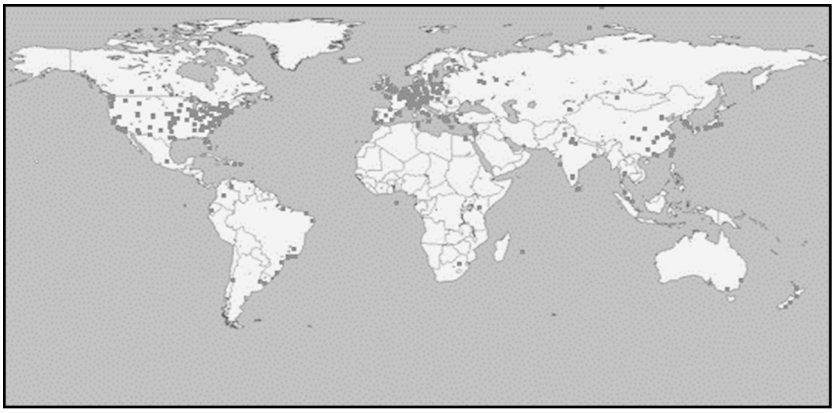

Fig. 1. Location of PlanetLab nodes [8]

PlanetLab is a set of machines distributed over the globe. Most of machines are located in academic institutions or routing centres. All of them are connected to Internet to manage updates and create some kind of "micro cosmos" to test behaviour of application. Creators wish was to "grow to 1,000 widely distributed nodes that peer with the majority of the Internet's regional and long-haul backbones" [6].

All PlanetLab machines run a specially prepared software package based on a Linux operating system (Fedora 8). This software deliver base operating system, mechanisms of bootstrapping nodes and distributing software updates, a collection of management tools that monitor node health, audit system activity, and control system parameters; and a facility for managing user accounts and distributing keys.

One of main purposes is to serve as a testbed environment. Research groups can require a slice of resources in which they can carry out various experiments in example: file sharing and network-embedded storage, content distribution networks, routing and multicast overlays, QoS overlays and network measurements. According to official website, there are currently over 600 active research projects running on PlanetLab.

PlanetLab is also designed to support long-running services ran by biggest ICT corporations: Intel, Hewlett Packard, Google, AT&T, and France Telecom. Those companies are using PlanetLab environment to check behaviour of new software in

large set of geographically distributed machines, realistic network substrate that experiences congestion, failures, and diverse link behaviours and the potential for a realistic client workload.

4. RESEARCH

Research were operated on ten PlanetLab nodes. The single test begins with checking route to the Gentoo servers using popular command *tracert*. This part of research brought information about route of packets and time of response the closest server to destination. After that PlanetLab nodes once in an hour were downloading file from all Gentoo mirror servers. All information from checking routes and downloading files were saved to the log files. Table below presents location of servers with downloadable files.

Table 1. List of Gentoo mirror servers

	Country	**City**	**Hostname**	**Administrator**
1	Australia	Melbourne	http://ftp.swin.edu.au/gentoo	Swinburne University of Technology
2	Brazil	Curitiba	http://gentoo.c3sl.ufpr.br/	C3SL, Federal University of Paraná
3	Canada	Waterloo	http://mirror.csclub.uwaterloo.ca/gentoo-distfiles/	University of Waterloo
4	China	Xianmen	http://mirrors.xmu.edu.cn/gentoo	Xianmen University
5	England	London	http://www.mirrorservice.org/sites/distfiles.gentoo.org/	The UK mirror service
6	Greek	Athens	http://files.gentoo.gr/	--
7	Israel	TelAwiw-Yaffa	http://mirror.isoc.org.il/pub/gentoo/	Hamakor FOSS Society
8	Japan	Nomi	http://ftp.jaist.ac.jp/pub/Linux/Gentoo/	Japan Advanced Institute of Science and Technology
9	Portugal	Coimbra	http://ftp.dei.uc.pt/pub/linux/gentoo/	University of Coimbra
10	Sweden	Lund	http://ftp.df.lth.se/pub/gentoo/	Lund University
11	Ukraine	Kiev	http://gentoo.kiev.ua/ftp/	--
12	USA	Atlanta	http://www.gtlib.gatech.edu/pub/gentoo	Georgia Tech

Table 2 presents parameters of file which was downloading.

Table 2. Parameters of downloading file.

Filename	livedvd-x86-amd64-32ul-11.0.iso.CONTENTS-squashfs.gz
Path	/releases/x86/11.0/
File type	Compressed text file
Size	4,4 MB

Mirror servers were located in North America (two servers), South America (one server), Europe (5 servers), Asia (three servers), Australia (one server). Please see figure 2.

Fig. 2. Locations of Gentoo mirror servers

Table 3 presents list of PlanetLab nodes use in research. One node was located in North America, four in Europe, two in Asia, one in Africa and Australia. Geographic location is showed in figure 3.

Table 3. List of PlanetLab nodes used in slice.

	Country	**City**	**Hostname**	**Administrator**
1	Australia	Melbourne	planetlab2.ru.is	PlanetLab Central
2	Czech Republic	Prague	ple2.cesnet.cz	PlanetLab Europe
3	England	Cambridge	planetlab2.xeno.cl.cam.ac.uk	PlanetLab Europe
4	France	Saint-Denis (Reunion Island)	lim-planetlab-1.univ-reunion.fr	Planet Lab Europe
5	Israel	Ramat Aviv	planetlab2.tau.ac.il	PlanetLab Europe
6	Norway	Oslo	planetlab1.ifi.uio.no	PlanetLab Europe
7	Poland	Gliwice	plab4.ple.silweb.pl	PlanetLab Europe
8	Poland	Wroclaw	planetlab1.ci.pwr.wroc.pl	PlanetLab Europe
9	Thailand	Bangkok	ple1.ait.ac.th	PlanetLab Europe
10	USA	Piscataway	planetlab1.rutgers.edu	PlanetLab Central

Fig. 3. Localization of PlanetLab nodes used to perform research

5. RESULTS

Research was performed from 12th of April to 10th of May 2014 on ten nodes PlanetLab network. Most of nodes were operating normally during research. Due to technical problems three of them (USA, Israel and Reunion) were down for few days. Size of all logs was about 700 MB.

Results were divided into three geographical parts – servers from America, Europe (with Israel) and Asia-Australia.

Download speed was high when file was downloaded from the same continent. Servers were quite close to each other. The most stable connection was to Curitiba in Brazil – on this link was the smallest deviation from average download speed. (fig 4.) After 2nd of May the average download speed from Canada and USA slightly decreased. Across American continents average download speed was about 3243 kBps, where speed across European continent was 874 kBps.

In these three cases average Round Trip Time (RTT) was between 31 ms from Canada to 161 ms from Brazil. The highest speed, and lowest RTT were from the closest server – server from Canada (fig. 5.).

Downloading file from other continent took more time to complete. Also connections were more unstable. Sometimes downloading file was corrupted and started from beginning, sometimes permanently cancelled. Average download speed was between 162–189 kBps (fig. 6.).

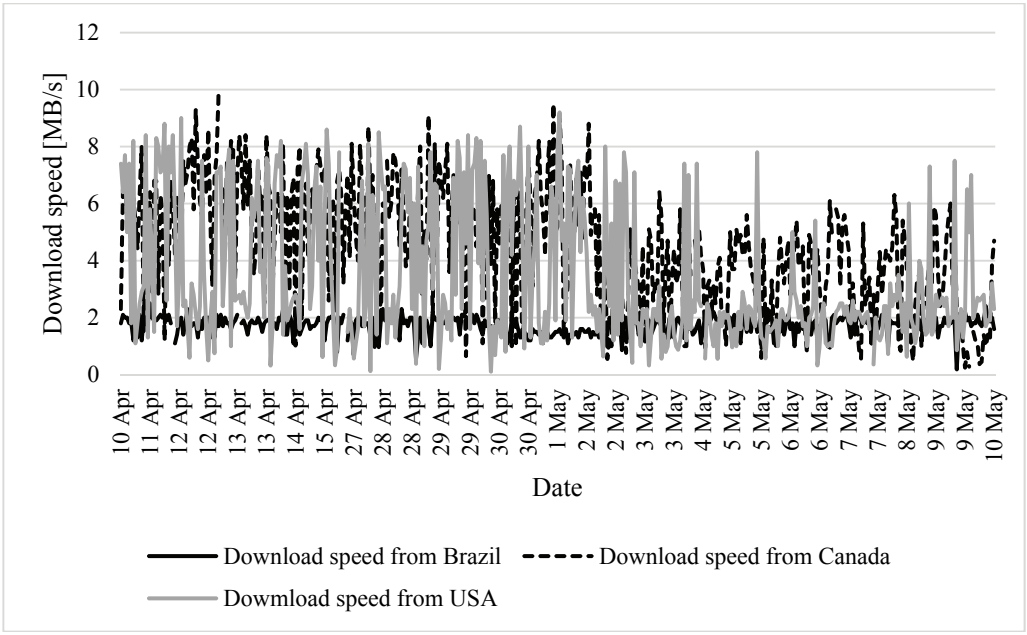

Fig. 4. Download speed from servers in America to PlanetLab node in Piscataway (USA)

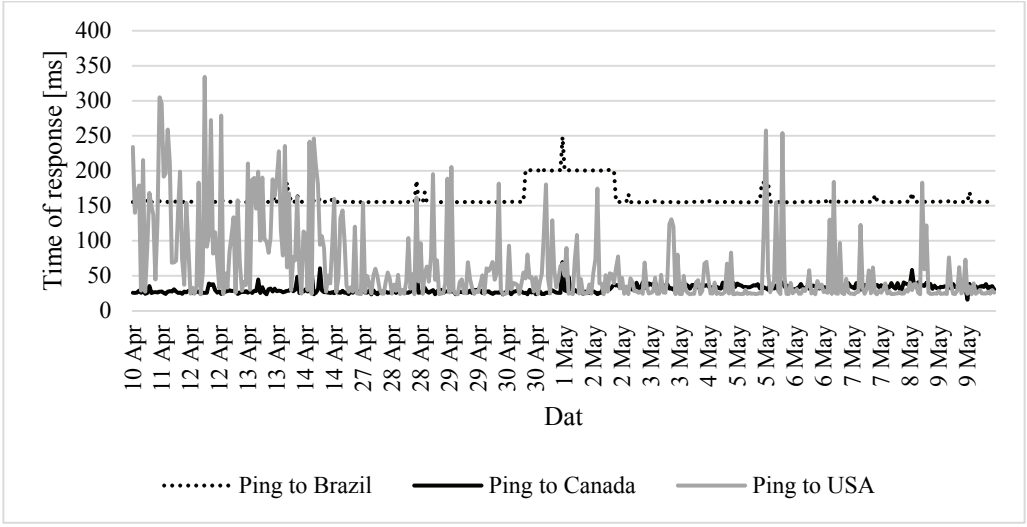

Fig. 5. Time of response servers in America to PlanetLab node in Piscataway (USA)

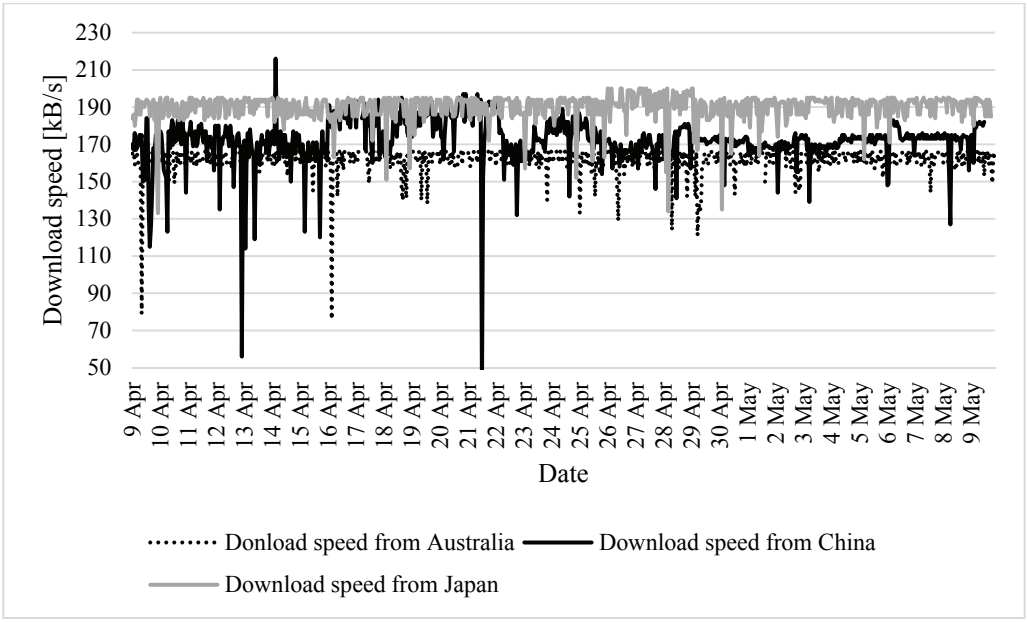

Fig. 6. Download speed from servers in Asia and Australia to PlanetLab in Wroclaw

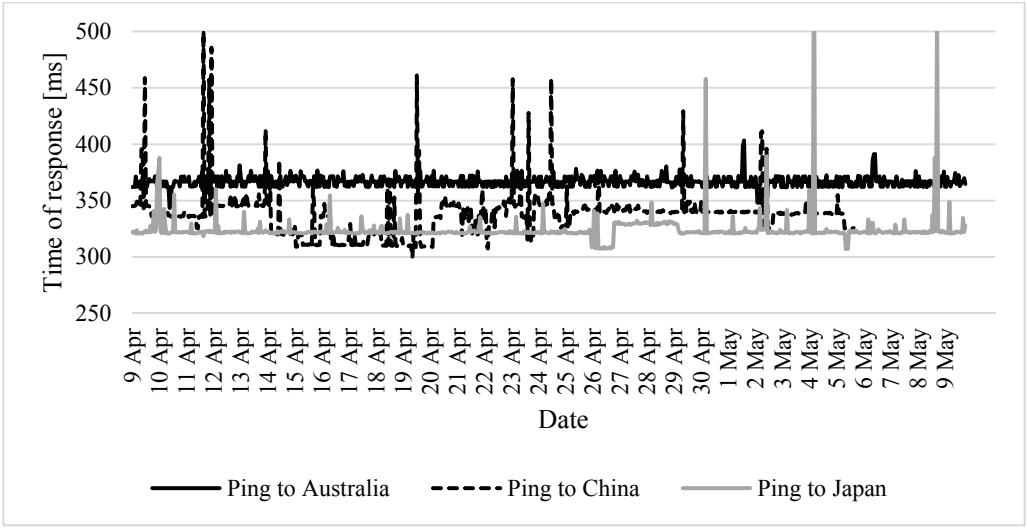

Fig. 7. Time of response servers in Asia and Australia to PlanetLab node in Gliwice

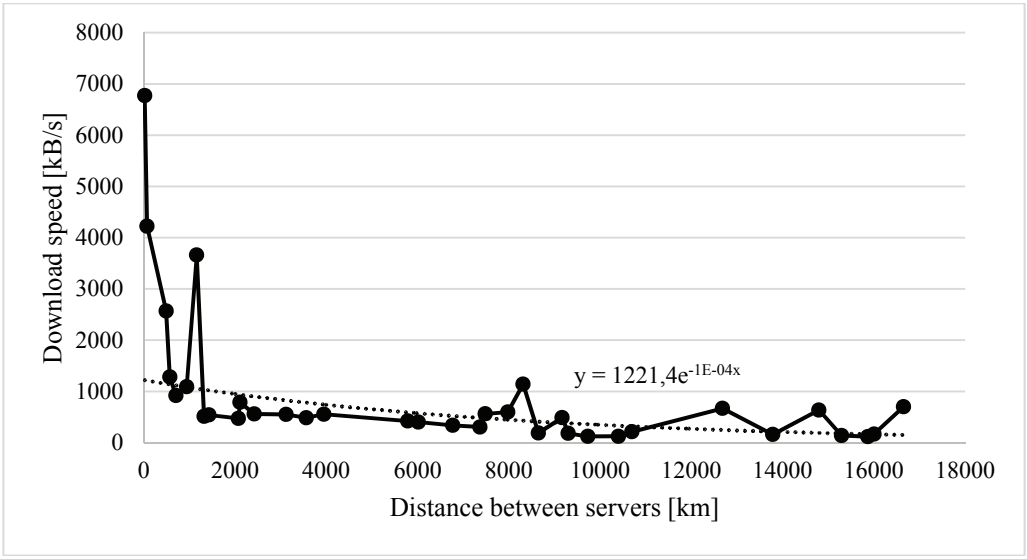

Fig. 8. Change of speed in relation to the distance between servers

From three servers far away from PlanetLab node RTTs are similar. The lowest average RTT was from server in Japan, the highest from China. Because of a lot of nodes in China Internet infrastructure RTT was also high during testing performance servers in Asia (fig. 7.).

Download speed between servers close to each other are faster than in case that distance between servers is bigger (fig. 8.). Fast change of information occurs in servers with short distance between them. On graph we can see some deviations from trend line. This situation can happen from multiple causes beginning from faster network equipment, ending on better quality of medium transmission.

6. CONCLUSIONS

Gentoo mirror servers were performing normally during experiment. Eight of eleven PlanetLab nodes also were working all the time. Node located on Reunion Island was performing normally to 1st of May, after that technical problems were increasing and results from this node was incomplete.

Speed of downloading files was depending on geographical distance between servers. The closer is server, the file is faster downloaded. In some cases time of day was also important. Downloading from England mirror server was the fastest in early morning: between 5 AM to 9 PM. In experiment time of downloading was checked twice: first time was including time consumed for resolving domain name server and

downloading, second time was time consumed for downloading file. Research showed that differences between both times were very small, what confirmed that Gentoo mirror servers were performing normally, and were not very busy.

RTT between servers on the same continents were between 100-150 ms, while RTT to other continents were about 250–300 ms. Measuring time of response confirmed statement that closer servers response faster. Number of single hops in route from source to destination is depending how complicated Internet infrastructure is in countries on the road of current packet.

PlanetLab environment is very good to perform variety of experiments including measurement of web servers performance. Using nodes with preinstalled operating system speeds up starting of experiments and allow researchers to focus on problem. Four weeks period of research showed that Gentoo mirror servers are performing very well, access to resources is quick and can sent files to node thousands kilometres away with acceptable speed.

REFERENCESS

[1] BROWNLEE N., LOOSLEY C., *Fundamentals of Internet Measurement: A Tutorial*, Keynote Systems 2001.
[2] BORZEMSKI L, NOWAK Z. *Empirical web performance evaluation with using a MWING system*, In: Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence. 2007, vol. 4496, s. 278-287.
[3] CALYAM P., KRYMSKIY D., SRIDHARAN M., SCHOPIS P. *Active and Passive Measurements on Campus, Regional and National Network Backbone Paths*, The Ohio State University, Columbus, Ohio, 2005.
[4] CLAFFY K., FOMENKOV M., KEYS K., KRIOUKOV D., HYUN Y., *Internet Mapping: from Art to Science*, San Diego Supercomputer Center, University of California San Diego, 2008.
[5] HAQUE M, PAWLIKOWSKI K., *Step By Step Guide for Using PlanetLab*, The Department of Computer Science and Software Engineering, University of Canterbury, 2011.
[6] KOURTIS A., SARSEMBAGIEVA K., GARDIKIS G., XILOURIS G., *SNMP-driven Active Measurements in DiffServ Networks*, Institute of Informatics and Telecommunications, Grece, 2013.
[7] www.planetlab.org
[8] www.planetlab.eu

Internet, autonomous systems,
grouping, classification.

Anton PETRAKIVSKYI*
Ziemowit NOWAK*

GROUPING METHODOLOGIES OF AUTONOMOUS SYSTEMS ON THE INTERNET

A review of grouping methodologies of autonomous systems on the Internet was concluded. One of the methodologies has been developed and implemented. Nearly 38 thousands active autonomous systems have been classified during the study.

1. INTRODUCTION

Attempts to group autonomous systems have been carried out for several years. They are based on the properties of connections obtained from the graph of connections between autonomous systems, built on the basis of the information inferred from BGP tables.

Govindan and Reddy (1997) classified autonomous systems into four levels according to their degree of dispersion [1]. In 2001, Ge, Figueiredo, Jaiswal and Gao classified the autonomous systems to seven levels, based on inferred relations (such as customer-supplier) between autonomous systems [2]. That method was based on the idea that the autonomous systems that provide Internet access services must be on a higher level than their customers. In 2002, Subramanian, Agarwal, Rexford and Katz classified autonomous systems into five levels, based on the inferred customer-supplier relations and peer-to-peer communication between them [3].

According to the authors, the most hitherto interesting way of classifying autonomous systems was the one suggested in 2006 by Dimitropoulos, Krioukov, Riley and

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

Claffy of CAIDA (Cooperative Association for Internet Data Analysis) [4]. They used a machine learning algorithm, based on a comprehensive set of heterogeneous data:

- Internet Routing Registries,
- inferred from the relationships between autonomous systems,
- graphs of connections between autonomous systems,
- IP prefixes.

In this methodology, the authors did not classified autonomous systems into levels, but identified a set of classes. The main criteria for defining the set were network properties of autonomous systems.

2. DATA SOURCES AND POSSIBLE WAYS OF THEIR USE

2.1. IANA

The *Internet Assigned Numbers Authority* (IANA) organization is responsible for the global coordination of the DNS Root, IP addressing and other Internet protocol resources. IANA allocates AS numbers to *Regional Internet Registries* (RIRs). The RIRs further allocate or assign AS Numbers to network operators in line with RIR policies. The five RIRs are: ARIN (*American Registry for Internet Numbers*), RIPE NCC (*Réseaux IP Européens Network Coordination Center*), LACNIC (*Latin America and Caribbean Network Information Center*), APNIC (*Asia Pacific Network Information Center*) and AfriNIC (*African Network Information Center*). IANA publishes a range of AS numbers assigned to each RIR [5]. The report can be downloaded and saved in the database.

2.2. INTERNET ROUTING REGISTRIES

Internet Routing Registries are dispersed databases that contain records of routing policies of autonomous systems, their associated IP prefixes, contact information, etc. The natural approach to identify the type of autonomous system is an autonomous system search by the number in the IRR and analysis of the record of the organizational description. In the terminology of RPSL (Routing Policy Specification Language), this record is a “descr” (description) attribute of class “aut-num” (autonomous system number) [5]. The record of the organizational description contains the name or a brief description of the organization that owns this autonomous system. Here are sample entries attribute “descr” from the IRR:

- PENTAGONFEDERALCU – Pentagon Federal Credit Union,
- UTPA – The University of Texas – Pan American.

Attribute "descr" has no standard representation. It usually contains short descriptions, but there are cases when it is composed of abbreviations such as "MIST – MIST Inc." or "CMR1122 – CMR LLC".

It is worth mentioning that Dimitropoulos and his team took a description record of the organization as the first attribute, which can bring useful information using text analysis techniques, in order to categorize autonomous systems.

2.3. CIDR RAPORT

This service generates a daily report that contains information about the numbers of autonomous systems and their associated short descriptions of organizations that are their owners [7]. Information is obtained from the RIRs. This report provides numbers of autonomous systems and their corresponding records of organization description. Figure 1 shows ten examples of CIDR report records.

```
AS3       MIT-GATEWAYS - Massachusetts Institute of Technology
AS7       UK Defence Research Agency
AS42      PCH PCH
AS76      SDC-CAM-AS - Unisys Corporation
AS89      DNIC-AS-00089 - Navy Network Information Center (NNIC)
AS558     NET2EZ - Net2EZ
AS13946   ETECHOHIO - eTech Ohio
AS14197   HMS-ASN - Health Market Science
AS14407   XO-ASN5 - XO Communications
AS64828   Private Use AS
```

Fig. 1. Sample records from the CIDR Report

All records can be downloaded using any scripting language and saved to the database.

2.4. POTAROO

Potaroo service [8] provides information about autonomous systems in the following format: AS number, a brief description, IP prefixes and paths that describe the sequence of autonomous systems on the way to the destination IP network. Figure 2 shows an example of advertised prefixes and paths for AS12.

IP prefix	Path to the destination
128.122.0.0/16	4608 1221 4637 6453 12 12 12
192.35.210.0/24	4608 1221 4637 6453 12 12 12
192.76.177.0/24	4608 1221 4637 6453 12 12 12
192.86.139.0/24	4608 1221 4637 6453 12 12 12
216.165.0.0/17	4608 1221 4637 6453 12 12 12

Fig. 2. Advertised prefixes and paths to AS12

Information about autonomous systems IP prefixes and paths to the destination network advertised by the systems can be downloaded using any scripting language and save to the database.

3. DATA ACQUISITION

To obtain the data has been used open source software:
- Ubuntu Linux operating system,
- Apache HTTP Server,
- PHP scripting language,
- cURL library,
- MySQL DBMS.

The database schema is shown in figure 3.

Whois	
id	*int*
number	*text*
assigned	*text*
whois	*text*

As_path	
Id	*int*
as_number	*int*
as_prefix	*text*
as_path	*text*

As_info	
id	*int*
as_number	*int*
as_name	*text*
equivalent_number	*int*
total_prefixes	*int*
as_class	*text*

Fig. 3. Database schema

The service IANA taken report on AS belonging to each RIR and saved to a MySQL database. Table *Whois* (fig. 4) contains information about the numbers of autonomous systems and the regional Internet registries to which they are assigned.

The CIDR Report collected numbers of autonomous systems and records assigned to them description of the organization. This information has been downloaded using PHP and cURL library. These data were collected in text format, parsed, and then stored in *As_info* table in *as_number* and *as_name* fields (fig. 5).

id	number	assigned	whois
1	1-6	Assigned by ARIN	whois.arin.net
2	7	Assigned by RIPE NCC	whois.ripe.net
3	8-27	Assigned by ARIN	whois.arin.net
4	28	Assigned by RIPE NCC	whois.ripe.net
5	29-136	Assigned by ARIN	whois.arin.net
6	137	Assigned by RIPE NCC	whois.ripe.net
7	138-172	Assigned by ARIN	whois.arin.net
8	173	Assigned by APNIC	whois.apnic.net
9	174-223	Assigned by ARIN	whois.arin.net
10	224	Assigned by RIPE NCC	whois.ripe.net

Fig. 4. Piece of content *Whois* table

id	as_nber	as_name	eq_nber	tot_prefixes	as_class
1	0	-Reserved AS-	0	0	rpnd
2	1	LVLT-1 - Level 3 Communications, Inc.	10	3	sml
5	4	ISI-AS - University of Southern California	259	4	edu
6	5	SYMBOLICS - Symbolics, Inc.	0	0	tnu
7	6	BULL-NETWORK for further information please visit ...	106	27	sml
8	7	UK Defence Research Agency	5	2	comp
9	8	RICE-AS - Rice University	518	4	edu
10	9	CMU-ROUTER - Carnegie Mellon University	522	8	edu

Fig. 5. Piece of content *As_info* table

The Potaroo, also using the cURL library, was retrieved information broadcasted by autonomous systems IP prefixes and paths to the destination network. Data were parsed and then stored in a table *As_path* (fig. 6).

Indirectly derived attribute is the number of IP prefixes advertised by each AS. As noted by Dimitropoulos [4], small autonomous systems with a small pool of IP addresses typically broadcast several IP prefixes; however, large autonomous systems, with a large pool of IP addresses tend to broadcast a considerable number of IP prefixes with different lengths.

It is worth noting that the IP prefixes have different sizes, so the attribute can be the equivalent number of IP prefixes of length /24, covering the entire pool of advertised IP addresses. This value depends on the size of autonomous systems. If an autonomous system is great or old (appeared long time ago on the Internet; e.g. academic networks), the equivalent number of IP prefixes of length /24 will be large. In small autonomous systems, this value is usually low because these systems are broadcasting few IP addresses.

id	as_number	as_prefix	as_path
1	1	192.240.141.0/24	4777 2516 209 721 27064 5927 1
2	1	199.26.72.0/21	4777 2516 3356 25819 {1}
3	1	199.248.203.0/24	4777 2516 3356 10753 1
4	2	128.4.0.0/16	4608 1221 4637 6461 34 2
5	3	18.0.0.0/8	4608 1221 4637 174 3
6	3	117.103.68.0/23	4777 2516 7473 23947 56234 56234 56234 3
7	3	117.103.68.0/24	4777 2516 7473 23947 56234 56234 56234 3
8	3	117.103.69.0/24	4777 2516 7473 23947 56234 56234 56234 3
9	3	117.103.70.0/24	4777 2516 7473 23947 56234 56234 56234 3
10	3	128.30.0.0/15	4608 1221 4637 174 3

Fig. 6. Piece of content *As_path* table

An example of determining the equivalent number of 24-bit IP prefixes from the prefixes of different sizes is presented below. If the IP prefix is broadcast with a length /16, it contains 65536 IP addresses, while the prefix /24 contains the pool of 256 IP addresses. The equivalent number of unique 24-bit prefixes that covers the 16-bit prefix is equal to 65536/256; this means that the prefix of 16 bits covers 256 unique 24-bit prefixes.

The equivalent number of prefixes and the total number of IP prefixes announced by the various autonomous systems recorded in the *As_info* table in *equivalent_number* and *total_prefixes* fields (fig. 5). The *as_class* field was supplemented during further described autonomous systems grouping.

In summary, the procedure for data acquisition was as follows:

1. A report containing numbers, names and brief description of all active autonomous systems was made, based on CIDR Report [7].
2. Numbers and brief descriptions of autonomous systems were selected from the report and then saved to the database.
3. The number of active autonomous systems was verified using the Potaroo service [8]; missing autonomous systems were saved to the database. During the verification, we also saved those AS numbers that do not have any entries in the Internet routing registers.
4. For each autonomous system, the list of broadcast IP prefixes and paths to the destination networks were downloaded from Potaroo service [8].
5. The equivalent number of 24-bit IP prefixes was calculated from the IP prefixes of different sizes that were broadcast by each autonomous system.

4. AS GROUPING

The set of classes for the autonomous systems has been defined on the basis of their network properties. When the set of classes was defined, it was assumed that

network properties of AS, which are in one class, must differ from the network properties of autonomous systems that belong to other classes.

We proposed the following set of classes:

1. *Large autonomous systems* (called Large ASes) - very large companies with intercontinental networks that broadcast a large number of IP prefixes. Objects belonging to this class provide (with a high probability) the Internet access service.
2. *Small autonomous systems* (Small ASes) - regional companies with small or larger regional urban networks. Objects of this class also provide, with a high probability, the service of connecting to the Internet.
3. *Customers* (Customer ASes) – companies or organizations that maintain their own network, but do not provide services of the Internet connection (e.g. banks, hospitals, military networks, etc.).
4. *Universities* – educational networks. This class has been separated from the class "customers" because they have much larger networks that support thousands of end hosts.
5. *Internet traffic exchange points* (Internet eXchange Points, IXPs) – a small network, serving as points of interconnection for the members of the first two classes.
6. *Network Information Centers* (NIC) – networks that provide important network infrastructure, such as root or TLD servers.

After the preliminary analysis of the collected data, it was decided not to classify autonomous systems which are reserved and used for private purposes, or do not have any records in a Potaroo service database [8]. What is more, ASes that were not broadcasting any public IP prefixes during this research were also not classified. Generally, such systems are not used at all or are used only for transit purposes.

Subsequently, the names and a brief description of each autonomous system were checked for the presence of the following terms:

- *internet exchange*,
- *network inform*,
- *univers, college*, *educat*.

If the term *internet exchange* was found for a given AS, the "Internet traffic exchange points" class was assigned to this AS; if the term *network inform* appeared, the "network of information centers" was assigned to the AS. Finally, if at least one of the terms *univers*, *college* or *educat* was found, the "universities" class was assigned to the given AS.

Then, in other autonomous systems, the equivalent number of 24-bit prefixes, covering the entire pool of broadcasted IP addresses was calculated. If the value of this attribute was less than 8.5, then the class "customers" was assigned for such AS. If the value was higher than 8.5, but less than 1999.5, the "small autonomous systems" class

has been assigned. And finally, if the value of this attribute was above 1999.5, then the class “big autonomous systems” was assigned.

5. RESUTLS

From all evaluated autonomous systems, 37711 were found active and then classified. Unused autonomous systems of those that broadcasted no IP prefix (this means that they are used for transit) were not classified. 16159 of such systems were detected. Moreover, autonomous systems that were used for private purposes and had no description of the organization or have reserved numbers were not also classified. There were 11666 of such systems.

Table 1 shows the results of the classification of autonomous systems.

Table 1. Number of autonomous systems in each class

	Large ASes	Small ASes	Custom-ers	Universi-ties	IXP	NIC
Quantity	640	15238	19847	1714	45	227
%	1.7	40.41	52.62	4.55	0.12	0.6

Among the classified autonomous systems, 52.62% are the companies or organizations that maintain their own networks, but do not provide Internet access services. 40.41% are regional and municipal companies, which provide (with a high probability) connection to the Internet. Educational networks represent 4.55% of the total. 1.7% are huge companies with the intercontinental networks, which provide (with a high probability) the Internet connection services. Internet traffic exchange points and network information centers constitute 0.6% and 0.12% respectively.

Table 2 summarizes the quantitative and the percentage share of autonomous systems in classes “universities”, “Internet traffic exchange points” and “network information centers” calculated during both the Dimitropoulos's team [4] research and this chapter studies. Three classes from the whole set were compared, because the criterion by which they were classified autonomous systems is the same as used in the study from 2006. Other classes could not be compared due to the significant differences in the classification algorithms.

Table 2. Comparison of the number and percentages of autonomous systems in the selected classes

	Universities		IXP		NIC	
Researcher	Quantity	%	Quantity	%	Quantity	%
CAIDA	877	4.7	33	0.2	332	1.8
WUT	1714	4.55	45	0.12	227	0.6

Reading table 2 it can be seen that the increase in the number of autonomous systems in the class "universities" is quite significant. Due to the fact that the number of all active autonomous systems so significantly increased the percentage of autonomous systems class "universities" is now lower than in 2006. For Class "Internet traffic exchange points" also reported a quantitative increase, but much smaller in comparison with class "universities". Just like in the class "universities" reported the percentage decline. The exception is class "network information centers", which, compared with 2006 reported quantitative and percentage decline.

In conclusion, it is worth noting that while the study found 18.234 (48%) of active autonomous systems more than in the case of research conducted by a team of CAIDA in 2006 [4].

6. SUMMARY

This chapter collected data about the current autonomous systems: numbers, names and a brief description, sequences of autonomous systems that describe the path to the destination networks, the number of broadcasted prefixes, as well as IP and broadcasted location. Data were collected from publicly available sources, analyzed, and later provided us with attributes by which autonomous systems were divided into classes. Our study has proposed a set that consists of six classes; two of them differ from the classes that have been proposed so far. During the classification of autonomous systems we found and grouped 37771 of them; this is about 18,234 (nearly 50%) more than was classified by the authors of [4] in 2006. The study also presented a comparison of the number of autonomous systems in classes "universities", "Internet traffic exchange points" and "network information centers" in 2006 and 2012. Other classes have not been compared, since quite substantial differences in the algorithms do not allow the juxtaposition of them.

Using the classification of autonomous systems, it can be deduced that the majority of the active autonomous systems constitute companies and organizations that maintain their own network, but do not provide services to connect to the Internet. On the other hand, the least of the whole are Internet traffic exchange points.

The obtained classification helps to better understand the Internet infrastructure. It can be used to create more realistic models of Internet topology [9] and to the study of spatial forecasting of performance of web servers belonging to different autonomous systems [10].

REFERENCES

[1] GOVINDAN R., REDDY A., *An analysis of inter-domain topology and route stability*, In: Proc. IEEE INFOCOM, 1997.
[2] GE Z., FIGUEIREDO D., JAISWAL S., GAO L., *On the hierarchical structure of the logical Internet graph*, In: Proc. SPIE ITCOM, 2001.
[3] SUBRAMANIAN L., AGARWAL S., REXFORD J., KATZ R., *Characterizing the Internet hierarchy from multiple vantage points*, In: Proc. IEEE INFOCOM, 2002.
[4] DIMITROPOULOS X., KRIOUKOV D., RILEY G., CLAFFY K., *Revealing the Autonomous System Taxonomy: The Machine Learning Approach*, In: Proc. Passive and Active Network Measurement Workshop (PAM), Adelaide, Australia, 2006.
[5] Internet Assigned Numbers Authority (IANA), *Autonomous System (AS) Numbers*, [Online] http://www.iana.org/assignments/as-numbers/as-numbers.xhtml
[6] ALAETTINOGLU C., VILLAMIZAR C., GERICH E., KESSENS D., MEYER D., BATES T., KARRENBERG D., TERPSTRA M., *Routing Policy Specification Language (RPSL)*, RFC 2622 (Standards Track), Internet Engineering Task Force, June 1999.
[7] BATES T., SMITH P., HUSTON G., *The CIDR report*, [Online] http://www.cidr-report.org
[8] HUSTON G., *Geoff Huston - potaroo.net*, [Online] http://www.potaroo.net/
[9] KOSZULIŃSKI P., KOTOWSKI G., NOWAK Z., *Analiza struktury połączeń w Internecie na poziomie autonomicznych systemów*, Studia Informatica (Gliwice), vol. 34, nr 3, 2013.
[10] BORZEMSKI L., KAMIŃSKA-CHUCHMAŁA A., *The Impact of Network Characteristics on the Accuracy of Spatial Web Performance Forecasts*, Recent Advances in Knowledge-based Paradigms and Applications, Springer International Publishing Switzerland, Volume 234, 2014.

CUDA, parallel computing, GPU,GPGPU, CULA

Damian RACZYŃSKI*

MATRIX COMPUTATIONS USING GPU

The purpose of this paper is to compare computing power of GPUs (Graphics Processing Units) and CPUs (Central Processing Units) for chosen matrix operations. The architectures of both computational units have significant differences, which affect the effectiveness of matrix computations. The paper presents a comparison of the time required to perform basic matrix operations such as multiplication and the QR decomposition for square matrices of various sizes. The advantages of using GPUs are also presented on the example of the Lyapunov equation solving. This issue, used for example in theory of model order reduction to computing the controllability and observability gramians, has a high computational complexity – $O(n^3)$, which results in long execution time in case of high order models. Two popular algorithms – the Smith's and the Sign Function methods has been implemented for both GPU and CPU environments.

INTRODUCTION

The graphics processors due to the large number of cores are very efficient for large data blocks processing. The development of libraries such as CUBLAS, CULA and MAGMA causes that graphics processors are increasingly used in scientific computing, especially when applied algorithms have a high computational complexity. The paper provides a brief comparison of CPUs and GPUs in terms of architecture and computing capabilities for chosen examples.

The first chapter contains basic information about the NVIDIA FERMI architecture applied in the GPU used in the calculations. The second part contains short description of the modern x86 processors architecture, which will be used to perform similar calculations on CPU. The third chapter contains a comparison of the two architectures for basic matrix operations such as multiplication and matrix decomposition. The last part contains a comparison of GPU performance in relation to CPU, when both units are applied to solve the Lyapunov equations. Parallel computing for solving the Lyapunov

* Opole University of Technology.

equation for large scale matrices is recommended due to the computational cost $O(n^3)$. The comparison is performed for the same algorithms implemented for both GPU and CPU environment.

1. NVIDIA FERMI ARCHITECTURE

Unlike the conventional CPUs, GPUs contain a large number of cores, performing basic arithmetic operations and a small cache. This architecture, ensures high performance for processing large data blocks. Graphics processors cope well with the performance of the matrix or vector calculations with a large number of elements. However, frequent exchange of data between main memory and graphics card memory can slows computations. If the size of the processed data is not large enough, or if implemented algorithm requires frequent references to main memory, more efficient is to perform the calculation using the CPU [1].

A single unit responsible for performing calculations in the GPU is the CUDA core (CUDA - Compute Unified Device Architecture). Each CUDA core (fig. 1) is equipped with an ALU (Arithmetic and Logic Unit) for integers and Floating Point Unit (FPU). In the FERMI architecture, each floating point unit supports arithmetic compliant with IEEE 754-2008 for both single and double precision data. IEEE Standard 754-2008, compared with IEEE 754-1985, which was used by the older GPUs generations, introduces the operation of the FMA (Fused Multiply Add), which in one step executes multiplication and addition of arguments (D = A * B + C), providing higher precision than in the case of separately performing the two operations [2]. Operation on single-precision data is performed by a single CUDA core, while for double precision data by two CUDA cores. Therefore, the computational power of the GPU for operations on double-precision data is two times lower [2]. In contrast to conventional processors, CUDA cores haven't got their own registers, cache or units capable of reading or writing data from memory [3].

Group of 32 CUDA cores with additional units form the Streaming Multiprocessor (SM) [4]. Sixteen Load/Store units (L/S) allows simultaneous execution of sixteen operations referring to memory to read operands or save the results. Special Function Units (SFU) are designed for special operations, such as determining the value of the transcendental functions, or inverse of a square root (fig. 1).

The Fermi architecture supports hardware threads management. The basic unit of Streaming Multiprocessor resource allocation is a warp. The warp consists of 32 threads. Each Streaming Multiprocessor is able to manage 48 warps, which means that the maximum number of threads allocated to a single Streaming Multiprocessor is 1536 [3]. Due to the limited number of CUDA cores, at a time Streaming Multiprocessor can execute up to 32 threads. Units responsible for scheduling (two warp schedulers and two dispatch units for each Streaming Multiprocessor) opt for the 16 threads

from two warps and assign them to simultaneous run on 16 CUDA cores, 16 Load/Store units, or four Special Function Units (fig. 1) [5]. They are responsible for the optimal use of Streaming Multiprocessor. If they detect that the execution of the current operation in the warp is associated with a long time delay (for example, a reference to the DRAM), they allocate resources of Streaming Multiprocessor to instructions from another warp. Switching threads in Fermi architecture takes only one machine cycle and is called as the zero-overhead thread scheduling [3]. This approach is a key solution for high efficiency of the described architecture equipped with a small amount of cache memory.

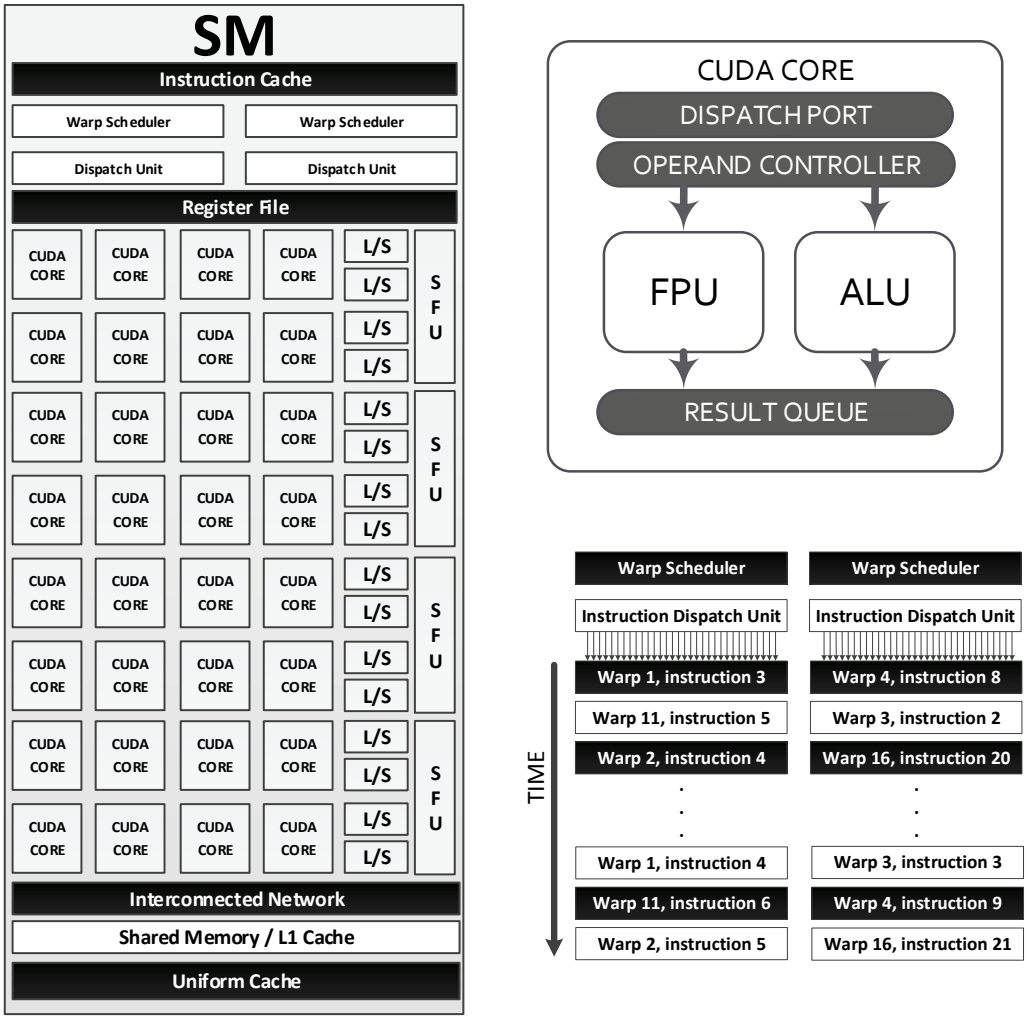

Fig. 1. Streaming Multiprocessor structure, CUDA CORE structure and dual warp scheme [5]

In the Fermi architecture, each Streaming Multiprocessor has a 64 KB of high speed memory, which is divided into two blocks corresponding to the shared memory and L1 cache. The size of both blocks can be configured in two ways into blocks of 48 KB size for shared memory and 16 KB of L1 cache or 16 KB of shared memory and 48 KB of L1 cache [2]. The specific configuration is determined by the behavior of executed threads. If threads mostly perform operations on local data, configuration with more shared memory is preferred. If is not possible to predict from which areas of the DRAM threads will load data, preferred is the configuration with a larger L1 cache [3].

A group of four Stream Multiprocessors creates Graphics Processing Cluster (GPC). The computational clusters together with L2 cache, DRAM controllers and GigaThred Engine form the GPU [6].

2. CPUs x86 ARCHITECTURE

Recent x86 processors have multi-core structure with vector units (MMX … AVX extensions) and multi-level cache hierarchy. Cache memory is designed to increase the speed of programs execution by reducing the access time to instructions and data (reducing the time delay associated with references to memory). In modern computer architectures the cache memory is divided into three levels defined as L1, L2, L3 (fig. 2). The L1 cache is the fastest type of memory, located closest to the processor core. A cache of this type is characterized by the smallest capacity. First level cache memory is divided into instruction cache and data cache. The L2 cache is slower, but much more capacious, designed for both data and instructions. The Level L3 cache is an area common to all processor's cores.

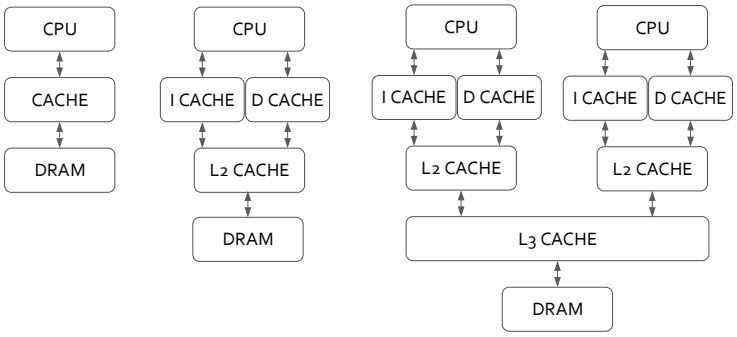

Fig. 2. The cache memory hierarchy structures

Vector processing consists in performing the same arithmetic or logic operation on multiple data simultaneously. With this approach, it is possible to reduce the time re-

quired to complete the program in case, where the operations are related to the data block. Development of vector processing technology in the x86 processors led to the introduction of AVX extension (Advanced Vector Extensions) which adds 16 new 256-bit YMM registers to processor architecture. This allows to perform operations on an eight-element vectors of single and four-element vectors of double precision data [7].

3. COMPARISON OF GPU AND CPU PERFORMANCE FOR BASIC MATRIX OPERATIONS

This chapter presents a comparison of the performance of GPUs and CPUs for basic matrix operations. Operations of multiplication of matrices are implemented using standard CUDA library. Program for matrix QR decomposition is based on the use of the CULA library. The CULA library enables the implementation of basic linear algebra operations using the GPU (the library contains selected LAPACK functions for the GPU) [8]. The CPU performance was determined by using MATLAB R2011a. The MATLAB R2011a environment uses the Intel Math Library version 10.2, which supports multi-core and data (vector extensions) parallelism [9]. The calculations were performed respectively on the CPU Core i7 3770 and the graphics card equipped with NVIDIA GTX580 GPU.

The calculations were performed on random square matrices with the dimensions given on the x-axis labels of the charts. In fig. 3-4 the charts of time required to perform the chosen matrix operation for the CPU and the GPU are presented.

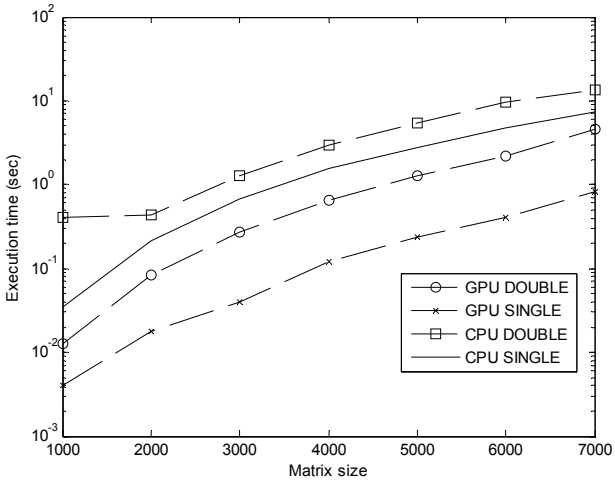

Fig. 3. The chart of time required to perform matrix multiplication

For every case, the programs executed on the GPU calculated the solution faster. The speedup coefficient determined from the relationship $\frac{CPU\ time}{GPU\ time}$, for matrices of size 7000×7000, equals respectively 2.98 for matrices multiplication and 2.85 for QR decomposition. Loading and storing matrices from/to the main memory causes the delay associated with the data transmission. A time required to perform 100 operations of matrices multiplication, for matrices of size 5000×5000, is presented on fig. 5. To minimize the impact of main memory references, the program performs only a single load and store operation.

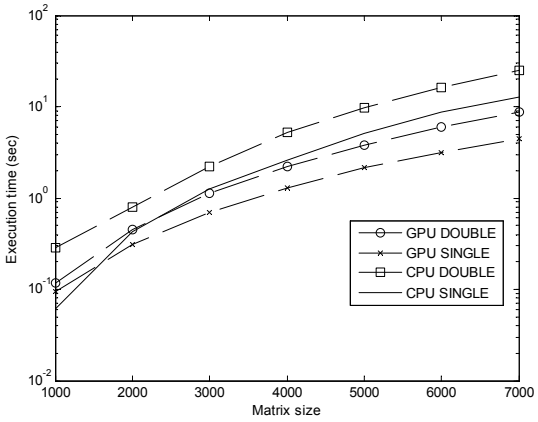

Fig. 4. The chart of time required to perform QR decomposition

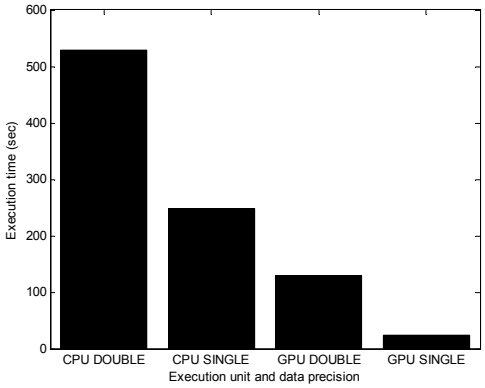

Fig. 5. The chart of the time required to perform 100 operations of matrices multiplication for the CPU and the GPU

4. COMPARISON OF GPU AND CPU PERFORMANCE FOR SOLUTION OF THE LYAPUNOV EQUATION

The Lyapunov equation expressed in the form:

$$AX + XA^T = Q \tag{1}$$

is used for example in theory of model order reduction. There are several methods for determining solution of the Lyapunov equation, for example [9]: Kronecker, Bartels-Stewart, Smith, ADI or Sign Function methods. In the chapter two selected methods – Smith and Sign Function algorithms are used to compare the performance of high complexity CPU and GPU computations in application to high-order matrices.
Smith's algorithm belongs to the group of iterative methods. The Lyapuov equation (1) is transformed to form [10]:

$$X - VXV^T = W \tag{2}$$

where [10]:

$$V = \left(qI - A^T\right)^{-1}\left(qI + A^T\right),\ W = 2q\left(qI - A^T\right)Q\left(qI - A\right)^{-1} \tag{3}$$

I – identity matrix

For negative definite matrix A, successive approximations of the solution can be determined from the following relationship [10]:

$$X_{k+1} = X_k + V^{2^k} X_k \left(V^{2^k}\right)^T \tag{4}$$

where P_0=W, q – arbitrarily assumed positive value (can accelerate the convergence of the algorithm to solution).

The Sign Function method is most widely used algorithm for solving Lyapunov equation with dense matrices of large size. The successive solution approximations are obtained from the relationships [10]:

$$A_0 = A,\ X_0 = Q,\ A_{k+1} = \frac{1}{2c_k}\left(A_k + c_k^2 A_k^{-1}\right),\ X_{k+1} = \frac{1}{2c_k}\left(X_k + c_k^2\left(A_k^T\right)^{-1} X_k A_k^{-1}\right) \tag{5}$$

where c_k – arbitrarily assumed value (can accelerate the convergence of the algorithm to solution).

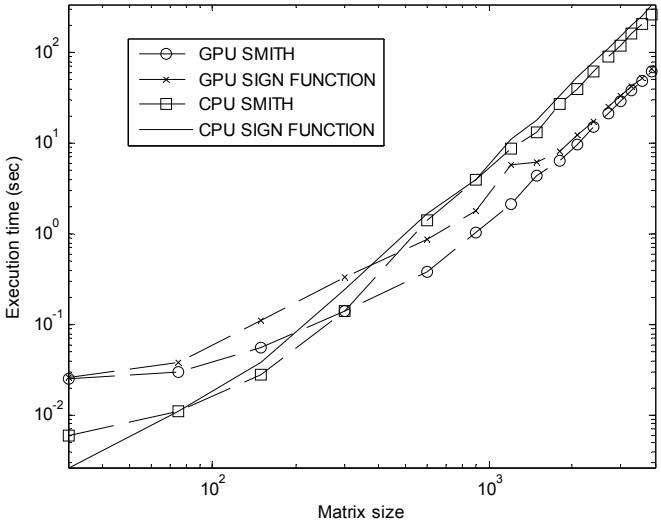

Fig. 6. The chart of time required to solve the Lyapunov equation for the CPU and the GPU

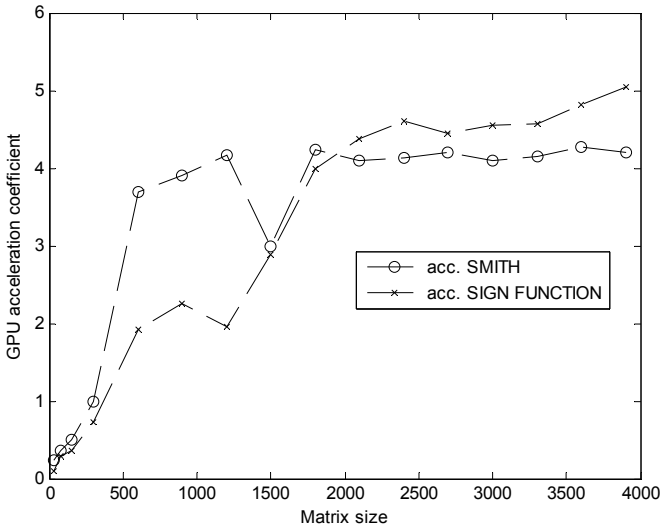

Fig. 7. The chart of the GPU speedup coefficient
for programs computing solution of the Lyapunov equation

The both implemented programs for the GPU environment use the CULA functions to perform parallel matrix computations such as matrix multiplication (GEMM function) or matrix inverse (GETRF and GETRI functions). Operations of matrix addition and subtraction are implemented using standard CUDA library, in SIMD type (many cores performs the same operation).

The chart of time required to solve the Lyapunov equation for the CPU and the GPU using the Smith and the Sign Function algorithms is presented in fig. 6.

The speedup of GPU computing is obtained only for matrices, which size is higher than 300×300 and is increasing with size of matrices (fig. 7).

CONCLUSION

The paper presents a comparison of two different computing architectures – CPUs and GPUs applied to massive matrix computations. A special attention is given to solve the Lyapunov equation, which is an important issue in the theory of model order reduction. The high computational complexity of algorithms computing the controllability and observability gramians (based on solving the Lyapunov equation) causes that the reduction of high-order models takes a long time. This is the main problem that prevents the use of methods based on gramians to reduce complex models [8].

In the literature, there are examples of using GPUs to solve the Lyapunov equation. The hybrid GPU-CPU method is proposed in [11,12]. The comparison of controllability gramians computation performance for GPU GTX465 and MATLAB R2007b is presented in [8]. This paper is an extension of research presented in [8]. The performance of both GPU and CPU is compared with use the same algorithms – in [8] the CPU performance was determined by MATLAB LYAP command.

The results show that GPU environment can be successfully used for scientific calculations with large size matrices. The programs executed on the GPU were about 4-5 times faster than the same programs executed on MATLAB R2011a .

REFERENCES

[1] KIRK D., HWU W., *Programming Massively Parallel Processors: A Hands-on Approach*, San Francisco, Morgan Kaufmann, 2010, 2–8.

[2] LEZAR E., *GPU Acceleration of Matrix-based Methods in Computational Electromagnetics*, phd. Thesis, Stellenbosch University, Department of Electrical and Electronic Engineering, 2011, 17–22.

[3] NICKOLLS J., DALLY W., *The GPU Computing Era*, IEEE Micro, Vol. 30, No. 2, 2010, 56–69.

[4] FARBER R., *CUDA Application Design and Development*, San Francisco, Morgan Kaufmann, 2011, 87–89.

[5] NVIDIA, *Whitepaper NVIDIA GF100*, 2009, 4–15.

[6] WITTENBRINK C., KILGARIFF E., Prabhu A., *Fermi GF100 GPU Architecture*, IEEE Micro, Vol. 31, No. 2, 2011, 50–59.
[7] STANISŁAWSKI W., RACZYŃSKI D., *Programowanie jednostek wektorowych procesorów rodziny x86*, Opole, Oficyna Wydawnicza Politechniki Opolskiej, 2009, 4–6.
[8] RACZYŃSKI D., STANISŁAWSKI W., *Controllability and observability gramians parallel computation using GPU*, Journal of theoretical and applied computer science, Vol. 6, No. 1, 2012, 47–66.
[9] INTEL CORPORATION, *Intel Math Kernel Library (Intel MKL) 10.2 In-Depth*, 2009, 2–3
[10] STANISŁAWSKI W., RACZYŃSKI D., *Metody wyznaczania gramianów sterowalności i obserwowalności dla ciągłych stacjonarnych układów liniowych*, In: Postępy Automatyki i Robotyki, KAiR PAN, Malinowski K., Dindorf R. (Eds.), Kielce, Wydawnictwo Politechniki Świętokrzyskiej, 2011, 571–580.
[11] BENNER P., EZZATTI P., KRESSNER D., QUINTANA-ORTÍ E. S., REMÓN A., *Accelerating model reduction of large linear systems with graphics processors,* In: Applied Parallel and Scientific Computing Vol. 7134, Kristján J. (Eds.), Berlin, Springer, 2012, 88–97.
[12] BENNER P., EZZATTI P., QUINTANA-ORTÍ E. S., REMÓN A., *Using hybrid CPU-GPU platforms to accelerate the computation of the matrix sign function,* In: Euro-Para 2009 – Parallel Processing Workshops, Lecture Notes in Computer Science Vol. 6043, Berlin, Springer, 2010, 132–139.

parallel computing, high performance computing,
GPGPU, OpenCL, heterogeneous systems

Paweł ROŚCISZEWSKI*, Jan CYCHNERSKI*, Adam BRZESKI*

A REGULAR EXPRESSION MATCHING APPLICATION WITH CONFIGURABLE DATA INTENSITY FOR TESTING HETEROGENEOUS HPC SYSTEMS

Modern High Performance Computing (HPC) systems are becoming increasingly heterogeneous in terms of utilized hardware, as well as software solutions. The problems, that we wish to efficiently solve using those systems have different complexity, not only considering magnitude, but also the type of complexity: computation, data or communication intensity. Developing new mechanisms for dealing with those complexities or choosing an existing system that suits the characteristics of our application best, requires defining and implementing adequate test applications. In this paper, we propose a regular expression matching application, which can be configured to reflect a certain computation to data intensity ratio. We support its usefulness by showing execution times of our OpenCL implementation on selected CPU and GPU devices. The results depend on input data sizes, numbers of parallel threads, but most importantly on the wildcard character properties, which allow to prepare test cases from a wide range of computation to data intensity ratios.

1. INTRODUCTION

Contemporary HPC systems allow parallelization of computations at multiple levels. In grid [1], cloud [2] and volunteer [3] computing schemes, the computations are performed by sets of geographically distributed clusters and workstations, connected by the Internet. Each cluster consists of multiple machines, usually connected by a high speed network (e.g. Ethernet, InfiniBand). Machines are equipped not only with CPU computing devices, but also, more and more often, various types of accelerators like GPU [4] or Intel Xeon Phi [5], that can be successfully used for general purpose computations. Finally, those devices consist of many cores, allowing to execute up to

* Department of Computer Architecture, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 11/12 Gabriela Narutowicza Street, 80-233 Gdańsk, Poland.

hundreds of thousands of threads in parallel. There is a trend to mix computational and network device models from various vendors and with different parameters in one system, making the systems highly heterogeneous. For example, according to the TOP500 [6] ranking, the two most powerful supercomputers at the time mix Intel Xeon CPUs with Intel Xeon Phi accelerators (Tianhe-2) or Opteron CPUs with NVIDIA K20x GPUs (Titan). This increasing heterogeneity forces software engineers to design new computing architectures, APIs and algorithms, that would allow to make the most of the available hardware. For instance, OpenCL [7] is a framework, that allows using a unified programming language to write applications, which are mapped to the specific architectures and executed in certain computing platforms. OpenCL can be then incorporated into larger systems, like KernelHive [8], which extend the parallelization to higher levels. One of the most challenging tasks of such software solutions is data management.

It should be reminded that the purpose of HPC systems is to solve some computational problems in an efficient way. The efficiency is usually related to execution time, but also increasingly often to energy consumption [9]. The problems that we might want to solve (may it be physical simulations, mathematical problems, big data analysis or any other) are usually different in terms of difficulty. Not all problems are suitable for solving in parallel. We can perceive their spectrum as proposed in Figure 1.

In general, solving a problem in parallel requires dividing the problem into subproblems, solving them independently by concurrent processes and finally merging the results. Certain problems require only partitioning the input data into chunks, which are processed independently. Problems of this type are called embarassingly parallel and are located in the lower left corner of the triangle in Figure 1. The upper corner represents problems, which require dense communication between the concurrent processes and finally, the problems in lower right corner are heavily data dependent. In extreme cases it could mean, that we only need to perform very simple computations, but on large datasets.

Obviously there are generally easier and harder problems, but the kind of their complexity is also important. Each point inside the triangle in Figure 1 represents an individual problem profile. While choosing and configuring a HPC system for our application, we should take into account its profile, because some solutions are better suited for certain types of applications. Versatile systems should be able to cope with as wide range of the problem spectrum as possible, however extreme applications cause completely different difficulties. In this paper we focus on the problem profiles varying from compute intensive to data intensive, as indicated by the red arrow in Figure 1. In case of extremely compute intensive applications, the most important goals that a HPC system should accomplish are connected with task scheduling and load balancing. In case of problems on the other side of the arrow, issues related to data storage and transmission turn to be more important.

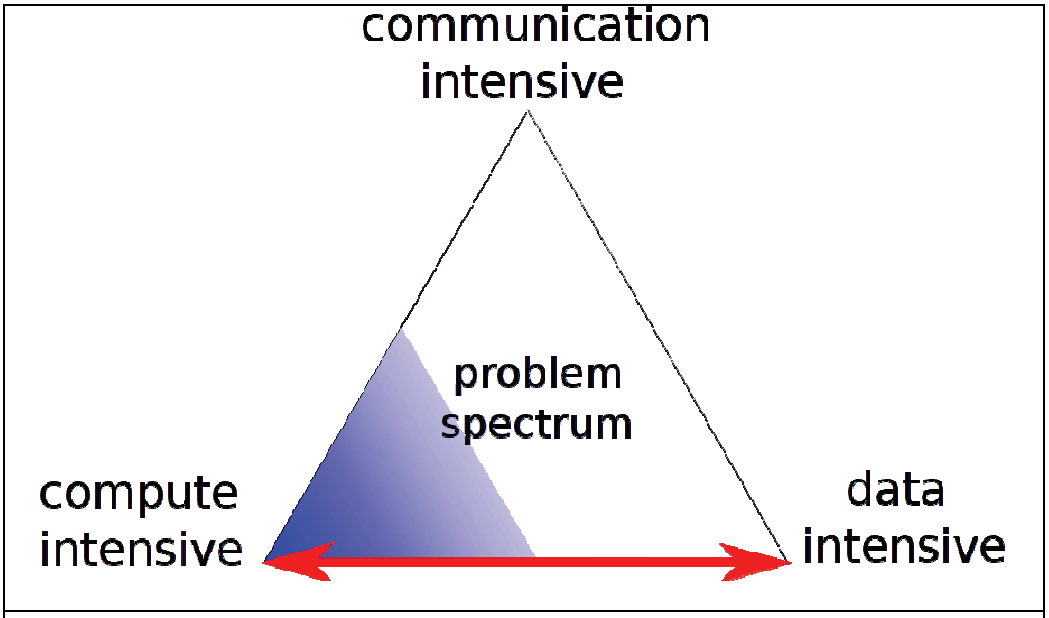

Fig. 1. Spectrum of computational problems from the parallelization viewpoint

The development process of a versatile HPC framework like KernelHive [8] or, for example choosing an appropriate solution for our purposes requires test applications, which would evaluate the fitness of the system under certain conditions. In this paper, we propose a specific regular expression matching application with configurable data intensity level, which allows to prepare test cases suitable for evaluating this fitness. We start with an overview of related work in Section 2. In Section 3, we describe the proposed application and its implementation written in OpenCL, which makes it useful for heterogeneous systems. In the experiments described in Section 4 we examine the scalability and various configurations of the application on CPU and GPU. Finally, we conclude and propose future work in Section 5.

2. RELATED WORK

During the development process, every software system should undergo software tests on various levels including unit testing, integration testing, component interface testing, system testing and acceptance testing. The application proposed in this paper focuses on the latter two, namely it should be used to verify if the completely integrated HPC system works as a whole and meets its requirements, especially concerning performance.

There is a number of benchmark applications designed for testing and comparing the performance of the hardware and software of HPC systems. For example, the LINPACK benchmark is used by the TOP500 [6] list of most powerful supercomputers.

There are also multiple applications used for testing chosen capabilities of the HPC systems. For example, the testbed of the Italian Grid Infrastructure [10] combines a set of applications in the fields of astronomy, astrophysics, oceanography, molecular dynamics, quantum chemistry, varying be real requirements from the HPC system. The application proposed in this paper is novel in the sense, that those requirements can be to some extent configured.

The proposed solution was already used as a test application in [11], where we propose a new data prefetching optimizer for the KernelHive framework.

3. PROPOSED APPLICATION

3.1. REGULAR EXPRESSION SYNTAX

The goal of the proposed application is to find all occurrences of a given regular expression and to return their absolute offsets in the input text. There are a few popular regular expression syntaxes, varying in delimiters and metacharacters used. In this work, we do not focus specifically on any of them. We use a basic proof-of-concept syntax instead. Apart from letters from the used alphabet, we allow one wildcard character, denoted by an asterisk (’*’), matching zero or more occurrences of any character.

In the existing solutions, described in Section 2.2 wildcard characters would behave either greedy or non-greedy. For example, searching for the pattern ”a*c” in the string ”abcabcabc”, a greedy regular expression would match one string: ”abcabcabc”, while a non-greedy would match three ”abc” substrings. From our application, we require to examine all possible matches, including the greedy, non-greedy and intermediate ones. In the above case, it should match the whole ”abcabcabc”, two ”abcabc” substrings and three ”abc” substrings, giving the result of six occurrences. This means, that the number of cases to investigate is exponential in the number of wildcard characters N. Parallelization of this search is possible only if we define a limit of the characters matched by the wildcard character. Otherwise, splitting the input data into chunks would not be possible. The maximum wildcard length M also influences exponentially the number of cases to be examined.

Hence, the expected execution time of the application is:

$$T(d,N,M) = O\left(d \cdot a^N \cdot b^M\right) \tag{1}$$

where d denotes the input data size and $a, b > 1$.

3.2. IMPLEMENTATION

Technically, the proposed application decomposes to multiple comparisons of characters, which is a basic functionality and can be implemented using any programming language or API. This makes our idea applicable to whichever HPC system or parallel technology. For the purposes of this paper, to be able to test the application on different programming platforms (CPU and GPU), we implemented it in OpenCL. In this Section, we describe the implementation issues.

In order to utilize concurrent capabilities of each device and search for the regular expression (called needle) in the input text (called haystack) in parallel, we divide the data into chunks called sub-haystacks. Calculation of the sub-haystack coordinates is listed in Figure 2. The memory addresses, which should be used by the current thread depend on its global id. Dividing the haystack into a set of disjunctive sub-haystacks would prevent from finding the regular expression matches that cross the sub-haystack boundaries. To avoid this, we allow each thread to match characters from the beginning of the next sub-haystack. The sub-haystack size is calculated taking into account the maximum length of a matched text (maxNeedleSize), but also with regard to the length of the whole haystack.

```
int id = get_global_id(0);

int maxNeedleSize = needleSize + N * (M - 1);

__global char *subHaystack = haystack + id * haystackSize /
    globalSize;
int subHaystackSize = haystackSize / globalSize + ((id ==
    globalSize - 1) ? haystackSize%globalSize : maxNeedleSize - 1);
int sizeLimit = haystackSize - id * (haystackSize / globalSize);
if(subHaystackSize > sizeLimit) subHaystackSize = sizeLimit;
```

Fig. 2. Code for dividing the input data into chunks

The search algorithm for an individual thread uses a stack of match cases to examine. A match case means the current position in the input data and in the regular expression. The stack is initialized by a set of match cases pointing to consecutive starting points in the data chunk and each to the beginning of the regular expression. The algorithm tries to match consecutive characters of the data chunk to the current charac-

ter of the regular expression. If the last character of the regular expression is matched, it saves the result in memory. If the current character in the regular expression is the wildcard character, it pushes M new match cases to the stack. The new match cases have consecutive starting points in the data chunk and correspond to M possible matches of the wildcard character.

The maximum size of each stack is $N \square M$ match cases. In case of GPU it may seem beneficial to keep the stacks in shared memory. However, the experiments showed, that this benefit is negligible in terms of performance, while the size limitations of the shared memory significantly limits the possible stack size. This means limiting either the data package size or the N and M parameters and excludes extreme application configurations. Therefore, we decided to use the global memory.

4. EXPERIMENTS

In order to examine the properties of the proposed application described in Section 3, we conducted a set of experiments measuring the execution times on two types of computing devices: Intel Core i5 CPU and GeForce GTX 480 GPU. Each experiment was repeated 3 times. In case of GPU, the measurements were nearly identical, but for the CPU it was important to average the results, because of different CPU load on the test machine. We implemented a dataset generator in Python, which allows to produce input packages of set size, consisting of random letters from a given alphabet. For our experiments we used an alphabet of 25 letters from ’a’ to ’z’. The searched regular expressions are constructed by interlacing required number of wildcard characters by consecutive letters from the alphabet. For example, for $N = 1$ we would use ”a*b”, for $N = 2$ we would use ”a*b*c” and so on.

The sequence of the experiments was as follows: first, we determined the optimal numbers of threads for each device, as shown in Section 4.1. Then, we observed how data size influences the execution time. The results are described in Section 4.2. Finally, we examined chosen configurations of the application by mixing different values of the wildcard properties. We show some possible application profiles in Section 4.3.

4.1. SCALABILITY

The scalability results for CPU are shown in Figure 3 and for GPU in Figure 4. We present the execution times in separate charts because the numbers of utilized threads fall in significantly different ranges. This is due to the device characteristics. In CPUs we have a few efficient physical cores (in this case 2), while on GPUs we benefit from the SIMD (Single Instruction, Multiple Data) architecture, where thousands of threads work on separate data fragments.

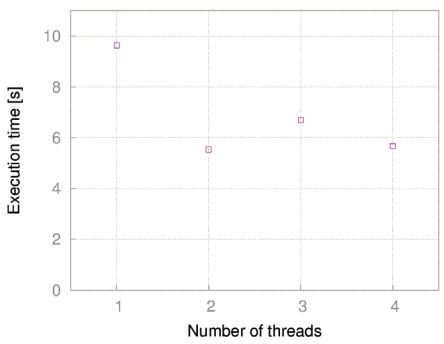

Fig. 3. Scalability on Intel Core i5 CPU (64MB, $N = 2$, $M = 100$)

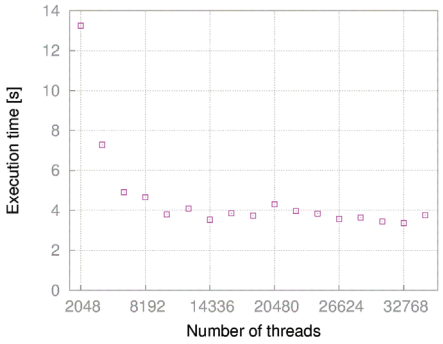

Fig. 4. Scalability on GeForce GTX 480 GPU (64MiB, $N = 2$, $M = 100$)

One of the problems in heterogeneous HPC systems is that different computing architectures, including the two mentioned above are suitable for different types of computations. There are algorithms, which execution times on one architecture can be hundreds of times faster than on the other. Because of that, in heterogeneous HPC systems it is often worth to distinguish multiple types of computations and prepare scheduling mechanisms aware of those types and computing device capabilities. The application proposed in this paper is designed for testing HPC systems with regard to data intensity. To avoid the issues with efficiency inequality, it is desirable that the execution times on different architectures should be similar. The execution times for heterogeneous architectures presented in Figures 3 and 4 are noticeably smaller on the GPU, but of the same order of magnitude as on CPU. This appears to be an advantage of using the application as a test case for heterogeneous systems.

What is more, the scalability tests show that the application utilizes the capabilities of both devices well. The optimal number of threads on the CPU is 2, because it is equipped with two physical cores. Unfortunately, the HT (Hyper-Threading) mechanism does not support this type of computations well enough to efficiently utilize three or four virtual cores. In case of the GPU, the application scales well up to ten thousand of threads. Then for subsequent numbers of threads, due to the parameters of the GPU model, the execution times oscillate around a fixed value, giving the optimal number of 32768 threads. In the following experiments, we use the optimal thread numbers.

4.2. DATA SIZES

As we expected based on the Equation 1, the results presented in Figure 5 confirm that the execution time of the application is linear in the input data size, with minor fluctuations. In the process of testing HPC systems we might need to use various data package sizes, depending on the system architecture and which mechanisms we test. Steady growth rate provides a good base for preparing various application profiles for given data package sizes by configuring the N and M parameters as described in Section 4.3.

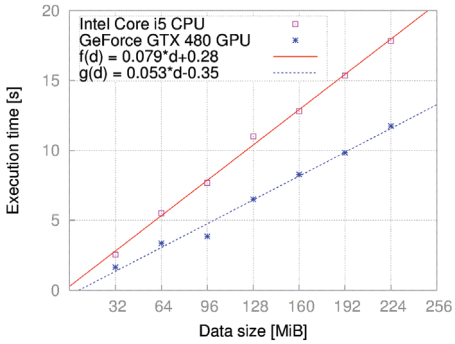

Fig. 5. Execution times depending on the data size d ($N = 2$, $M = 100$)

4.3. APPLICATION CONFIGURATIONS

Figures 6 and 7 show, that the parameters N and M indeed influence the execution time exponentially. In case of N, the base of the exponent is high, which means that small changes in N cause big changes in the execution time. On the contrary, in case of M the function is nearly linear.

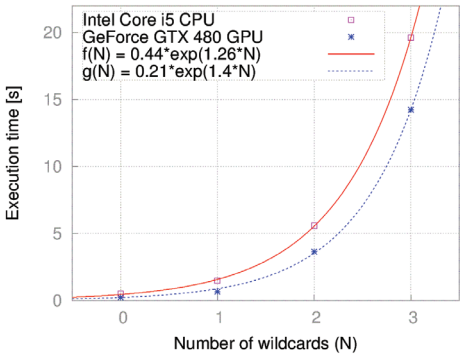

Fig. 6. Execution times depending on the number of wildcards N ($M = 100$)

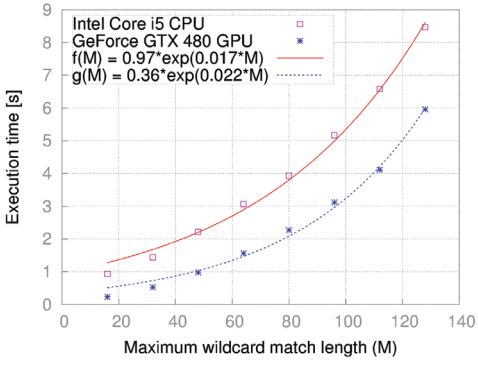

Fig. 7. Execution times depending on the maximum wildcard length M ($N = 2$)

The above dependencies allow to tune a problem instance to a precise computation to data intensity ratio by performing the following steps:

- choose the desired data package size;
- tune the data intensity using the *N* parameter
- adjust the data intensity precisely using the *M* parameter.

The proposed technique can be used to prepare test applications from a range of profiles depicted by the red arrow in Figure 1. The sample configurations of extremely computation intensive and extremely data intensive cases along with their execution times on CPU are listed in Table I.

Table 1. Exemplary application profiles

Description	Data size	*N*	*M*	CPU execution time
Compute intensive	1MiB	5	256	431.563s
Data intensive	512MiB	0	0	3.207s

5. SUMMARY

In the paper we proposed a regular expression matching application that can be configured to resemble applications from a wide range of data intensity. We described our OpenCL implementation and showed execution times on CPU and GPU for various data sizes and wildcard character parameters. We showed how to prepare required test cases by modifying those parameters. We determined the configurations of exemplary extreme configurations, where the execution time on CPU can be over 7 minutes for 1 MiB of data, but also about 3 seconds for 512 MiB of data.

We presented the results for environments with single CPU and GPU devices. Additionally, in [11] the application has been proven useful for developing and testing a dynamic data management scheme in the KernelHive heterogeneous framework. Tests of the application in environments with different characteristics indicate that it is general enough to be utilized by a wide range of parallel environments.

ACKNOWLEDGEMENTS

The work was performed partially within grant “Modeling efficiency, reliability and power consumption of multilevel parallel HPC systems using CPUs and GPUs” sponsored by and covered by funds from the National Science Center in Poland based on decision no DEC-2012/07/B/ST6/01516.

REFERENCES

[1] GRANDINETTI L., Ed., *Grid computing*. Amsterdam [u.a.]: Elsevier, 2005.

[2] PANDEY S., KARUNAMOORTHY D., and BUYYA R., *Cloud Computing: Principles and Paradigms*, Wiley Press, New York, USA, February 2011, ch. Workflow Engine for Clouds, pp. 321–344, ISBN-13: 978-0470887998.

[3] ANDERSON D.P., *Volunteer computing: the ultimate cloud*, ACM Crossroads, Vol. 16, No. 3, pp. 7–10, 2010.

[4] THOMPSON C.J., HAHN S., and OSKIN M., *Using modern graphics architectures for general-purpose computing: a framework and analysis*, in MICRO. ACM/IEEE, 2002, pp. 306–317.

[5] Intel Corp., Intel Xeon Phi Coprocessor Instruction Set Architecture Reference Manual, 7 September. [Online]. Available: https://software.intel.com/sites/default/files/forum/278102/327364001en.pdf

[6] DONGARRA J. and LUSZCZEK P., "Top500." in Encyclopedia of Parallel Computing, D. A. Padua, Ed. Springer, 2011, pp. 2055–2057.

[7] Khronos OpenCL Working Group, The OpenCL Specification, version 1.0.29, 8 December 2008. [Online]. Available: http://khronos.org/registry/cl/specs/opencl-1.0.29.pdf

[8] CZARNUL P., LEWANDOWSKI R., ROŚCISZEWSKI P., and Schally-Kacprzak M., *Multi-level parallelization of computations using clusters with gpus*,, Poster presented at the GPU Technology Conference, San Jose, USA, 2013.

[9] CZARNUL P. and ROŚCISZEWSKI P., *Optimization of execution time under power consumption constraints in a heterogeneous parallel system with gpus and cpus*, in ICDCN, ser. Lecture Notes in Computer Science, M. Chatterjee, J.-N. Cao, K. Kothapalli, and S. Rajsbaum, Eds., Vol. 8314. Springer, 2014, pp. 66–80.

[10] ALFIERI R., AREZZINI S., BARONE G.B., BECCIANI U., BENCIVENNI M., BOCCIA V., BOTTALICO D., CARRACCIUOLO L., CESINI D., CIAMPA A., COSTANTINI A., COZZINI S., PIETRI R.D., DRUDI M., GHISELLI A., MAZZONI E., OTTANI S., VENTURINI A., and VERONESI P., *The hpc testbed of the italian grid infrastructure*, 2014, 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Vol. 0, pp. 241–248, 2013.

[11] ROŚCISZEWSKI P., *Dynamic Data Management among Multiple Databases for Optimization of Parallel Computations in Heterogeneous HPC Systems*, The Fourth International Conference on Computer Science, Engineering and Applications, pp. 10–19, 2014.

virtual machine monitoring,
self-healing, KVM, VirtIO-Serial, virtualization

Kamil SZCZYGIEŁ*
Krzysztof BIELAWSKI**

MONITORING AND SELF-HEALING OF KVM VIRTUAL MACHINES USING VIRTIO

Virtual environments consisting of large amounts of virtual machines require administrative monitoring on daily basis. It is usually performed through centralized management software such as VMware vCenter or Microsoft System Center Virtualization Manager. Mentioned tools provide overall virtual machine information, resource usage statistics, ability to trigger alarms etc. With introduction of platforms to execute analytics of virtual environments such as VMware vCenter Operations Manager monitoring is more advanced than ever. However, in open source world there is no such developed software leveraging KVM hypervisor to provide detailed control over environment and application health. Solution proposed in this chapter tries to address this issue by providing networkless way to monitor virtual machines operating system with agent running inside. By using VirtIO-Serial transport mechanism in this solution it is possible to track health of services, applications, system events and perform basic diagnosis and self-healing when fault has been detected.

1. INTRODUCTION

Leading technology in modern data centers is virtualization. It consists of large amounts of hardware (servers, storage arrays, network hardware) and virtualized components (virtual machines, virtual switches, storage area networks). All of these require administrative monitoring on daily basis. Administrators perform tasks such as:

- checking virtual machines operating system (are there fatal errors, are all services running properly, check if it was not compromised),
- monitoring bandwidth usage of switches (checking if device is not overloaded, not compromised, checking for network problems),

* Intratel Sp. z o.o.

** Bialystok University of Technology.

- monitoring storage arrays (checking space usage, I/O efficiency, hardware components health such as disks or controllers).

All of these tasks require a lot of time dedicated only to monitoring datacenter infrastructure. To address this issue virtualization vendors created tools such as VMware vCenter Operations Manager [1] or Microsoft System Center Operations Manager [2] to control virtual and physical environment more efficiently from one, centralized component. Included analytics allows administrator to acknowledge problems very fast and perform required steps to fix them before they will influence users. However, in open source world there is no so well developed software leveraging KVM [3] hypervisor to provide detailed control over virtual environment health. There are many solutions exploiting Xen [4] hypervisor such as Snooze [5] – open-source, scalable, autonomic and energy-efficient management framework. It allows to control life-cycle of virtual machines and implements basic self-healing mechanisms. Another software based on Xen is CloudDVMM [6] – distributed virtual machine monitor for cloud computing. Very similar solution to CloudDVMM is DMMVM [7] – dynamic management and monitoring of virtual machines. It enables user to participate in monitoring by deploying security policies according to the workload. It is easy to notice that KVM is in minority since all of these solutions are based on Xen. This inspired us to create a monitoring and self-healing solution for KVM virtual machines harnessing VirtIO [8] on both – Linux and Windows platforms providing following features:

- Monitoring guest operating system log files, event logs,
- Performing basic diagnosis and self-healing when known fault has occurred,
- Monitoring running services and restarting them when they are not running,
- Monitoring running applications and restarting them when they are not running,
- Network-less communication between guest operating system and virtual machine host,
- Email notifications of performed actions, detected problems,
- Centralized platform to manage and monitor virtual infrastructure health.

These features would allow administrators running KVM based datacenters to have more detailed and efficient way to control managed environment. It is also aimed to be secure and flexible platform for future growth and modifications with few requirements highlighted in this paper [9] such as small performance impact or ability to monitor any data on target OS.

2. VIRTUAL MACHINE HEALTH

2.1. WHAT IS VIRTUAL MACHINE HEALTH

Virtual machine health represents overall state of the guest operating system and underlying virtual hardware. When there is no problems (bottlenecks, compromises, application or service failures) virtual machine is healthy. However, when a problem occurs it will impact other components of operating system leading to data corruption or big performance issues. Detecting such failures early allows administrators to react quickly to minimize losses in virtual environment. Available mechanisms require administrative effort while data center self-healing mechanisms minimize it. Implementing self-learning analytics provide a way to learn how infrastructure changes, which component might become corrupted in near future, which performed actions cause loss in performance. Having these information in central place allow administrators to minimize amount of work and time dedicated for environment monitoring and better planning for emerging growth.

2.2. COMMON METRICS

As it was mentioned health is very general term. It consists of many metrics which computed together determine overall virtual machine state and performance. In modern monitoring software common metrics consist of:

- CPU usage,
- Memory usage,
- Storage usage,
- Disk I/O operations (IOPS),
- Network I/O operations.

However, there are many more elements essential for the operating system monitoring such as running application and services. It is crucial to keep track of process resource usage for efficiency and occurring system events for detailed information about operating system health. Proposed solution include these extended metrics.

2.3. HEALTH MONITORING COMPARISION

Currently most popular software to monitor virtual infrastructure is VMware vCenter Operations Manager [1] and Microsoft System Center Operations Manager [2]. Although they have similar names, they are operating slightly different. Tool from VMware provides more overall information about infrastructure health including risk calculation (if resources available will be enough for future growth) and efficiency (if virtual machines are configured in efficient manner). It also include analytics mecha-

nism that provide information such as time remaining for CPU, memory, disk space, disk I/O, network I/O based on environment life until recent time. Using this information administrator is able to evaluate if current hardware is enough for future infrastructure growth at current rate. Product from Microsoft focuses on different metrics. It evaluates occurring system events in virtual or physical operating systems and provides detailed information about warnings and errors. It also contains rich knowledge base about possible causes and resolutions for problematic events existing in infrastructure. Used approach allows for very detailed operating system and application monitoring.

2. PRINCIPLE OF OPERATION

Proposed solution leverages agent architecture presented previously in paper [10]. It consists of three components:

- Central controller – gathers all health information from virtual machines and presents them to administrator through web portal. This allows for analysis of overall infrastructure health and virtual machine health, brings out problems immediately to analyzer attention,
- Proxy controller – resides in host operating system and functions as relaying proxy for information from central controller to virtual machines and vice versa,
- Guest agent – resides in guest operating system as a system service. Monitors health of particular components, immediately reports detected problems and performs self-healing if occurred failure identifier is found in knowledge base represented as XML file which include resolutions to known events.

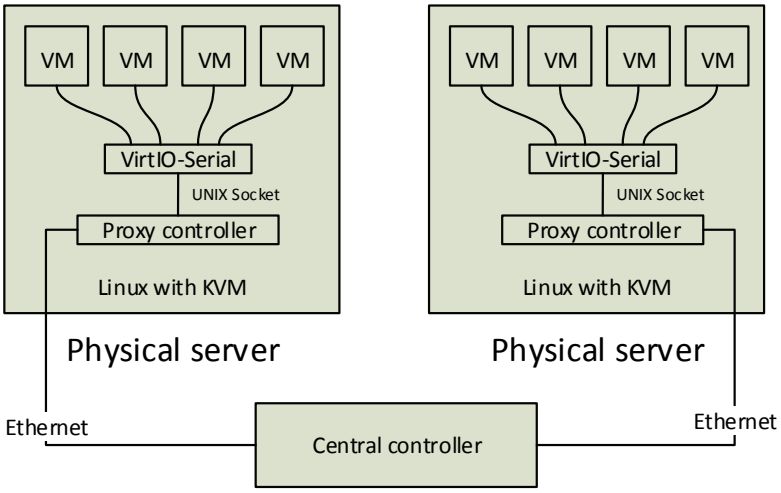

Fig. 1. Architecture diagram of proposed solution

All communication is based on exchanging messages in JavaScript Object Notation (JSON) [11] format. Central controller periodically poll health status from all virtual machines by sending a request. When proxy controller receives communicate it is relayed to the virtual machine guest agent using VirtIO-Serial transport channel. VirtIO-Serial is transparent, hardware communication layer between virtual machine and physical server, where this particular virtual machine resides. It can be presented in host operating system as many devices such as TCP servers/clients, text files, named pipe, UNIX socket and more. In guest operating system it is presented as PCI device. In proposed solution VirtIO-Serial resides as UNIX socket. Received request is relayed through this socket to the guest agent in guest operating system. When agent receives communicate it performs whole system check on demand and sends gathered results back to the central controller.

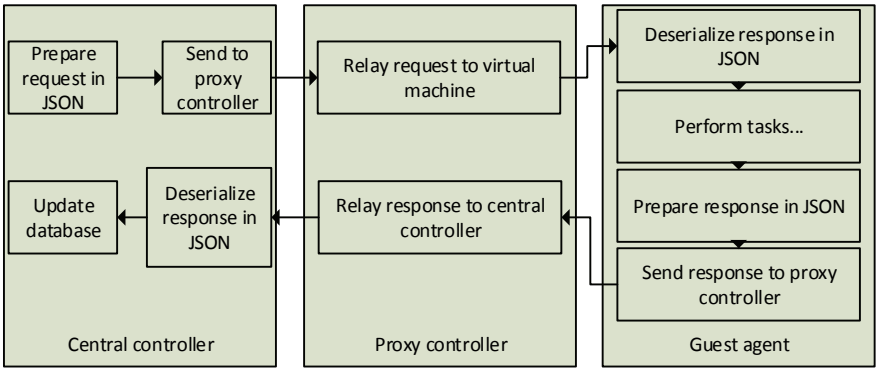

Fig. 2. Communication diagram of proposed solution

Additionally guest agent is monitoring resources such as:

- System logs, events for warning and errors by analysing files and event logs,
- System services and application health by checking process states,
- Network connectivity by performing ping tests,
- Resource usage by monitoring resource usage per process.

When abnormal situation is detected agent send a notification to the central controller and administrator is notified through email about anomaly. Events such as lost network connection, very high resource usage by process, warnings, errors, hanged services or applications are considered as abnormal system operation. Guest agent also consists of basic knowledge base about anomalies and what actions it should perform. In case of detection, after notifying administrator service will check knowledge base for possible resolution. When resolution is found agent will perform tasks included in it to recover from failure. For example administrator would want to know when virtual machine lose network connectivity. Resolution to that situation is depicted in Figure 3.

At beginning agent check if there is a network connectivity to Google DNS server (8.8.8.8) by using ping. If not, network card status is determined. In case of failure agent will force graceful reboot of the virtual machine, in other case it will continue. Next test is to check if network card has any configuration. If not, agent will ask central controller for last known network configuration.

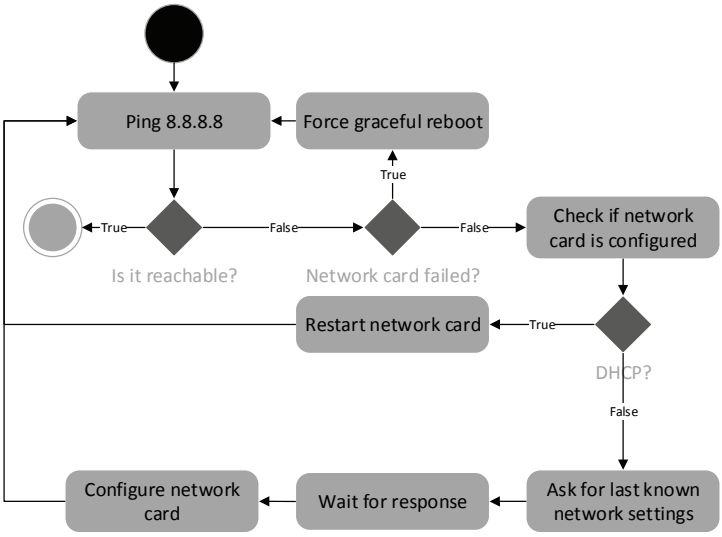

Fig. 3. Activity diagram of network diagnosis

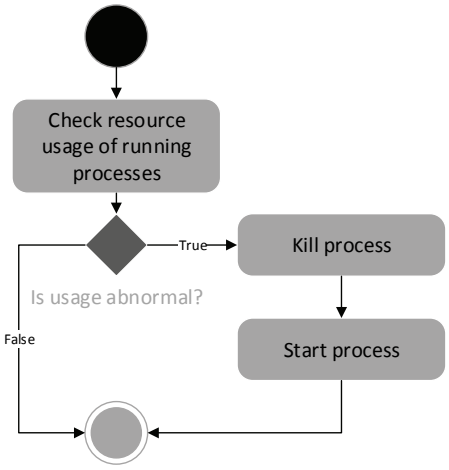

Fig. 4. Activity diagram of process resource usage diagnosis

Guest agent is also capable of monitoring running processes inside guest operating system. It is possible to detect if application is responding and if not restart it. Agent is able to monitor if process is running (when it should) and if it is not running (when it should not). When such situation is detected either process is killed or started. It is possible to monitor resource usage per process. If abnormal usage is detected specified action should be performed. Example resolution is shown at Figure 4. Agent will check resource usage of all running processes and when very high usage is detected it will restart problematic process.

Guest agent is also capable of monitoring system events and log files every specified interval. It will check if any new events are present and when they are found action is performed depending on event type:

- Information – no action is performed,
- Warning – email notification is sent to the administrator,
- Error – agent checks if resolution to this event is available in knowledge base. If yes, it will perform self-healing action, if not no action is performed. In both situations email notification is sent to the administrator.

3. EXAMPLES OF USAGE

Proposed solution allows for network-less monitoring of KVM virtual machine guest operating system, basic self-diagnosis and self-healing. It is possible to:

- Monitor system events and log files to inform administrator about issues inside guest operating system,
- Monitor applications and services state and notify when they are not running with performing restart when required,
- Monitor resource usage and react in case of very high usage by one process,
- Perform self-diagnosis on regular interval like checking network connectivity,
- Perform basic self-healing when event with resolution available in knowledge base occurs.

Specifying which services should be monitored is done through XML file. It contains service name and parameter specifying if service should be monitored. An example structure of that file is shown below:

```
<?xml version="1.0" encoding="utf-8" ?>
<Services>
  <Service Name="DHCP" Monitored="true"/>
  <Service Name="EventLog" Monitored="true"/>
</Services>
```

Knowledge base for system events is also presented as XML file. It contains two sections:

- Actions – represents possible actions available in guest agent,
- Events – represents events with their source. It contains description and action which agent should perform in case when error occurs.

An example structure of that file is shown below:

```
<?xml version="1.0" encoding="utf-8" ?>
<KnowledgeBase>
  <Actions>
    <Action Id="1" Name="Ignore"/>
    <Action Id="2" Name="Resync time"/>
  </Actions>
  <Events>
    <Event Id="5719" Source="NETLOGON">
      <Action Id="1"/>
      <Description>Netlogon service may start before the network is ready, the computer may be unable to locate the logon domain controller.</Description>
    </Event>
    <Event Id="129" Source="Microsoft-Windows-Time-Service">
      <Action Id="2"/>
      <Description>The Windows Time service cannot discover the configured time source peer.</Description>
    </Event>
  </Events>
</KnowledgeBase>
```

Guest monitor is also able to monitor for application specific events. It checks Application system log and checks if resolution to the found application errors is available in knowledge base. If resolution is found, it will perform self-healing action. If not no action is performed. In both cases administrator is notified about error and about tasks that were performed. Example of software that can be monitored is Microsoft SQL Server. Resolution example is depicted in Figure 5. Guest agent monitors Application event log for SQL Server errors. If an error is found monitor notifies administrator and checks event identifier (in this case it is 107) and search for resolution to this event in knowledge base. Next test is to check if SQL Server Database Engine is started. If not, it will restart SQL Server. Otherwise agent will perform next action which is restarting SQL Server Reporting Services service.

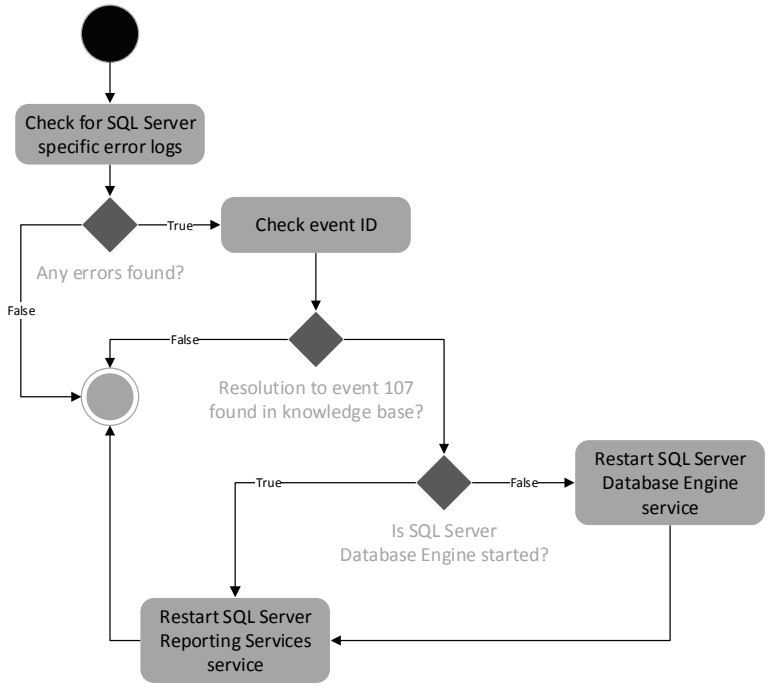

Fig. 5. Activity diagram of SQL Server event 107 diagnosis and resolution

4. CONCLUSIONS

The proposed solution is facing the challenge of KVM virtual machine guest operating system monitoring. Compared to the competitors it is rather limited, but architecture allows for flexible and easy development towards integrating more advanced solutions. As the demand for virtualization and centralized infrastructure monitoring is increasing, introducing solutions as described in this paper is going to be necessary in every virtual environment. By harnessing VirtIO-Serial requirement for network connection in virtual machines is removed. Because of that one of points of failure is eliminated and managed infrastructure can be more reliable and secure. Proposed self-diagnosis and self-healing mechanisms provide basic solutions to possibly occurring errors or failures with non-invasive, simple resolutions. Proposed monitor is highly expandable and flexible. Adding new applications or services that should be monitored boils down to modification of a XML file. Updating knowledge base for occurring events is also done through XML file. Because of that it is simple yet powerful agent solution.

ACKNOWLEDGEMENTS

This work was supported under the grant UDA-POIG.01.04.00-20-002/11 by the Polish Agency for Enterprise Development and Intratel Sp. z o.o.

REFERENCES

[1] VMWARE, *VMware vCenter Operations Manager,* http://www.vmware.com/pl/products/vcenter-operations-management, 04.05.2014.
[2] MICROSOFT, *Unified management for Cloud OS with System Center 2012 R2,* https://www.microsoft.com/pl-pl/server-cloud/products/system-center-2012-r2/default.aspx, 04.05.2014.
[3] KVM, *Kernel Based Virtual Machine,* http://www.linux-kvm.org, 04.05.2014.
[4] XEN, *The Xen Project*, http://www.xenproject.org, 12.05.2014.
[5] FELLER E., RILLING L., MORIN C., *Snooze: A scalable and Autonomic Virtual Machine Management Framework for Private Clouds*, Cluster, Cloud and Grid Computing (CCGrid), 2012, 482–489.
[6] LI RUAN, JINBIN PENG, LIMIN XIAO, XIANG WANG, *CloudDVMM: Distributed Virtual Machine Monitor for Cloud Computing*, Green Computing and Communications (GreenCom), 2013, 1853–1858.
[7] WEI CHEN, QIAOYAN WEN, *An architecture for dynamic management and monitoring of virtual machines*, Cloud Computing and Intelligent Systems (CCIS), 2012, 444–448.
[8] VirtIO, *Paravirtualized drivers for KVM,* http://www.linux-kvm.org/page/Virtio, 04.05.2014.
[9] PAYNE B.D., DE CARBONE M.D.P., WENKE LEE, *Secure and Flexible Monitoring of Virtual Machines*, Computer Security Applications Conference, Twenty-Third Annual, 2007.
[10] SZCZYGIEŁ K., BIELAWSKI K., *Controlling KVM virtual machine guest with VirtIO*, In: Design, development and implementation of real-time systems, Warsaw, 2013, 169–179.
[11] JSON, *JavaScript Object Notation*, http://www.json.org, 14.05.2014.

embodied agent, agent software architecture
robotic middleware, autonomous mobile robot,

Henryk DOBROWOLSKI*

MIDDLEWARE IN SOFTWARE ARCHITECTURE FOR AN AUTONOMOUS MOBILE ROBOT — EMBODIED AGENT

This paper presents an approach to robotic middleware as a tool to facilitate the programming and testing of complex tasks for autonomous mobile robot with a very limited hardware resources. The motivation of a project was to enable experiments with a low cost multi-robot system. This intermediate layer, middleware, allows to execute selected tasks in the environment of an external, stationary computer instead of robot on-board system. It allows testing these tasks in a relatively comfortable workstation environment and facilitates conducting experiments. It can also be used when the computational complexity of a task prevents its execution on the robot's on-board computer because of its limited resources. Middleware is bound with a system architecture but can be placed at different levels of this architecture through the exchange of interfaces of its application layer protocol. It can be easily adapted to different types of communication and its protocol stack taking into account a possible unreliability of a physical channel.

1. INTRODUCTION

Design and programming system for an autonomous mobile robot is a complex objective, requiring different competencies - including embedded real-time systems programming, image processing, digital filtering, movement control etc. Such a robot, capable to act independently in the environment and seeking to achieve given goal can be called an agent [1,2] – and to differentiate it from software agent we can talk about *embodied* agent. If we are dealing with the autonomous mobile robots performing jointly submitted job, we say that they form embodied multi-agent system (EMAS). You can find a variety of practical applications for such systems, e.g. relating to situa-

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

tions risky to human activity, like in search of survivors after the earthquake or looking for the source of leaks of toxic or radioactive substances.

The aim of our experiments with mobile robots in essence is not robotics for itself, but rather issues related to the design of the complex, "intelligent" embedded systems with limited hardware resources, and in which the tasks are belonging to different branches of computer science, that is, all of these problems that may exist in the embodied multi-agent systems.

2. MIDDLEWARE AND ROBOT SYSTEM ARCHITECTURE

Experiments with autonomous mobile robots require software development consisting of cooperating tasks for different purposes – control, building map (model) of environment and localizing robot position, action planning, cooperation with other robotic agents, etc. The whole thing is a real-time system. To make this possible we need first to specify the system architecture, including dependencies and interfaces between tasks. Development of software for each task requires an exhausting testing and “tuning” and involves making changes in the code, and often experiments with diverse versions of algorithms. This, in turn, poses a problem when the robot onboard computer system provides neither tools to support testing nor programmer’s interface. Similarly, if the tasks are to be designed and tested independently of the hardware platform specific mobile robot, and then used in the control of different, mutually incompatible, mobile robots. In both these cases, you can design an abstract architecture of a robot and accomplish the robot control through the middleware. In the simplest case, its role is to provide sensor and actuator online interfaces to software running on a desktop computer (workstation).

2.1. ROBOTIC MIDDLEWARE

During the last several years there was a series of projects including middleware, usually associated with a particular system architecture and integrated with the development framework. Often these environments can also simulate the operation of mobile robot. Most of them originated as a part of university-based research programs (e.g. Player [3, 4], MIRO [5], MARIE [6] or RSCA [7]),but there are also those that are being developed within the framework of the projects carried out by public authorities (like CLARAty [8, 9, 10] developed by NASA JPL up to 2006 in connection with the unmanned Mars missions), or even been designed for commercial purposes associated with the production of robots, including military applications (e.g. iROBOT Aware). We must also mention the freely available MRDS (*Microsotf Robotic Developer Studio*) environment [11] for .NET-based programming to build robotics applica-

tions and remote control of the robot. A comprehensive review of major robotic middleware can be found in the [12].

If all the software of the robot (apart from services reading sensors and controlling effectors) is executed in a workstation and not in robot's on-board microcomputer, than such a mobile robot can be called only conceptually autonomous, because any loss of communication between the robot and the external real-time control system means the end of robot operation. For testing purposes and at a small distance we can assume the reliability of physical channel (it is commonly Wi-Fi), however, it is difficult to accept if the robot operates in inaccessible environment and a loss of the connection is caused by its accessing in an area with a strong suppression of the propagation of a radio signal. Actually just CLARAty assumes that the system architecture is composed of two compound layers, one of which, functional layer, is implemented in the robot onboard system and the second, decision layer, is working in a remote system.

2.2. AGENT ARCHITECTURE

Agent system architecture was designed after an analysis of previously developed concepts, including:

- subsumption architecture proposed by R.A. Brooks [13], which represents a hierarchy of possible behaviors competing to exercise control over the agent, and higher levels are able to subsume lower levels in order to create viable behavior;
- TOURINGMACHINES, a hybrid architecture developed by Ferguson [14], consisting of perception and action subsystems and three control layers: modelling, planning and reactive, embedded in a control framework, which mediates between the layers;
- INTERRAP – hierarchical layered architecture, according to the authors "pragmatic approach to designing complex dynamic agent societies (..) and cooperative scheduling" [15].

The last two above mentioned architectures were verified only by simulation experiments .

The proposed architecture is an attempt to integrate ideas of the aforementioned approaches, and has been divided into the following layers (starting with lowest):

(1) performing simple orders associated with the movement, including reflexive reactions (e.g. not colliding with an obstacle)
(2) performing compound, conditional sequences of movements (behaviors) described by state machines (not necessarily deterministic, i.e. some transitions can be probabilistic), e.g. follow along the wall on your left/right side, or random "sightseeing" to provide data for mapping; actions within a state machine are orders for the layer 1

(3) making choice of behavior from the library with the ability to modify the probability of chosen transitions (e.g. to try avoiding an obstacle to the left or right side), or generating a new sequence of actions mandated by the higher level (4), to be executed by the layer 2
(4) route design, based on the model of the environment and the current goal – the direction of exploration or, if interesting objects in the field of view – the objective can be an object indicated by the layer 5; second part of this layer is an incremental environment modeling i.e. creation of robot's "world" map and specifying its own position at this map (SLAM – *Simultaneous Localization and Mapping*)
(5) specifying the current goal on the basis of negotiations (layer 6) and the current state of the world model – this layer can be seen as a very simplified BDI (*Belief-Desire-Intention*) agent architecture [1, 2]
(6) social behavior layer involving negotiations with other robots forming the multi-agent system to determine the current goal of the activity (task allocation), but also, for example, priority negotiation if the motion paths of two robots are colliding (as run through a narrow gate in opposite directions).

Layers 0 to 2 have a reactive functionality. It means that at most time robot can act without any intervention from higher levels. We can say that these layers are by 'body' side of an agent. Layers 3 to 5 are responsible for an 'intelligent' behavior and by analogy to 'body' can be called a 'mind' of a robotic agent.

The individual layers have different complexity and work concurrently, but the decisions of the higher layer in the hierarchy have binding nature for the layer located below (e.g. reflex action is to avoid an obstacle, but this reflex can be prohibited by a higher layer by a request to try to push through this obstacle). In the above description a number of properties are not shown for the brevity.

2.3. MIDDLEWARE FOUNDATIONS

The main objectives of the middleware project were:

a. hiding the lower layers of communication protocol stack (according to OSI model – transport layer and below) to allow the use of different physical channels depending on the needs and capabilities of the specific project,
b. taking into account the physical channel failure by incorporating a mechanism of notification of loss of connection and its automatic recovery – the reaction for connection loss can be, for example, executing the behavior to return to the last place at which the connection was active or/and continuation of the autonomous action according to the last strategy until its exhaustion or connection recovery,

c. providing real-time communication using prioritization of messages depending of their nature,
d. such a specification and design of application layer within the communication protocol stack, that its interfaces on both sides could be easy changed allowing to place the middleware between different layers of the system architecture for a given project.

Fig. 1 shows possible variants of middleware location within proposed architecture, in a case of relatively small resources of robot's on-board controller.

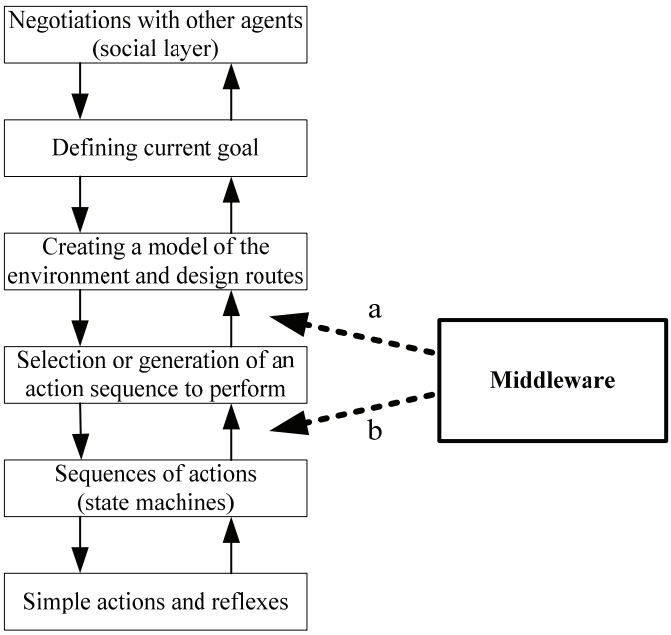

Fig. 1. Location of middleware within agent architecture

3. TEST IMPLEMENTATION OF THE MIDDLEWARE

The initial idea for the middleware project was not binding to the specific type of physical channel and corresponding protocols. Experimental middleware has been implemented for a robot built on the LEGO Mindstorms NXT, an old version of a good known toy robot. Small resources of robot's controller meant that it would be impossible to implement more complex and computational intensive tasks on it. Since NXT is equipped only with Bluetooth (802.11 or Wi-Fi only in EV3) so in the case of a communication between the workstation and mobile robot we are dealing with

a unreliable physical channel, especially considering the small range and signal attenuation by a wall separating robot and workstation.

Hardware determinants of the implementation of middleware on the robot side were as follows:

- processor with ARM7 architecture (Atmel AT91SAM7S256), 64kB RAM and 256kB FLASH,
- Bluetooth 2.0 with implemented only virtual serial port profile RFCOMM,
- nxtOSEK operating system [16] instead of original NXT firmware (because of its limitations).

It was decided that the service-oriented approach and the connection mode of communication should be used in the project – services on the robot and clients on the workstation side, providing interfaces for tasks of the parent layer in the agent architecture to be executed on the workstation. For a connection workstation-robot can theoretically exist any number of pairs client-service – the limitation is caused by limited resources on the robot side. The project provides two types of such pairs:

- for packet communication – it involves the short length messages that can be sent in a single frame,
- for mass communication – if the message is so long that it must be divided into a larger number of frames.

Robot can have connection with a single workstation. But in order to facilitate experiments with a robot team one workstation can be connected with more than one robot as shown in Fig. 2. This can be used e.g. to test executing cooperation scenarios.

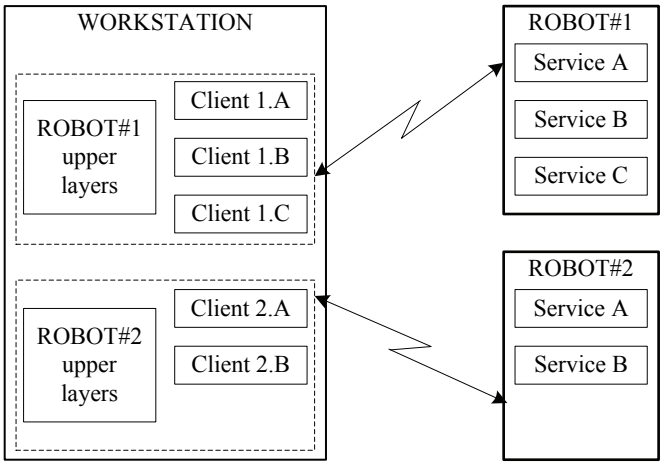

Fig. 2. Middleware – services and clients

Middleware protocol stack on top of NXT implemented Bluetooth layers is shown in Fig. 3.

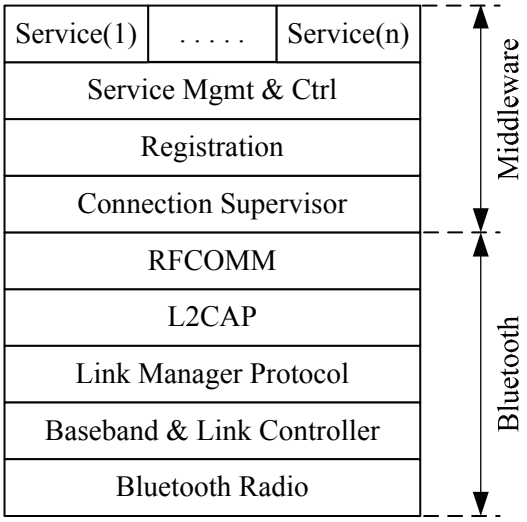

Fig.3. Middleware protocol stack (robot side)

Bluetooth layers can be briefly described as follow, starting from the bottom:

- Bluetooth Radio defining frequency bands and hopping, modulation technique and power classes. Its functionality partially covers the Physical Layer in OSI model.
- Baseband Layer and Link Control specifying the lower level operations at the bit and packet levels (e.g. forward error correction, cyclic redundancy check calculations and Automatic Repeat Request Protocol).
- Link Manager specifying connection establishment, authentication, release of the asynchronous connectionless or connection-oriented (ACL) and synchronous connection-oriented (SCO) channels, traffic scheduling, link supervision, and power management tasks.
- L2CAP – Logical Link Control and Adaptation Protocol layer is an interface to data transport protocols, and it controls the multiplexing of higher layer protocols and the segmentation and reassembly (SAR) of large packets.
- RFCOMM is a simple transport protocol, with additional provisions for emulating of RS-232 (EIATIA-232-E) serial ports [17]. It is based on ETSI standard TS 07.10. This is the only transport protocol available by MINDSTORMS NXT – but on top of RFCOMM can be found other adopted network protocols, like UDP/TCP-IP, not available in this case.

A comprehensive documentation of Bluetooth architecture and its protocols can be found on the web portal [18].

The following Table 1 shows the middleware layers of the protocol stack and their basic properties.

Table 1. Properties of layers within the middleware on robot side

Layer	Functionality	Provided interface
Service	– Data presentation – Management of service configuration – Ensuring the reliability of data packet transfer – Retransmission of data if necessary	– Sending data with the specified priority with or without confirmation – Sending a new service configuration
Service Management and Control	– Addressing messages to services and clients – Management of priorities – Starting new services	– Sending data with the specified priority – Client registration
Registration	– Robot connection registration – Sustaining session if connection lost	– Notifications about session registration – Sending packets in the context of the current registration – Notification of the completion of the session
Connection Supervisor	– Monitoring the status of connection – Ensuring the reliability of transmission	– Sending packets within the connection – Reliable sending control packets – Notification if connection broken and about reconnects

In the test implementation [19] for Mindstorms NXT the middleware was placed between layers (2) and (3) of the agent architecture as shown in Fig. 1 at position pointed by "b". For this location of the middleware following services were introduced:

i. Choosing from the library, activation and termination of the state machine representing the robot behavior.
ii. Submitting a new state machine to the library or deleting existing one from the library.
iii. Direct control of the robot using orders for layer (1) – by this time the activity of layer (2) is suspended.
iv. Cyclical sending frames containing robot status (including robot sensors data).

The implementation of middleware on the robot side was to volume just below 26kB of FLASH memory (from 256kB), leaving sufficient space for the code and library data of the two lowest layers of system architecture.

The correctness of the implementation and assumptions of the project have been confirmed by developed functional tests which covered:

- performance tests associated with transmission speed,
- message queuing based on prioritization as well as for the case of transmission errors (artificial injection),
- detection of connection break and connection recovery.

4. CONCLUSIONS

In this paper a middleware project for an autonomous mobile robot was presented. The middleware is related to the layered agent architecture, but it can be placed at different levels of this architecture according to needs and available hardware resources of the robot's on-board computer. The motivation of the project was to allow developing and experimenting with variety of EMAS problems in a physical environment at a very low cost of lab equipment.

The proposed middleware allows to execute selected tasks in the environment of an external, stationary computer instead of robot on-board system. This is helpful while testing the code and can ease the experiments with various versions of solution. Designed middleware can be placed at different levels of the system architecture through the exchange of interfaces of its application layer protocol. Furthermore, the problem of communication channel reliability was taken was taken into account. So, if the connection is broken, appropriate actions may be taken in order to restore it and, in the meantime, the robot can continue the actions previously performed or start another activity provided for this situation. This is because the middleware has an embedded notification mechanism of a loss of connection and its automatic recovery.

ACKNOWLEDGMENTS

I would like to thank Paweł Marchewka, for his work by implementation and testing of NXT-Bluetooth version of the middleware project. I also would like to express my gratitude to Professor Janusz Sosnowski who previewed this paper and corrected my mistakes.

REFERENCES

[1] WOOLDRIDGE M., *An introduction to Multi Agent Systems*, John Wiley & Sons, 2009.
[2] WEISS G. (Editor), *Multiagent Systems*, MIT Press, 2013, 2nd edition.

[3] COLLETT T. H., MACDONALD A., GERKEY B. P., *Player 2.0: toward a practical robot programming framework*, In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA '05), Sydney, Australia, 2005.
[4] The Player Project: http://playerstage.sourceforge.net
[5] UTZ H., SABLATNÖG S., ENDERLE S., KRAETZSCHMAR G., *Miro – Middleware for Mobile Robot Applications*, In: IEEE Transactions On Robotics And Automation, Vol. 18, no. 4, pp. 493–497, 2002.
[6] CÔTÉ C., BROSSEAU Y., LÉTOURNEAU D., RAÏEVSKY C., MICHAUD F., *Robotic software integration using MARIE*, In: International Journal of Advanced Robotic Systems, Vol. 3, No. 1, pp. 55–60, 2006.
[7] YOO J., KIM S., HONG S., *The robot software communications architecture (RSCA): QoS-aware middleware for networked service robots*, In: in Proceedings of the International Joint Conference (SICE-ICASE '06), pp. 330–335, October 2006.
[8] NESNAS I. A. D., *The CLARAty project: coping with hardware and software heterogeneity*, In: Springer Tracts in Advanced Robotics, vol. 30, pp. 31–70, Springer, Berlin, Germany, 2007.
[9] NESNAS I. A. D., SIMMONS R., GAINES D. et al., *CLARAty: challenges and steps toward reusable robotic software*, In: International Journal of Advanced Robotic Systems, Vol. 3, No. 1, pp. 023–030, 2006.
[10] Official website of CLARAty project: https://claraty.jpl.nasa.gov
[11] Microsoft Robotic Developer Studio official website: http://msdn.microsoft.com/en-us/library/dd939239.aspx
[12] ELKADY A., SOBH T., *RoboticsMiddleware: A Comprehensive Literature Survey and Attribute-Based Bibliography*, In: Journal of Robotics Vol. 2012, Article ID 959013, Hindawi Publishing Corporation.
[13] BROOKS R. A., *A Robust Layered Control System for a Mobile Robot*, A.I. Memo 864, MIT AIL 1986.
[14] FERGUSON I. A., *TouringMachines: an architecture for dynamic, rational, mobile agents*, Technical Report Number 273, University of Cambridge Computer Laboratory 1992, ISSN 1476-2986
[15] FISHER K., MÜLLER J. P., PISCHEL M., *Unifying Control in a Layered Agent Architecture*, Technical Report TM-94-05 from DFKI GmbH.
[16] nxtOSEK home webpage: http://lejos-osek.sourceforge.net
[17] BLUETOOTH SPECIFICATION Version 1.1. Part F:1, RFCOMM with TS 07.10. Serial Port Emulation, 2003.
[18] https://www.bluetooth.org
[19] MARCHEWKA P., *Design and Implementation of Communication protocol in Embodied Agents Systems*, M.Sc. Thesis, II PW 2013 (in Polish).

UML, code generation, C#,
generation gap

Anna DEREZIŃSKA*, Karol REDOSZ*

REUSE OF PROJECT CODE IN MODEL TO CODE TRANSFORMATION

Model to code transformation can be performed many times during a development process based on the Model Driven Engineering ideas. A code project generated from structural as well as behavioral models can be further extended with program details, e.g. method bodies. A straightforward regeneration of an updated model could result in overwriting of the implemented parts of a project. Therefore, it is beneficial to reuse a code originated from the previous project during a model to code transformation realized in a next development stage. Different approaches to code generation gap are discussed in the paper. One of approaches, based on reuse of elements of non-modified classes, is specified in terms of reuse criteria devoted to different programming notions. The proposed criteria were implemented in an extended version of the Framework for Executable UML tool (FXU). The criteria were evaluated in experiments on UML models transformed into C# programs.

1. INTRODUCTION

In a Model Driven Engineering (MDE) approach, a model to code transformation takes a significant role [1]. In typical MDE processes a structural model, described for example by class diagrams, can be transformed into a code project. Moreover, some behavioral specifications built with CASE tools, e.g. state machine diagrams, can be used as source models in the code generation [2–5].

However, a detailed code is usually supplemented at the code project level using a development environment, which is more convenient for programmers. This kind of code refers, for example, to bodies of selected methods.

The problem arises when a model is modified and its transformation is repeated. A simple model to code transformation would provide to losing of an implemented

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

code, hence it is not a satisfactory solution [6–8]. It could be beneficial to have full roundtrip engineering, but so far it is available only for some solutions based on structural models, mainly when a class diagram is generated by reverse engineering from an existing code. The reverse engineering for code generated from more extended models, including state machines, is more sophisticated and ambiguous.

In this paper, we address the code generation gap problem [8]. We present different approaches and propose code reuse criteria for a selected solution. It is assumed that a model to code transformation process is mostly automated, without need of a user interaction. The selected approach was implemented in the Framework for Executable UML (FXU) [9,10], which is a tool for generation of C# code from UML class diagrams and state diagrams. Experiments were carried out to verify and tune the criteria [11].

This paper is organized as follows. We present the gap generation problem and different solutions in the next Section. In Section 3 code reuse criteria are introduced. We discuss experiments with FXU in Section 4 and conclude the paper in Section 5.

2. APPROACHES TO CODE REUSE

The following assumptions were made about the considered MDE process. A model M^{-1} was automatically transformed into a code project. The project was further extended with some code extracts, e.g. bodies of methods or constructors were added; and some elements like fields, methods, parameters were added or deleted. The current updated project will be denoted as *P*.

In the next development stage, a new modified version of the model was prepared. It is called *M*. The model *M* will be transformed into a new code project *P'*. The transformation uses also data from the project *P,* merging information from *M* and *P*.

We considered four approaches (A-D) taking into account the following features:

– reuse of the code from the source project *P* in the target project *P'*,

– sources of potential compilation errors because of reuse of the existing code from *P* together with the new generated code from *M*,

– involvement of a user in the selection of model/code elements to be included into the final project *P'*.

2.1. GENERATION OF THE MODIFIED CLASSES ONLY (A)

A model to code transformation is performed only for classes that are new or were modified in the model *M*. It is necessary to determine which class was modified. It can be established by a user or automatically. Automatic detection of a class modification can be realized in different ways. A simple solution is comparison of the previous

model M^1 with the modified model M, but it can be inconvenient for a user to maintain both models. Another method is usage of information that identifies the previous model and is stored in the code project in the form of additional comments or annotations. A user should only not to interfere into this part of the code.

Advantages of the approach are: a simple implementation and little sources of potential errors. A disadvantage is a fact that a class with a simple change will be classified as modified, and the whole class will be substituted. In an extreme situation, a change of a class name will provide to rewriting of the class files.

2.2. GENERATION GAP PATTERN (B)

The problem is addressed also by a design pattern called *generation gap pattern*. Each class of a model is transformed into two classes related by inheritance. The base class is an abstract class that comprises elements generated from a model. No other elements are included in the class and a user should not modify it. A user works on the second class, inherited from the base one. Typically base and inherited classes are stored in different packets and/or folders with some meaningful names, such as *src-gen* and *src-once* [7, 8].

A positive of the approach is that automatically generated code is explicitly separated, and manually implemented code fragments can be easily reused. The regeneration process is fully automated. A user should only follow the implementation rules and do not modify the base classes.

A drawback is the high number of classes, duplicated in comparison to the model. The generated code is also less comprehensible. Using of a class hierarchy in a model can cause problems of merging inheritance relations from two origins: a conceptual model and the design pattern. It is especially questionable in languages that do not support multiple inheritance, as Java or C#. Repeated generation of base classes can produce errors in the inherited classes, e.g. an inherited class uses an element of the base class which was removed in the modified model.

2.3. REUSE OF NON-MODIFIED CLASS ELEMENTS (C)

Elements that were not modified in a model are reused in the target project. This approach is similar to A), but takes into account different elements not only the whole classes.

There are different ways to specify that an element was modified. An automatically generated code can be separated in a class using comments or annotations. Therefore the file structure of a project is not affected. For example, elements generated by IBM Rational Software Architect [12] in UML to Java transformation are labelled with *@generated* annotations. Code modifications are substituted by the newly generated

code based on the current model. However, those parts of method and constructor bodies that are placed between comments “begin-user-code” and “end-user-code” are not substituted.

Moreover this approach requires criteria about reuse of various kinds of elements. One solution is a selection of only simple elements, e.g. a method or constructor body, that have a non-modified signature. However, a code extract to be copied can have range of references to other elements of the same and of different classes. A simple coping without strict rules can provide to errors. On the other hand, too severe restrictions result in the losing of too much of the code that will be not reused.

An advantage is possibility to reuse extracts of the code (methods, constructors) of classes that were partially modified. Using some selection criteria, the transformation process can be realized without a user interaction.

The approach has several disadvantages. Change of a class name in a model provides to a substitution of the whole class and its implementation. Non-modified elements of a class that were copied can depend of other elements that were modified in the model. Therefore, there is a risk of errors due to reused code. In comparison to the A) approach this solution is more complex in terms of reuse criteria and implementation.

2.4. FULL MERGING OF CODE WITH MODEL (D)

In the first three approaches we assume the fully MDE approach, i.e. a model is correct and superior in the code generation process. A modified model is used as a reference in the selection of elements to be generated in the revived project. Therefore, elements that exist in the previous code project but are not included in the new version of the model should be omitted. In result the model should be complete in the reference to the current stage development and include all elements indispensable in the project.

The fourth approach is based on different assumptions. A model and a current code project include complementary parts of the development that can be merged into the final project. A user should know which parts of the previous model can be reused. A merging engine can inquire a user about doubtful elements and automatically handle elements encountering in both a modified model and a previous code project. There are also different policies based on the automating merging rules:

I) *A model is superior.* An analysed element is generated if exists in the current model M, regardless weather it has a corresponding element in the project P.

II) *A code project is superior.* An element is copied from P to the target project P'. If an element exists only in the model it is omitted during the code generation.

III) *A model supplements a code project*. Each element from the project P is reused in P'. New elements, not present in P are generated from the model M.

IV) *Model and code project are equally important* and their intersection is a valid development. If an element is used in the model M and the project P then the element is reused in P'. Otherwise the element is omitted.

An advantage is reusing parts of the code (methods, constructors) of classes that were partially modified. Deciding of code extract reuse is highly flexible. An amount of code that would be lost in the project re-generation can be minimized.

However, this approach has much more complex implementation than the previous ones. In several policies a user is involved into the generation process. There are high requirements on the model and the code comprehension. There are many possible sources of conflicts and errors due to merging of inconsistent code originating from a model and a project.

3. EVALUATION CRITERIA FOR REUSE OF CODE EXTRACS

General policies discussed above have to be companioned by specific criteria that try to avoid situations of the inconsistent code. While extending the Framework for Executable UML (FXU) [10], the approach C) was selected to be implemented. It is a compromise solution that supports an automatic selection of code extracts to be reused. It is not necessary for a user to have a complete knowledge about a previously generated project or modifications that were introduced into a model.

In general a modified model M was treated as a superior correct model. It was assumed that if an element existed in the previous model M^{-1} but was omitted in M it was deleted by a user on purpose. In the reuse mechanism it was assumed that each system change will be introduced in a model. An exception is a constructor, as in FXU constructors are not generated from a model.

The following proposed criteria can be in the most cases applied to any object-oriented language. However, they were designed in the context of the C# language. The criteria were specified precisely using formulae. As an example, one formula for classes is shown. The remaining are omitted due to brevity reasons.

A packet encounters in the most of programming languages, including C#, represented by a folder structure. A packet is an important structural element. A packet P_M defined in a model is equivalent to a packet P_P generated in project if the following conditions are met:

– both packets have the same names, both packets are not included in any superior packet, or are included in equivalent packets.

An interface includes declarations of methods. The methods should be implemented in a class (or classes) that realize them. Assuming that a model is complete there should be no implementation elements that are not present in the model. Therefore

interfaces are generated from models and do not have special criteria for matching with the code.

A class is structurally considered as a configuration of various sub-elements. A class C_M defined in a model is equivalent to a class C_P generated in project if the following conditions are met:

– the classes are comprised in equivalent packets,

$$((C_M \in CLASSES(P_M)) \wedge (C_P \in CLASSES(P_P)) \wedge (P_M \equiv P_P)) \quad (1)$$

where $CLASSES(P_X)$ means a set of classes in the packet X, and $\equiv$ denotes equivalence.

– the classes have the same names, and have the same descriptors, e.g. *abstract.*

It should be noted that the criteria do not take into account inheritance or interface realisation. It was assumed that these conditions would be too restrictive.

An attribute can be fully generated from a model. It can have a type and an initial value that is initialised in its definition or in a constructor. A problem erases when attributes are created or deleted in implementation. Such modifications should be reflected in a model. It can be done by hand or by simple reverse engineering facility.

For the further merging step, the following criteria state that an attribute A_M from a model is equivalent to an attribute A_P from a project, if:

– the attributes belong to the equivalent classes, have the same names, are of the same types, have the same visibility, and have the same descriptors, e.g. *static*, *abstract*.

A constructor is typically created as a default constructor**.** This constructor can be supplemented in implementation, and/or additional constructor can be developed. They can be used in a new project. A constructor is copied from a previous project to a new one, if its class in a model includes all attributes with the same names and types, as attributes defined in the equivalent class in the project.

A method has, apart from its signature, a body that is usually not represented in a model. A method code can be written in a model, but it is often inconvenient as it is not verified on-line. Generating a new code from a model can lead to losing an implemented code. Therefore method code from a project should be reused in the new project after model to code transformation. A method M_P from a project can be merged with a method M_M defined in a model, if the following criteria are met:

– the methods are in the equivalent classes, have the same name, have the same return type, have the same visibility, and have the same descriptors, e.g. *static*.

– the methods have parameters of the same names and types, but the parameter order can be different.

However, there also exist special methods, e.g. methods generated to reflect behavioral models that should be not modified by a user. They are generated from a model disregarding the above criteria.

4. EXPERIMENTS

4.1. FXU – FRAMEWORK FOR EXECUTABLE UML

The Framework for eXecutable UML (FXU) is an environment for generating C# programs from UML models [10]. The first version of the tool was created in 2006 [9]. It consists of two parts FXU Generator and FXU run-time Library. FXU Generator transforms UML classes and their state machines into a C# project. The project includes skeletons of the classes, the code reflecting structures of the state machine elements and references to library items. The Library comprised the code corresponding to realization of all state machine elements used in run-time of an application. Further versions of FXU were developed, which among others support extended GUI, model tracing interfaces, and different UML semantic variants [13], [14]. The 5th version of the FXU generator was enhanced with the code reuse facility [11].

4.2. EXPERIMENT SETTINGS

In the discussed process, modified models are transformation sources in consecutive development stages. Therefore a simple *modification process benchmark* was proposed. The following basic actions can be distinguished in a process: addition, removal and modification. Each of these actions can be applied to different software elements – action subjects: a packet, class, class attribute, method and class constructor.

Considering the program generation cycle, there are three possible areas when an action can be completed: in a model only, in a code project only, both in a model and in the code project. Combination of the above variants (i.e. action type, action subject and application area) constitutes a modification process benchmark used in the experiments.

The experiments were evaluated in terms of the usefulness of the implemented approach and the applied criteria. Therefore some metrics were proposed to evaluate the process in a quantitative way. Utilization of the code from the project (P) during the generation of the new project (P') was calculated by metrics:

E – a rate of the code added to project P but not reused in the project P' and R=100–E – a rate of the code added during project P implementation and reused in P',

$$E = \left(\frac{PLOC - GLOC_{mrg}}{PLOC - GLOC_{all}} \right) * 100 \tag{2}$$

where $PLOC$ – number of code lines in the P'

$GLOC_{all}$ – number of code lines generated in P', when no reuse mechanism was applied,

$GLOC_{mrg}$ – number of code lines generated in P', when the reuse mechanism was applied.

Lines of code were calculated without lines of comments and white lines.

The experiments were performed on two models, so-called a *training model* and a *test model*. As a training model we used a model of a control system of an orthotic robot [11]. The model was enhanced in order to cover various elements of state machines modelled in UML. This model was used as a benchmark during evaluation of transformations of a UML class and state machine model to code. Transformations carried out by three tools were experimentally compared, namely FXU, IBM Rational Software Architect and IBM Rhapsody [11]. In the experiments referred in this paper the same model was applied for the preliminary verification and tuning of the reuse criteria.

Preliminary version of reuse criteria were implemented in FXU and applied to the training model. After experiments, some criteria concerning reuse of constructor and methods were modified. The criteria described in Section 3 cover already those modifications.

The second model used for the final testing of the discussed code generation approach was based on an open source C# project. It was published in the GitHub service an open source repository [15]. This is a SharpUnit library devoted to creation of unit tests in the C# language. Therefore it was a real commonly used project developed independently of the FXU environment and experiment participants. Its UML model was generated by reverse engineering methods, using facilities included in the IBM Rational Software Architect environment [12].

Both models were modified by different actions according to the modification process benchmark.

4.3. EXPERIMENT RESULTS

In Table 1, we presented how performed modifications influence on the rate of the code reuse. Only such modifications were considered that have an impact on conditions deciding on an element reuse. For example, a method modification in which only an attribute order is changed is not taken into account.

The results of the test model showed that the automatic reuse was applied to almost 90% of the code. This is substantial improvement in the comparison to the pure model re-generation, but still it is not fully satisfactory.

Merging of code is threatened by inconsistency errors in the final project. A source of errors can be deletion or modification of an attribute in a model that was used in a method body in the previous project. The lines of such kind are not counted as reused in the R and E measures. It was observed that R rate was therefore lowered of 0.2%.

Table 1. Reuse rate (*R*) of implemented code during model to code re-transformation

Modification type (action, subject)	Modification area		
	In model	In project	In model and project
No changes	89.3 %	89.3 %	89.3 %
Attribute add	88.8 %	88.6 %	88.8 %
Attribute delete	87.7 %	89.6 %	89.8 %
Attribute modify	87.0 %	86.8 %	89.3 %
Method add	87.1 %	86.4 %	89.5 %
Method delete	89.1 %	88.3 %	89.1 %
Method modify	86.8 %	86.1 %	89.3 %
Constructor add	-	89.6 %	-
Constructor delete	-	89.0 %	-
Constructor modify	-	89.3 %	-

Calculated values of metric *E* present a distribution of code extracts that have to be implemented in a new code project. If a project is generated from a non-modified model the most of the lost code (about 7.4%) was associated with attributes. This effect was caused by C# specific way of defining *getter* and *setter* accessors, which was not considered in the presented reuse approach. The modification of the criteria of attributes would improve this result.

The rest of elements that should be re-implemented referred to: methods 1.92% of code, imports 1.1% and constructors 0.27%. These code losses were caused by few limitations of FXU generator, i.e. some C# elements are not generated, like virtual descriptor and calling of a base constructor in a constructor of an inherited class.

The similar analysis of a lost code (*E*) was performed for different modifications of a model. In all kinds of modifications the highest code loss (about 7%) was for attributes, because of the above mentioned reasons.

The code loss of method bodies and constructors depends on the modification type. It can be caused by an attribute modification, attribute deletion or method modification. Modifying or deleting an attribute affects a part of a constructor where the attribute is initialized. A method modification results in overwriting its body. The problem can be omitted if modification is introduced in the model and in the project.

5. CONCLUSION

An approach to code reuse in a model to code transformation was proposed and experimentally evaluated. First experiments were applied to the tuning of criteria of programing element reuse. Further experiments, performed on an independently de-

veloped UML model, shown that about 90% of the implemented code were reused and merged with the code automatically generated from a modified model.

However, the approach needs further improvements. We have observed that accessors created for attributes, known as getters and setters, were not automatically reused. This problem can be solved by extending given reuse criteria. Moreover, further experiments should refer to a more complex modification process. A model can include many various modifications, which should be verified during model to code transformation.

REFERENCES

[1] FRANCE R., RUMPE B., *Model-driven Development of Complex Software: A Research Roadmap*, In: Proc. of Future of Software Engineering at ICSE 2007, pp. 37–54. IEEE Soc., 2007.

[2] DOMINGUEZ E.; PEREZ B; RUBIO A. L.; ZAPATA M. A., *A systematic Review of Code Generation Proposals from State Machine Specifications*, Information & Software Technology, vol. 54, no. 10, 2012, 1045–1066.

[3] EDWARDS G.; BRUN Y.; MEDVIDOVIC N.: *Automated Analysis and Code Generation for Domain-Specific Models,* In: Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture, 2012, pp. 161–170.

[4] DANG F.; GOGOLLA M.: *Precise Model-Driven Transformations Based on Graphs and Metamodels, sefm,* 2009 Seventh IEEE International Conference on Software Engineering and Formal Methods, 2009, pp.307–316.

[5] NIAZ I. A.; TANAKA J.: *An Object-Oriented Approach To Generate Java Code From UML Statecharts*, International Journal of Computer & Information Science, Vol. 6, No. 2, June 2005, 83–98.

[6] FOWLER M., *Domain-Specific Languages*, Addision-Weslay Prof., Boston, 2010.

[7] VLISSIDES J., Pattern hatching: design patterns applied, Addison-Wesley, 1998

[8] BEHRENS H., *Generation gap pattern*, 23 April 2009, [online] http://heikobehrens.net/2009/04/23/generation-gap-pattern/

[9] PILITOWSKI R., DEREZIŃSKA A., *Code Generation and Execution Framework for UML 2.0 Classes and State Machines*, In: T. Sobh (Ed.) Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, Springer, 2007, 421–427.

[10] FXU Framework for eXecutable UML V5.0 Feb 2014, http://galera.ii.pw.edu.pl/~adr/FXU/

[11] REDOSZ K., *Automatic code generation from UML models – development of the FXU tool*, Master Thesis, Institute of Computer Science, Warsaw Univ. of Technology, 2014 (in Polish).

[12] IBM Rational Software Architect, http://www-03.ibm.com/software/products/en/ratisoftarch [access May 2014]

[13] DEREZIŃSKA A., SZCZYKULSKI M.: *Tracing of State Machine Execution in Model-driven Development Framework*, In A. Konczakowska, M. Hasse (Eds.), IEEE Explore Proc. of the 2nd Int. Conf. on Information Technology, ICIT'2010, 2010, 109–112.

[14] DEREZIŃSKA A., SZCZYKULSKI M.: *Interpretation Problems in Code Generation from UML State Machines - a Comparative Study*, In: T. Kwater (ed.) Computing in Science and Technology 2011: Monographs in Applied Informatics, Dep. of Applied Informatics Faculty of Applied Informatics and Mathematics Warsaw Univ. of Life Sciences, 2012, 36–50.

[15] GitHub , https:github.com/mgants4/SharpUnit [downloaded 2013]

authentication, data security, human factors
in data security, risk management

Tomasz BILSKI*

SOME REMARKS RELATED TO HUMAN BEHAVIOUR IMPACT ON DATA PROTECTION PROCESSES

The purpose of the paper is to show some examples of human behaviour impact on data protection and to try to determine a set of different sources of human negative impact related to such security areas as: general models, risk analysis, authentication (with special emphasis on some innovative methods and tools). Mobile devices and wireless transmission systems are of particular interest here. Unfortunately, some security researchers focus only on technological issues and qualitative measures (e.g. encryption key length, password length, fault rate) of data security. Human factor should not be excluded. A lot of research is dedicated to a single security factor, e.g. a new protection method, security protocol. Here, we try to incorporate more complete approach – discussing technology issues together with human behaviour. The results show that there are many different sources of human destructive impact on data protection. They are related to awareness, human behaviour, errors in risk perception, lack of security policy compliance. The paper should be of interest to many different groups of people, among them are researchers, designers, policy makers and users of security systems. The paper links different findings and research results from diverse areas: models of security systems, standards, technology, behavioural sciences. Linking the diverse areas we may increase security level. The paper offers a useful starting point for further research in the area.

1. INTRODUCTION

Researchers from the security area frequently focus mainly on technological solutions to eradicate vulnerabilities and to prevent attacks or accidental faults. They have not yet fully adopted a sociotechnical methodology that addresses human behaviour and also organizational facets of the problem. It should be noted here that there are some exceptions from the rule, nevertheless the majority of research is concentrated on technological aspects of security. It is common that the problem is studied from

* Institute of Control and Information Engineering, Poznań University of Technology. The work is supported by research project 04/45/DSPB/0105.

a single point of view. There are many examples. *De facto* standard quantitative security measures like: encryption key length, password length, RTO (Recovery Time Objective), RPO (Recovery Point Objective), MTBF (Mean Time Between Failures), packet error rate, BER (Bit Error Rate) for communication channels and hash function collision probability are essential. Nevertheless, they are scarcely related to human behaviour. Regardless of the values of security measures, if human factors directly or indirectly affect the design, implementation and use of security controls, the overall effect on security can be detrimental.

We observe continuous advances in security protection technology at the same time statistics show that human errors continue to be a major source of concern in the field of organizational information security.

Requests for more examinations in various areas are regularly mentioned in scientific society. Furthermore, linking the miscellaneous areas we may increase security level. The role of human factors in the data security has many diverse aspects. It is analysed by scientists from such areas as: psychology, cognitive science, computer science, management science. An example of such multifaceted efforts is Annual Interdisciplinary Workshop on Security and Human Behavior. The Workshop is a meeting of psychologists, computer security researchers, sociologists, behavioural economists, philosophers, political scientists, lawyers, anthropologists, business school professors, neuroscientists[1].

The problem is also in the area of interest of standardization bodies. Let us mention a single example: appendix A.7 of standard ISO/IEC 27001:2013 "Information technology – Security techniques – Information security management systems – Requirements" [5] is related to human resource security. The appendix defines 6 controls that are applied before, during, or after employment.

Data security is a domain, first of all, of: scientists, programmers, IT staff, company management. These groups of people are relatively well prepared to provide high level of security. Nevertheless, ordinary IT users are also involved in a growing number (see section 4.2) of processes related to data protection. It must not be assumed that this group of people is well prepared to data protection tasks. They are often careless, they do not have appropriate IT background. The increasing popularity of powerful personal electronics with wireless communication abilities and IT systems based on BYOD (Bring Your Own Device) rule and some innovative, human assisted protection methods have made the problem more urgent than ever before.

Humans are involved in many phases of data protection: modelling, development of methods and tools, software development, risk management, planning of security system, making decisions related to selection of particular tools and methods, implementation, exploitation. Involving humans in any process means:

[1] http://www.cl.cam.ac.uk/~rja14/shb14/

- increased time of operation,
- increased error rate of operation, especially in the case of persons without IT background.

In general, computers are much faster and more accurate than humans[2] – this drawback of human actions will not be discussed here. There are many examples of human errors related to data security: significant number of bugs in software, errors in risk assessment, errors in security tools and protocols implementations (e.g. weak passwords, lack of backup copies). The accidental causes (i.e. human errors) are not the only problem, deliberate security violations should also be included in a complex security analysis. Human behaviour impact is usually qualitative, in just a few cases we may provide quantitative data [7].

Due to the limited size of the paper we will focus on selected examples related to human, accidental behaviour impact on data protection processes. The examples show qualitative data (e.g. related to risk analysis) and quantitative data (e.g. related to authentication processes). A lot of research has been carried on user authentication methods. Weaknesses and drawbacks of password systems are very well known. Many modifications to password authentication systems have been proposed in scientific papers in order to make it more secure and more user friendly. Some institutions (like Wi-Fi Alliance or Universal Serial Bus Forum) have also been working on human assisted authentication.

2. GENERAL ISSUES OF DATA SECURITY

2.1. SECURITY MODELS AND PROTOCOLS

There are many models of data security which incorporate human factors. The general rule is that security aspect of IT system is in contradiction to functionality of the system. Furthermore, organizational factors such as e.g. time pressure, high workload escalate a conflict of interest between two basic features of every IT system: functionality and information security [11]. All this implies that better, more secure models and their implementations are hardly ever used.

As an example we may indicate two models related to access control system that are used in operating systems or data base management systems: discretionary (e.g. Harrison–Ruzzo–Ullman model) and mandatory (e.g. Bell–LaPadula model). The

[2] There are some exceptions from the rule. In order to prevent programs from abusing online services CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) are used. Humans are asked to perform a task, such as deciphering distorted characters, that computers cannot accomplish well.

second one is more safe, but at the same time it is more complex and more difficult to use in real IT systems. As a result, common operating systems and data base management systems use discretionary model of access control, which is easier to implement and much simpler. At the same time this model do not prevent some security violations.

A lot of additional examples may be provided, in which more secure options are discarded or introduced only after an extensive delay. Insecure SMTP protocol has been chosen as *de facto* standard for Internet mail instead of more secure (and more complex) X.400. DNSSEC extension (available since year 1997) of DNS system is implemented very slowly – up to today most of country domains are not signed with a use of DNSSEC signatures. It has been demonstrated many times, that simple, short and insecure passwords are commonly selected by users instead of intricate and reasonably long.

All these examples are related to human decisions. Such choices have detrimental impact on general security level. The decisions should be thoroughly evaluated against their negative impact on security. Functionality should not be the only criterion of decision making.

2.2. AWARENESS AND BEHAVIOUR

There are three basic features of general security problem: awareness, behaviour and reasons. Poor information security awareness and behaviour impacts on data security. Awareness is generally recognized as knowledge and understanding of an object, idea or thought. In the security area, that means: user knows threats, vulnerabilities and security controls. It is very important to know real level of security (see Section 3.2). Significant correlations between good password-related behaviours and training and awareness is a well-known fact. Behaviour is generally defined as the actions of a person in the case of particular situation. In the security area, that means: user acts according to the security policy. Reasons are related to culture and rules of enforcements. Culture is determined by participation in a particular social (real or virtual) group or organization. This participation implies specific approach and specific behaviour of a given user. There are also some other problems of cultural issues. For example, it is frequent that employees talk about sensitive data in social portals like Facebook, share data through email, phone, etc. [3].

It is widely known that user behaviour is learned most of all through the consequences of the actions. If the consequences are positive for a given user, then the particular behaviour will be repeated. If the consequences are negative – the behaviour will not be repeated. This aspect of security should be carefully integrated with data security policy of each organization.

3. PROBLEMS RELATED TO RISK MANAGEMENT

3.1. GENERAL REMARKS

In the area of data security there are many classes of events that are non-deterministic, with different level of probability: hardware malfunctions, software error incidents, attacks, hash function collisions, errors and overloads in communication channels, security incidents related to environmental factors. So, there are many issues related to probability theory and risk management.

Risk in computer system security domain is defined as a function (e.g. arithmetic product) of negative event occurrence probability and level (e.g. measured in money loss) of the event impact on IT resources (data, services, hardware). Comprehensive risk management process should include three requirements (confidentiality, integrity, availability), all categories of security violations (human/adversarial, intentional, non-intentional and non-adversarial), all contexts/states (parts of IT system) in which resources may be violated. Many diverse groups of people should be involved in different risk management processes [6]. Risk analysis is a basic step in the security processes. Unfortunately, there are many human-related issues associated with data security risk perception. Let's focus on representative three of them:

- issues related to risk compensation theory,
- the certainty effect in data security,
- the gambler's fallacy and the hot hand.

3.2. RISK COMPENSATION THEORY

Risk compensation theory [16] is related to the level of security that is perceived by people. In general people, that feel protected are willing to take less cautious behaviours. A lot of research has been done in the area of transportation security. In particular, the relationship between human behaviours and the introduction of protective devices, e.g. safety belts, anti-lock braking systems has been examined. It was proved that while the protective devices help improve safety of driver or passengers, they also tend to encourage people to engage in more dangerous behaviours, e.g. speeding. So, the overall safety on roads level is not increased.

The same is true in the context of information security. Implementation of IT security protection mechanisms (e.g. antivirus, cryptography, firewall, …) – may sometimes lead to less cautious behaviours, e.g., users with antivirus software installed may be assured that the level of protection is high. Taking this into account, they may dare to open suspicious email attachments or to visit some flawed websites. Perceived high technological security protection mechanism is destructively related to end-users' intention to comply with data security policy.

3.3. THE CERTAINTY EFFECT

The certainty effect defined by D. Kahneman and A. Tversky [14] in 1986 is associated to human behaviour in the case of some twofold decisions related to perceived risk. The certainty effect is observable in the case of two options: one is nondeterministic another one is deterministic.

In general, significant fraction of people do not take risk to reach big profit if the other option gives 100% certain but small profit. At the same time substantial percentage of people take risk connected with very high losses if the other option means 100% certain but small loss.

In the case of IT security, the certainty effect means that people do not want to pay for security controls – the cost of controls that should be implemented is relatively (to potential losses) small but at the same time it is 100% guaranteed (security controls that are free are usually less functional and weaker than commercial ones). Unfortunately a lot of people prefer to stay without paying for security controls – the security violations and potentially high losses related to them are not 100% assured. It is obvious that lack of controls means very low level of security and increases probability of security violations.

3.4. THE GAMBLER'S FALLACY AND THE HOT HAND

The gambler's fallacy is an incorrect belief in negative autocorrelation of a non-auto correlated random sequence of outcomes. A representative human (gambler) assumes that if something happens frequently during a given period of time then it will happen less frequently in the future or *vice versa*. The belief is false in situations where what is being observed is truly random (i.e. independent trials of a random process).

The hot hand is an incorrect belief in positive autocorrelation of a non-auto correlated random sequence of outcomes like winning or losing. An exemplary human (gambler) assumes that if he (or she) won several times in a row then the subjective probability that he (or she) will win in the future is higher. For people with probability background, it is obvious that the belief is false in situations where the chances of win or lose are truly random.

In the research related to the gambler's fallacy and the hot hand, it was proved that individuals are more likely to make decisions that are based on false assumptions. For example, roulette gamblers are likely to bet on numbers that have recently won. This is partly because these particular numbers are easily called to mind [13].

In the context of information security that means that a lot of people make some decisions related to data security taking into account some incorrect assumptions and beliefs. For example, choosing a particular antivirus software one doesn't take into account important features of the software (e.g. malware detection effectiveness) he or

she chooses the one that is easily called to mind. Of course this aspect of human behaviour is widely exploited in the marketing area.

4. HUMAN ASSISTED AUTHENTICATION METHODS IN WIRELESS SYSTEMS

4.1. AUTHENTICATION IN WIRELESS SYSTEMS

Now, let's focus on one very important security control – user authentication. The problem of weak passwords is widely known (e.g. [12]) and will not be discussed here. Some other problems emerge in mobile IT environment. Authentication (process known also as security association or pairing of the devices) in mobile, wireless environment is a problem reasonably hard to solve. Risk related to threats such as: sniffing, spoofing, MiTM, DoS is relatively much higher in comparison to wired networks. Mobile devices are commonly used in the areas (e.g. financial) that need very high level of security.

Human behaviour related to authentication processes provides us some quantitative data on human impact on security.

Common solutions to the authentication problem are based on cryptography (asymmetric cryptography), Public Key Infrastructure or Trusted Third Party (e.g. IEEE 802.11, IPSec, SSL/TLS). The level of security is high, nevertheless cryptography in mobile environment induces some drawbacks:

- relatively high complexity,
- the processes need extra resources (energy, memory, processor power, communication channel throughput) which are limited in mobile devices and wireless networks,
- prospective replacement of the broken crypto algorithms will be painful and resource consuming task,
- some preliminary relations between parties are necessary, in *ad hoc* networks such preliminary relations usually do not exist.

4.2. OUT OF BAND AUTHENTICATION

In order to solve the problems some new authentication methods (for netbooks, smartphones, …) have been proposed. Among them are methods based on out-of-band authentication channels and human assisted authentication (e.g. protocols based on MANA (Manually Authenticated Strings) [8].

Many out-of-band authentication channels and methods have been suggested in the last years:

- visual (e.g. Seeing is Believing [10], [9]),
- infrared (e.g. [1]).

- acoustic (e.g. Loud & Clear [4]),
- location limited channel (e.g. methods based on NFC transmission),
- physical contact (e.g. [2]).

The general assumption of the methods is that user is an important part of authentication process. User may be involved into the process in two ways:

- user is responsible for verification of the authentication space in order to detect intruder in the physical range of communication,
- user needs to perform some tasks, for example:
  - to compare strings of characters,
  - to compare data from acoustic channel with data written on a label or displayed on a screen,
  - to take picture with smartphone camera of a particular 2-dimensional barcode printed on another device.

In all the examples human is asked to perform a given task and the security level of authentication process is directly dependent on the human part of the process. It must be noted that the users of mobile devices are often without technical expertise. So, we may not assume that the user will act faultlessly. Another important issue is delay of the process. Time of each operation of the user is much longer than time of the comparable operation performed by computer.

4.3. HUMAN ERROR RATE IN AUTHENTICATION PROCESSES

The problem of error rates (quantitative measure) in some actions performed by humans during authentication has been studied by Uzun et al. [15]. In the study comparative usability analysis of different methods has been done. The purpose was to assess the level of security of different operations performed in the human assisted authentication procedures. Simple operations were evaluated: comparison, copying, confirmation and selection. It has been shown that total user error rate in the worst case may reach up to 45%. Even in the case of more user friendly methods error rate reach 5%. Simultaneously, it was demonstrated that the time needed for authentication performed by human may reach near 30 s. In the conclusion the authors indicate that carefully designing (utilizing e.g. results from cognitivity research) human assisted authentication process error rate may be significantly decreased.

3. CONCLUSION

Data security is multifaceted problem. Increased complexity of IT systems means that it is difficult to understand merely technology issues of the systems. Even though we well recognize technology we may not stop here the evaluation of the security. We should not estimate security level with a use of single metrics, like password length or

encryption key length. The metrics are important but should not be evaluated in isolation. Evaluating security of the existing systems and designing new systems we have to incorporate not only technological aspects of the systems. It is very important to take into account many diverse issues related to human behaviour in each phase of the security process, from risk analysis to security controls utilization. Linking the miscellaneous areas we may increase security level. For example, it has been shown that carefully designing authentication process we may decrease human error rate in a meaningful way. We have to repeat an axiom: there are significant correlations between overall security level and human training and awareness.

Whatever enforcement strategy is selected the key (from the human point of view) is repetition and consistency. Human training is very important, but at the same time user behaviour is learned most of all through the consequences of his or her actions. If the consequences are positive for him or for her, then the particular behaviour will be repeated. If the consequences are negative – the behaviour will not be repeated. So, the security policy should be defined in such a way that appropriate behaviour is accurately defined and is reinforced through the system of awards and penalties. Security designers must identify the causes of detrimental user behaviour, and address these to design effective security systems.

Further research in the area is necessary. This is space for cooperative research of scientists with IT background and management background. Human behaviour impact has usually qualitative ration, in just a few cases we may provide quantitative data. So, the research for additional quantitative data is necessary. We need to identify some potential interactions and pathways among different factors that may have both indirect and direct effects on various security issues. More general models of such interactions are also required.

REFERENCES

[1] BALFANZ, D., DURFEE, G., GRINTER, R.E., SMETTERS, D.K., STEWART, P., *Network-in-a-box: how to set up a secure wireless network in under a minute*, [in:] SSYM 2004. Proceedings of the 13th conference on USENIX Security Symposium, Berkeley, CA, USA, USENIX Association, 2004, 207–222.

[2] BICHLER D., STROMBERG G., HUEMER M., LÖW M., *Key Generation Based on Acceleration Data of Shaking Processes*, J. Krumm et al. (Eds.), UbiComp 2007, LNCS 4717, Springer-Verlag Berlin–Heidelberg, 2007, 304–317.

[3] D'ARCY J., HOVAV A., GALLETTA D., *User Awareness of Security Countermeasures and Its Impact on Information Systems Misuse: A Deterrence Approach, Information System Research*, March 2009, Vol. 20, Iss. 1, 79–98.

[4] GOODRICH, M.T., SIRIVIANOS, M., SOLIS, J., TSUDIK, G. AND UZUN, E., *Loud and clear: human-verifiable authentication based on audio*. Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), 2006, 1–10.

[5] *ISO/IEC 27001 – Information technology – Security techniques – Information security management systems – Requirements*, ISO/IEC, 2013.

[6] *ISO/IEC 27005 – Information technology – Security techniques – Information security risk management*, ISO/IEC, 2011.
[7] KRAEMER S., CARAYON P., CLEM J., *Human and organizational factors in computer and information security: Pathways to vulnerabilities*, Computers & Security, October 2009, Vol. 28, Iss. 7, 509–520.
[8] LAUR S., NYBERG K., *Efficient mutual data authentication using manually authenticated strings*. Proceedings of Cryptology and Network Security (CANS), 2006, 90–107.
[9] MAYRHOFER R., WELCH M., *A Human-Verifiable Authentication Protocol Using Visible Laser Light*, Second International Conference on Availability, Reliability and Security (ARES '07), 2007.
[10] MCCUNE J.M., PERRIG A., REITER M.K., *Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication*, International Journal of Security and Networks Special Issue on Secure Spontaneous Interaction, 2009, 4(1–2), 43–56.
[11] SASSE M.A., BROSTOFF S., WEIRICH D., *Transforming the "weakest link" – a human/computer interaction approach to usable and effective security*, BT Technol. J., July 2001, Vol. 19, No. 3, 122–131.
[12] SATER CARSTENS D., MCCAULEY-BELL P.R., MALONE L.C., DEMARA R.F., *Evaluation of the Human Impact of Password, Authentication Practices on Information Security*, E. Cohen (Ed.), Informing Science Journal, 2004, Vol. 7.
[13] SUNDALI J., CROSON R., *Biases in casino betting: The hot hand and the gambler's fallacy*. Judgment and Decision Making, July 2006, Vol. 1, No. 1, 1–12.
[14] TVERSKY, A., KAHNEMAN, D., *Rational Choice and the Framing of Decisions*, The Journal of Business, 1986, 59, 251–278.
[15] UZUN E., KARVONEN K., ASOKAN N., *Usability Analysis of Secure Pairing Methods*, S. Dietrich, R. Dhamija (eds.), FC 2007 and USEC 2007, LNCS 4886, IFCA, Springer-Verlag, Berlin–Heidelberg 2007, 307–324.
[16] ZHANG J., REITHEL B.J., LI H., *Impact of perceived technical protection on security behaviors*, Information Management and Computer Security, 2009, Vol. 17, Iss. 4, 330–340.

user interface, efficiency,
objects arrangements, digital signage

Rafał MICHALSKI*
Jerzy GROBELNY*
Piotr MUSIEL
Jadwiga PRAŹNIKÓW

THE ROLE OF COLOR AND A GRAPHICAL LAYOUT IN INTERACTIVE DIGITAL SIGNAGE SOFTWARE

The main objective of this study is to investigate the how selected graphical factors affect the efficiency of performing three typical scenarios of different complexity in software developed for finding eating places and looking through their offers. The research involves two factors: two interface layouts (based on bars and tiles) and three color versions (red, blue, grey), which resulted in six graphical interface variants. The colors were determined using HSV color scheme with the controlled value of the saturation. In the research 29 student subjects participated. The descriptive statistics of gathered data were provided and. The recorded data were also analyzed by means of analysis of variance.

1. INTRODUCTION

The importance of usability issues is nowadays not only important for the software manufactures or web site developers. An emerging area where graphical user interface is gaining on significance is digital signage. Modern solutions in this field are becoming more and more interactive (Grobelny and Michalski 2011). They take advantage for instance touch sensitive screens, mobile devices (Cheverst et al., 2005) or even hand gestures (Chen et al., 2009). In the digital signage area investigators usually are focused either on preferences toward various design aspects or their impact on the willingness to buy specific products. Since there exists an apparent shift from passive marketing information presentation to more interactive one, it is justifiable to treat them more like graphical user interfaces. If so, they may be

* Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370, Wrocław, Poland.

investigated not only from the subjective point of view (e.g. Grobelny and Michalski, 2011; Grobelny et al. 2013), but additionally in terms of efficiency and effectiveness as it is recommended by the ISO 9241 (1998).

In the present paper we apply this approach and try to analyze two factors differentiating the graphical user interface that can be used in interactive systems supporting digital signage based marketing activities. The rest of the article presents in detail the prepared software and its purpose along with the investigated factors. Next, the gathered data are analyzed and finally discussed.

2. METHOD

2.1. PARTICIPANTS

Students and graduates from the Wrocław located universities took part in the examination. Most of them were from Wrocław University of Technology (21 persons). The age of the participants ranged from 20 to 28 years with the average of 23.8 and standard deviation of 1.53. There were 13 males and 16 females among the subjects, and they were not involved in any professional IT activities, so one may describe them as regular users.

2.2. APPARATUS

A custom-made application was developed to conduct the experiment. Two versions of the software was prepared in the MIT App Inventor (2014) environment initially created by Google Labs, and nowadays supported by the Massachusetts Institute of Technology (MIT). The App Inventor allows for creating simple applications running under the Android platform. The whole development process is requires only the Internet browser and it is not necessary to install any additional components. Apart from designing the graphical user interface, the prepared software also recorded all necessary data regarding the task completion efficiency. The examination was carried out on a Nexus tablet, version 2013, with a 7 inch IPS LCD screen and a 1920×1080 pixels resolution. The device was equipped with a 2 GB RAM and was running under the Android 4.4.2 operating system.

2.3. EXPERIMENTAL DESIGN

Two different layout versions were designed and they are demonstrated in Figures 1 and 2. The first arrangement involved target touchable objects arranged generally in one column where the height was considerably lower than their width. In the second user

interface version, the two column layout dominated. In this case the width and height of the touchable object was comparable and they resembled bars. The two layouts were additionally differentiated by three color versions: red, blue and grey. Different shades of these colors were specified taking advantage of the Hue Saturation and lightness Value (HSV) color space. While selecting red and blue tones, the saturation was controlled. For the grey color, the shades were selected in such a way to match the respective tints of red and blue colors. The full specification of the used colors is provided in Table 1. These two examined factors produced six variants of the experimental applications' user interfaces (2 layouts × 3 colors). The prepared software was meant for finding the route to various types of eating establishments.

Table 1. Color versions of the graphical user interface used in the conducted experiment

	Background			Button1			Button2		
	Color sample	HSV	RGB	Color sample	HSV	RGB	Color sample	HSV	RGB
Red		H = 0 S = 30 V = 100	R = 255 G = 178 B = 178		H = 0 S = 60 V = 100	R = 255 G = 102 B = 102		H = 0 S = 70 V = 100	R = 255 G = 76 B = 76
Blue		H = 240 S = 30 V = 100	R = 178 G = 178 B = 255		H = 240 S = 60 V = 100	R = 102 G = 102 B = 255		H = 240 S = 70 V = 100	R = 76 G = 76 B = 255
Grey		H = 0 S = 0 V = 70	R = 178 G = 178 B = 178		H = 0 S = 0 V = 40	R = 102 G = 102 B = 102		H = 0 S = 0 V = 30	R = 76 G = 76 B = 6

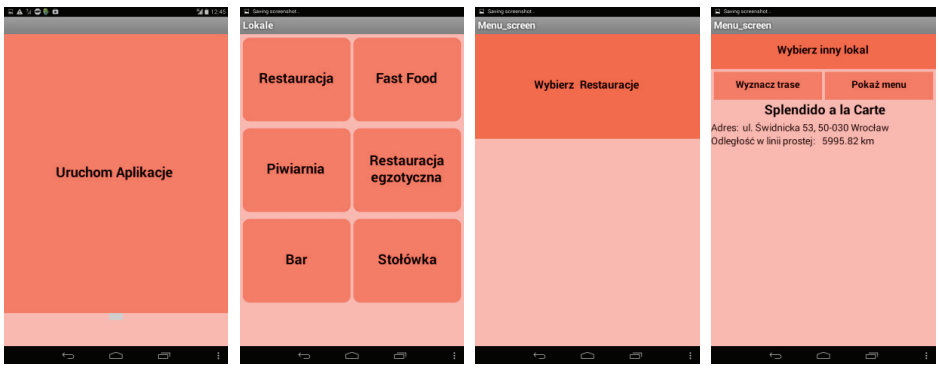

Fig. 1. The first layout of the graphical user interface involving objects arranged as tiles in two columns in a red color version

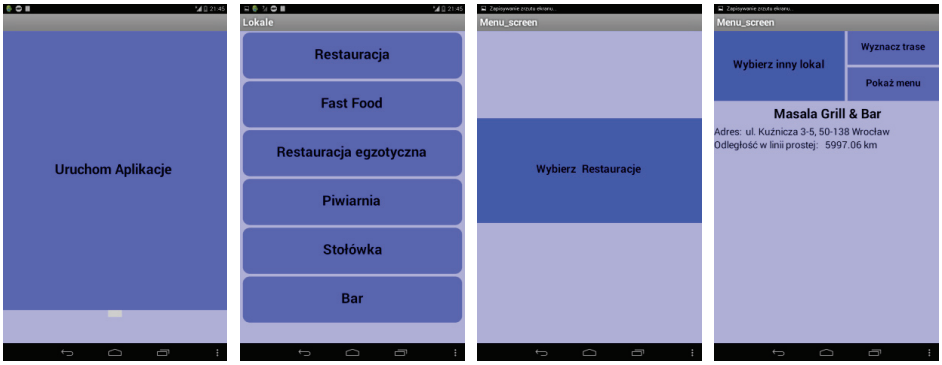

Fig. 2. The second layout of the graphical user interface involving objects arranged as vertically in one column in a blue color version

2.4. EXPERIMENTAL PROCEDURE

For each of the visual interfaces, three typical scenarios were elaborated. They differed markedly in their complexity. The participants were asked for finding a restaurant, checking three restaurants and selecting one of them and, finally, finding available eating places in three kinds of eating establishments. The details about those experimental tasks are as follows:

The first task: Select the "Restaurant" option > Look through all of the possibilities and choose one restaurant > Press the "Show menu" button > Select the option "Select different eating place" > Look through available places and select one of them > Press the button "Determine the route".

The second task: Select the "Restaurant" option > Look through all of the possibilities and choose one restaurant > Press the "Show menu" button > Select the option "Select different eating place" > Look through available places and select one of them > Press the "Show menu" button > Select the option "Select different eating place" > Look through all of the possibilities and choose one restaurant > Press the „Show menu" button > Press the button „Determine the route".

The third task: Select the „Fast food" option > Look through all of the possibilities > Go back to the "Eating place type" menu > Select the „Beerhouse" option > Look through all of the possibilities > Go back to the "Eating place type" menu > Select the "Restaurant" option > Look through all of the possibilities > Select one of the available restaurants > Press the button "Determine the route"

All of the participants performed all three scenarios. If a subject made a mistake, the results were not included in the analysis and the given task was repeated. The software recorded times needed for performing each step of the experimental procedure. Overall times needed for completing individual tasks were also stored and are analyzed in the following sections.

3. RESULTS

3.1. DESCRIPTIVE STATISTICS

The basic descriptive statistics are put together in Table 2 and the calculated means are demonstrated in Figure 3. From these data one can easily observe that the shortest mean times were recorded for the first scenario with a blue color version of the graphical interface and buttons arranged as tiles. On the other hand, the longest time was needed for completing the third task operated in a red version of the software with a bar-like layout.

Table 2. Descriptive statistics for three scenarios and all graphical interface versions

Scenario	Arrangement	Color	Mean	MSE	Median	SD	Minimum	Maximum
Task1	Bars	Red	15.9	1.49	16.0	4.71	10.7	26.4
Task1	Bars	Blue	16.4	2.03	14.6	6.09	10.8	31.2
Task1	Bars	Grey	15.5	1.82	14.1	5.75	9.15	30.3
Task1	Tiles	Red	14.9	1.65	13.8	5.22	8.48	25.4
Task1	Tiles	Blue	13.4	0.890	13.6	2.81	9.87	18.0
Task1	Tiles	Grey	17.4	2.48	16.6	7.43	8.78	35.0
Task2	Bars	Red	23.8	1.12	23.1	3.36	18.6	29.6
Task2	Bars	Blue	25.0	1.49	26.2	4.72	16.1	32.3
Task2	Bars	Grey	21.5	1.72	20.5	5.43	12.8	29.4
Task2	Tiles	Red	20.0	1.40	19.7	4.19	13.8	26.5
Task2	Tiles	Blue	20.2	1.01	20.1	3.21	15.9	24.9
Task2	Tiles	Grey	24.1	2.73	23.3	8.64	15.9	46.5
Task3	Bars	Red	26.8	1.88	26.2	5.96	16.5	35.0
Task3	Bars	Blue	24.9	1.65	24.6	5.23	17.2	32.8
Task3	Bars	Grey	24.2	1.77	23.9	5.30	16.7	33.4
Task3	Tiles	Red	26.1	1.68	24.6	5.32	19.9	38.2
Task3	Tiles	Blue	24.2	1.67	21.9	5.02	17.5	32.1
Task3	Tiles	Grey	24.1	1.57	24.4	4.97	14.6	30.2

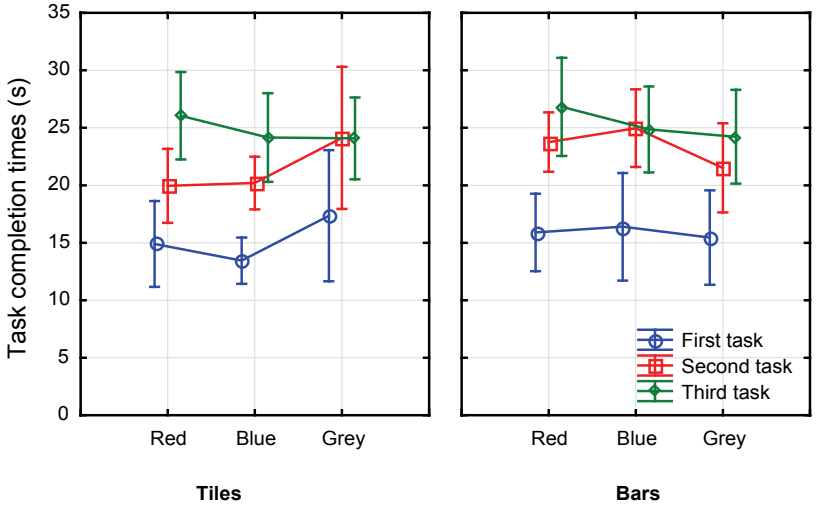

Fig. 3. Tasks completion means for all experimental conditions.
Whiskers denote 95% confidence intervals

3.2. ANALYSIS OF VARIANCE RESULTS

The descriptive results presented in the previous section need to be statistically verified. The three way analysis of variance has been applied for this purpose, and the outcomes are demonstrated in Table 3. The obtained results show that the task completion times strongly varied between the three task types.

Table 3. Three way analysis of variance results

Parameter	Sum of Squares	Degrees of Freedom	Mean Sum of Squares	F – statistics	Probability
Scenario (SC)	2762	2	1381	48	*<0.00001
Color version (CV)	10	2	5.1	0.18	0.84
Arrangement (AR)	50	1	50	1.7	0.19
SC × CV	80	4	20	0.69	0.60
SC × AR	18	2	9.2	0.32	0.73
AR × CV	145	2	73	2.5	**0.083
SC × AR × CV	69	4	17	0.60	0.66

* $p < 0.05$, ** $p < 0.1$.

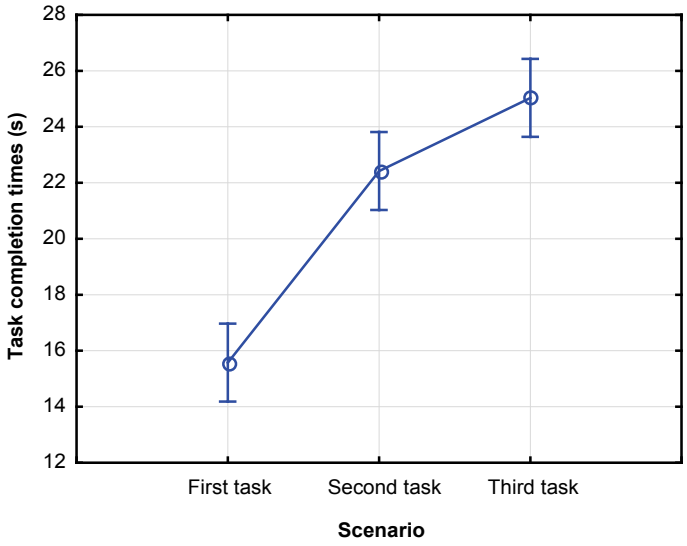

Fig. 4. The effect of different task types on the mean completion times ($F = 48$, $p < 0.0001$). Whiskers denote 95% confidence intervals

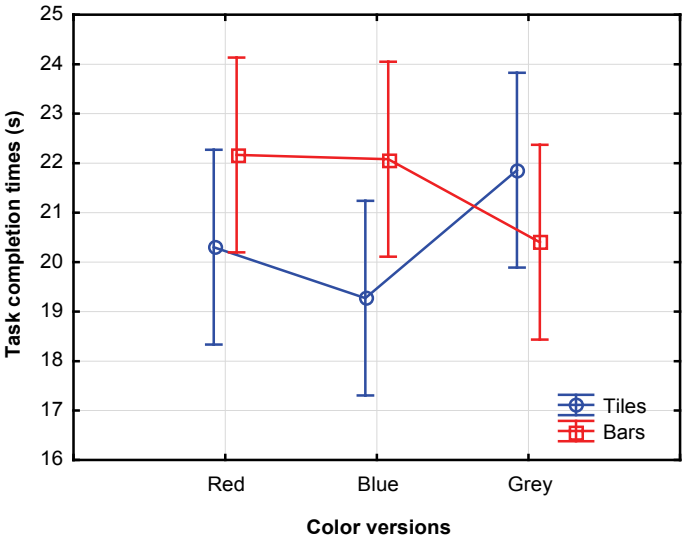

Fig. 5. The effect of interaction between color and arrangement factors ($F = 2.53$, $p = 0.083$). Whiskers denote 95% confidence intervals

The further post-hoc analysis showed that differences between average times of all pairs of the performed tasks were statistically significant at a level of $p < 0.05$. The data are graphically illustrated in Figure 4.

The layout and color factors were not meaningful. From among the interactions only the interaction between the interface graphical layout and its color versions occurred to be statistically significant ($p < 0.1$). The results are presented in Figure 5 and show that the blue version of the interface is the most efficient one when the layout is based on tiles. The grey version of the tiled interface was considerably the worst in terms of the mean time completion times. The situation for bar-like layouts was somewhat different. The grey version was the best while red and blue ones were markedly worse.

4. DISCUSSION AND CONCLUSIONS

The presented data in general confirmed the assumed differences between complexities of examined tasks. Naturally, the more difficult the scenario was, the longer the average completion times were observed. As far as the other examined factors are concerned the situation occurred to be very interesting. Although the layout and color version factors were statistically irrelevant, their interaction significantly differentiated the mean task completion times. A closer analysis of the interaction reveals the better operation times for the tile-based layouts than their bar-like counterparts both for the red and a blue color versions. Surprisingly, for the grey color version the bars occurred to be better than tiles. This finding indicates that color versions of the interfaces should not be analyzed separately as the efficiency may be moderated by other factors. This effect is in compliance with some previous studies showing interactions between color usage and the experimental context (see e.g. Taft 1997; Schloss et al. 2012).

The obtained results regarding the applied layouts are generally in concordance with some previous findings reported, for instance, by Michalski et al. (2013) where the grid layout of products in an electronic shop were faster searched for than their list-type presentations. The reason for that is probably connected with the way people process visual information. The field of view resembles horizontally oriented ellipse and in our culture people read from the left hand side to the right hand side thus two tiles next to each other are better processed than two bars one under the other (compare the work of Michalski et al. 2006). This phenomenon, however, is completely reversed when grey color versions are concerned. This outcome is very hard to explain and requires further studies. It is possible that some preattentive mechanisms play an important role in the process of completing experimental tasks (Michalski and Grobelny 2008). Perhaps analyzing times between performing every action during the examination could lead to finding reasons of this surprising result.

The findings presented in this research should be treated as preliminary as they exhibit a number of limitations. Some of them are certainly concerned with the relatively small number of subjects involved in the examination. Analyzing the results one should also take into account that we have included only 3 out of many possible scenarios of using such software. Naturally, many other color versions could be examined and some other graphical factors included in the experiments. The efficiency results may also be supplemented by some subjective evaluations that would present more comprehensive view on the researched subject (compare Michalski 2011; 2014).

Despite these limitations the obtained results may be of interest to the software designers and marketing managers. The findings might also constitute a starting point for further scientific investigations in this field.

ACKNOWLEDGEMENTS

The work was partially financially supported by the Polish National Science Center grant no. 2011/03/B/HS4/03925.

REFERENCES

[1] CHEN Q., MALRIC F., ZHANG Y., ABID M., CORDEIRO A., PETRIU E.M., GEORGANAS N.D., *Interacting with digital signage using hand gestures*, Proceedings of the 6th International Conference on Image Analysis and Recognition, Halifax, Nova Scotia, Canada, July 6–8, 2009, Springer-Verlag, pp. 347–358.

[2] CHEVERST K., DIX A., FITTON D., KRAY C., ROUNCEFIELD M., SAS C., SASLIS-LAGOUDAKIS G., SHERIDAN, *Exploring bluetooth based mobile phone interaction with the hermes photo display*, Proceedings of the 7th International Conference on Human–Computer Interaction with Mobile Devices and Services, September 19–22, 2005, Salzburg, Austria, ACM, pp. 47–54.

[3] GROBELNY J., MICHALSKI R., BŁAŻEJEWSKA J., *The role of the package design factors in a digital signage based marketing*, [in:] Information Systems Architecture and Technology: *Network Architecture and Applications*, A. Grzech et al. (eds.), Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2013, pp. 183–193.

[4] GROBELNY J., MICHALSKI R., *Various approaches to a human preference analysis in a digital signage display design*, Human Factors and Ergonomics in Manufacturing and Service Industries, 2011, 21(6), 529–542, DOI: 10.1002/hfm.20295.

[5] ISO 9241 1998, *Ergonomic requirements for office work with visual display terminals (VDTs)*, Part 11. *Guidance on usability*, International Standard.

[6] MICHALSKI R., *Examining users' preferences towards vertical graphical toolbars in simple search and point tasks*, Computers in Human Behavior, 2011, 27(6), 2308–2321. DOI: 10.1016/j.chb.2011.07.010.

[7] MICHALSKI R., *The influence of color grouping on users' visual search behavior and preferences*, Displays, 2014, 35(4), 176–195, DOI: 10.1016/j.displa.2014.05.007.

[8] MICHALSKI R., GROBELNY J., KRAKOWIAK M., *The influence of e-shop graphical properties on the product information search*, [in:] Information Systems Architecture and Technology: *Network architecture and applications*, A. Grzech et al. (eds.), Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2013, pp. 173–182.

[9] MICHALSKI R., GROBELNY, J., KARWOWSKI W., *The effects of graphical interface design characteristics on human–computer interaction task efficiency*, International Journal of Industrial Ergonomics, 2006, 36(11), 959–977, DOI: 10.1016/j.ergon.2006.06.013.

[10] MICHALSKI R., GROBELNY J., *The role of colour preattentive processing in human–computer interaction task efficiency: A preliminary study*, International Journal of Industrial Ergonomics, 2008, 38(3–4), 321–332, DOI: 10.1016/j.ergon.2007.11.002.

[11] SCHLOSS K.B., STRAUSS E.D., PALMER S.E., *Object color preferences*, Color Research and Application, 2012, DOI: 10.1002/col.21756.

[12] TAFT C., *Color meaning and context: Comparisons of semantic ratings of colors on samples and objects*, Color Research and Application, 1997, 22(1), 40–50. DOI: 10.1002/(SICI)1520-6378(199702)22:1<40::AID-COL7>3.0.CO;2-4.

automatic recommendation, semantic similarity,
case study, user profile

Mariusz MULKA, Grzegorz POPEK*

HYBRID METHOD FOR MOVIES' RECOMMENDATION

Automatic recommendation successfully makes its way into e-commerce systems. While existing solutions use mostly purely statistical metrics, it is – in particular – interesting to build recommendation mechanisms based on users' preferences and semantic similarity of recommended objects. This paper describes a case study – an automatic recommendation of movies. We propose a hybrid filtering method formed as a combination of collaborative filtering and content based filtering. Information about users is stored in user profiles, while semantic similarity of movies is calculated using coincidence matrices of actors and genres. The proposed method is compared with collaborative filtering (which uses Pearson similarity) proposed in the literature, and with an "average" outcome (average rate assigned by a particular user). Quality of the solution is evaluated using a standard root mean square error metric. Studies presented in the paper focused mostly on an impact on the quality of recommendation imposed by a size of user profile and user preferences' dynamics.

1. INTRODUCTION

Overwhelming amounts of information have caused a content-filtering to become one of key aspects of knowledge processing. In particular, one can enhance user satisfaction by recommending them individually the most befitting items based on explicitly and implicitly gathered information.

Traditional recommendation systems [1–6] use two fundamental approaches to the filtering: *content-based filtering* (CBF) and *collaborative filtering* (CF). And both of these approaches can include semantic-based elements. Semantic methods can be used in order to evaluate a similarity of recommendation items in CBF and additionally to evaluate similarity of users' preferences in CF.

This paper describes a hybrid filtering method based on a crossover of CBF and CF (based on Pearson's similarity) applied – as a case study – in a scenario of movies' recommendation. A Pearson-based solution is chosen for CF as it has been success-

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

fully applied in the very same scenario of movies' recommendation [7]. The paper aims to show that the crossover of CBF and CF can improve the overall quality of the recommendation as it helps to deal with problems correlated with both respective approaches.

A user of the system is described using user profile [3, 5]. We focus both on the structure of the profile, on its content, and on a general dynamics (initialization and actualization) of the profile. However, the main scientific element of the described research lies in a development of a semantic method for an evaluation of similarity of movies.

Proposed semantic method for an evaluation of movies' similarity is based both on Pearson's similarity and on information stored in the profiles. A quality of the numerical recommendation is evaluated on a movie-recommendation test-set [8] using standard metric: root mean square error (RMS).

A successful implementation of a recommendation system needs to deal with a lot of problems [1] such as: cold-start, sparsity, over-specialization, and domain-dependency. Additional decisions need to be made in order to choose a proper internal representation of data related to users' profiles. Typical choices include, but are not limited to: properties' vectors for handling profiles themselves and ontologies (or at least taxonomies) for handling semantic aspects of data.

2. USER PROFILE

2.1. DATASET

The entire dataset, which was obtained from GroupLens [8], comprises 2113 users, 10197 movies and 855598 ratings given to movies by users. It contains also 20 distinct genres and 95321 actors which can be related to movie. The minimum number of ratings per user is 20, and it is a requirement of the MovieLens system. Such dataset was chosen because of concluding all the necessary data used in later considerations.

2.2. DATA REPRESENTATION

Dataset was imported to the user profile which includes the basic necessary information useful for providing recommendations. It was divided into training set and test set in order to perform a standard cross-validation and – in consequence – to evaluate results.

Let M be a set of all movies m, G is a set of all genres, and A is a set of all actors, and let U is a set of all users having a defined user profile. Then movie m is represented as follows $m = [G_m \quad A_m \quad T_m]$, where $G_m \subseteq G$, $A_m \subseteq A$ and T_m is movie title.

Let the R_u represents a collection of all movies which were rated by user *u*, then the user profile can be represented as a vector $P_u = \{m \in R_u : (m, r_{u,m})\}$, where *m* is a movie, $r_{u,m}$ is the rating which was given to the movie *m* by user *u*.

2.3. COINCIDENCE MATRIX

Coincidence matrix describes how often the individual elements occur together. Each entry can be transformed in different ways [2]. Number of common occurrences of genres *i* and *j* is referred as to $g_{i,j}$. Then the coincidence matrix for the genres is defined as follows:

$$g^*_{i,j} = \frac{g_{i,j}}{\max_i g_{i,j}} \tag{1}$$

The $\max_i g_{i,j}$ denotes the maximum value which occurs in the *i*-th row. To calculate the similarity in terms of genres, all available genres of movies are taken into account. Values of the coefficient are between 0 and 1, and when genres are identical, it is equal 1. The coincidence matrix of actors is calculated in an analogous manner, but only 20 main actors were taken into account, and is denoted as $a^*_{i,j}$.

2.4. MOVIES SIMILARITY

After defining the coincidence matrices of genres and actors, it is possible to calculate the degree of similarity between two movies *m* and *n* in terms of those two factors. It will be calculated as follows:

$$p^{(g)}_{m,n} = \begin{cases} \dfrac{\sum_{g_m \in G_m} \max_n g^*_{m,n}}{|G_m|} & \text{if } |G_m| \geq |G_n| \\ \dfrac{\sum_{g_n \in G_n} \max_m g^*_{m,n}}{|G_n|} & \text{otherwise} \end{cases} \tag{2}$$

Let $|G_n|$, $|G_m|$ denote number of genres movies *n* and *m* respectively, and let G_n is a set of all genres of movie *n*. The maximum value of the coefficient $p^{(g)}_{m,n}$ is 1 when both movies contain the same set of genres. Restriction $|G_m| \geq |G_n|$ aims to reduce the similarity between a movie with one genre to movie with many genres. The assumption $p^{(g)}_{m,n} = p^{(g)}_{n,m}$ might be not fulfilled if order of movies was not taken into account.

In turn, the degree of similarity between the two movies in terms of the actors is calculated as follows:

$$p_{m,n}^{(a)} = \begin{cases} \dfrac{\sum_{a_m \in A_m} \max_n a_{m,n}^*}{|A_m|} & \text{if } |A_m| \geq |A_n| \\ \dfrac{\sum_{a_n \in A_n} \max_m a_{m,n}^*}{|A_n|} & \text{otherwise} \end{cases} \tag{3}$$

Let $|A_n|$, $|A_m|$ are the number of all actors of the movie n and m respectively. The sets A_n, A_m consist of only the first ten actors playing in the movie. After defining how the coincidence matrices are calculated, the movies similarity function can be defined. The weighted average value of $p_{m,n}^{(g)}$ and $p_{m,n}^{(a)}$ is calculated in equation (4).

$$p_{m,n} = \frac{\left(2p_{m,n}^{(g)} + p_{m,n}^{(a)}\right)}{3} \tag{4}$$

As a result is the value of similarity between movies m and n in the range [0, 1], and it can also be seen that:

$$\forall_{m \in M} \forall_{n \in M} p_{m,n} = p_{n,m} \tag{5}$$

$$\forall_{m \in M} p_{m,m} = 1 \tag{6}$$

The first postulate (5) informs that the movie m is as similar to the movie n, as the movie n is similar to movie m. While the second postulate (6) informs that the movie m is most similar to itself, and the similarity is equal 1. Fulfilling the both postulates helps to obtain less random results and makes it easier to define filtering method.

Coincidence matrix allows finding similar movies in an easy way. The process of calculating the matrix can be done once and be stored. The matrices should be updated only when there is a new movie or a movie was edited. Analogously similarity matrix between the movies can also be calculated offline.

2.5. USERS SIMILARITY

The similarity between users is calculated using the Pearson similarity[6]. The algorithm for determining the similarity is in the form such as it was defined at the page [7].

$$r(x,y)=\frac{\sum_{m\in movie}\left(rate_{x,m}-\overline{rate_x}\right)\left(rate_{y,m}-\overline{rate_y}\right)}{\sqrt{\sum_{m\in movie}\left(rate_{x,m}-\overline{rate_x}\right)^2\sum_{m\in movie}\left(rate_{y,m}-\overline{rate_y}\right)^2}} \tag{7}$$

Let *x*, *y* are users for which are selected such movies that have been rated by both users. Let $\overline{rate_x}$ and $\overline{rate_y}$ are the average rating of user *x* and *y*; and let $\overline{rate_{x,m}}$ is the rating given to the movie *m* by user *x*. Once the similarity between users is defined, the weighted similarity between them can be calculated (8). Let $n_{x,y}$ is the number of movies that were rated by both users.

$$r^*(x,y)=\begin{cases} r(x,y) & \text{if } n_{x,y}\geq 50 \\ \dfrac{r(x,y)}{n_{x,y}} & \text{if } 0<n_{x,y}<50 \\ -1 & \text{otherwise} \end{cases} \tag{8}$$

The degree of similarity between users is a value in range [–1, 1]. However, calculating the similarity in this way has a drawback, namely, if the all ratings of user *x* are 1 and the all ratings of user *y* are 5 and they rated at least one the same movie, $r(x,y)$ would be 1. The weighted rating should eliminate similar users y, which have rated only several movies from R_x.

3. EXPERIMENTS AND ANALYSIS

The proposed hybrid filtering method is a combination of collaborative filtering and content based filtering using taxonomy called feature combination. It will be compared with two methods, the first is average rate assigned by a particular user and the second is an implementation of algorithm proposed by Netflix in 2006 [7]. This algorithm has been chosen because of its good performance at a relatively low computational complexity. As a metric was chosen root mean square error.

3.1. PROPOSED APPROACH

The general principle of the proposed method is presented in Figure 1. It describes a way in which the method works and shows a simultaneous possibility for an evaluation of obtained results.

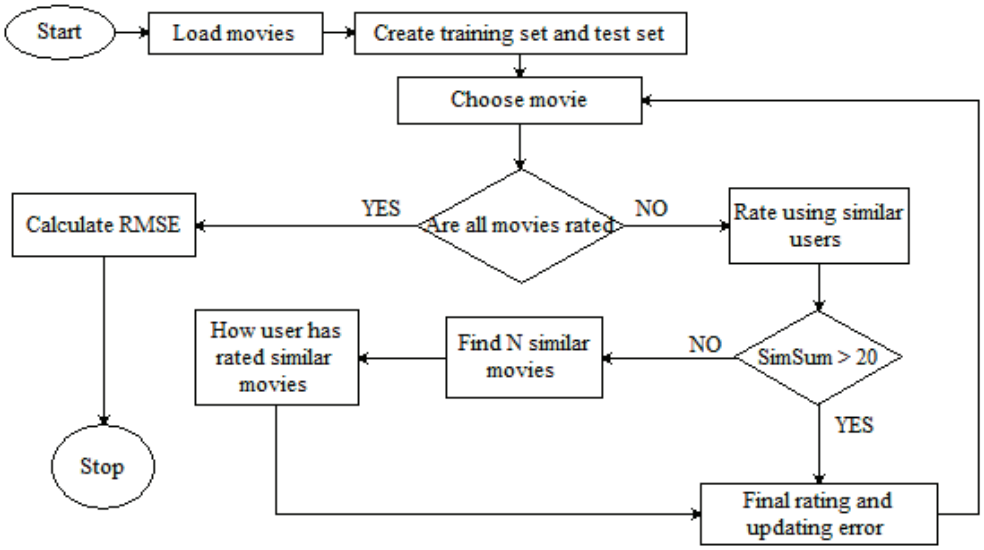

Fig. 1. Block diagram of the proposed method

At the beginning, it is necessary to load data from the database, which are necessary in recommendation process. All movies are imported with their titles and with ratings of users. The dataset is divided into a training set and a test set. All data from training set were imported to user profiles, from which the coincidence matrices, movies and users similarity matrices were calculated using equations described in chapter 3. The next step was selecting a movie from the test set for which rating should be predicted by proposed method. Further step is analogous to the method proposed by Netflix [7]. Afterwards, it is calculated how the most similar users have rated the movie *m*. But the maximum number of similar users *y* is 30 and they are contained in set Y. The first step of calculating the predicted rating is as follows:

$$rate_{x,m} = \sum_{y \in Y} \left(rate_{y,m} - \overline{rate_y} \right) r^*(x,y) \tag{9}$$

In addition, parameter $simSum_{x,m}$ is calculated, which describes how the user *x* was similar to all other users *y*, who have rated the movie *m*. The coefficient is a value in the range [–30, 30]. It is calculated as follows:

$$simSum_{x,m} = \sum_{y \in Y} r^*(x,y) \tag{10}$$

The next step is to calculate predicted rating using content based filtering. Content-based filtering will be used, if the restriction $simSum_{x,m} < 20$ is fulfilled. The value 20 was empirically obtained. If a lot of similar users rate the movie, it will not be necessary to add unreasonable computations that may undermine the results. Otherwise, additional filtering is used. Let N is a set of the most similar movies to the movie *m*, in which all movies have been rated by user *x,* and l*et* $\overline{N}$ is its cardinality. The maximum number of similar movies was empirically set to 10. The equation (11) calculates predicted rating using similar movies.

$$rate^{*}_{x,m} = rate_{x,m} + \sum_{n \in N} \left(f\left(rate_{x,n}, \overline{rate_m}, \overline{rate_n}\right) - \overline{rate_x} \right) * p_{m,n} \tag{11}$$

where $rate_{x,m}$ is the rating calculated using equation (9), *n* is a similar movie to the movie *m*, $\overline{rate_x}$ is the average rating of user *x*, and $p_{m,n}$ denotes degree of similarity between movies. The function (12) predicts rating of the movie *m* given by user *x.*

$$f\left(rate_{x,n}, \overline{rate_m}, \overline{rate_n}\right) = \begin{cases} \dfrac{rate_{x,n} * \overline{rate_m}}{\overline{rate_n}} & \text{if } \overline{rate_m} \neq 0 \wedge \overline{rate_n} \neq 0 \\ \overline{rate_{x,n}} & \text{other} \end{cases} \tag{12}$$

Let $\overline{rate_m}$ and $\overline{rate_n}$ are the average ratings of the movies *m* and *n* respectively. Assume that the average ratings for both movies were defined. If a result of the operation exceeds the value of the evaluation range [0.5, 5.0], it will be set to 0.5 or 5.0 depending on situation. The new value of the total similarity $simSum^{*}_{x,m}$ is calculated according to equation (13), and it uses the previously calculated value (10).

$$simSum^{*}_{x,m} = simSum_{x,m} + \sum_{n \in N} p_{m,n} \tag{13}$$

Let $p_{m,n}$ is the degree of similarity and N is a set of similar movies. Final rating given to the movie is calculated in the same manner as in the case of using the collaborative filtering and it is as follows:

$$rate^{*}_{x,m} = \begin{cases} \overline{rate_x} + \dfrac{rate^{*}_{x,m}}{simSum^{*}_{x,m}} & \text{if } simSum^{*}_{x,m} \neq 0 \\ \overline{rate_x} & \text{otherwise} \end{cases} \tag{14}$$

where $\overline{rate_x}$ is the average user rating. If the total value of $simSum^*_{x,m}$ is zero, estimated rating will be equal average user rating. In addition, the rating was transformed according to the function (15).

$$g(rate^*_{x,m}) = \frac{1}{10}\left\lceil 10 rate^*_{x,m} \right\rceil \tag{15}$$

The function $g(rate^*_{x,m})$ has been added in order to enforce that the values assigned by the functions belong to the same set of ratings that users could give, that is R = {0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}. Such assumption was added in order to not overestimate the quality of proposed method.

After all the steps are completed, RMS error is updated. Finally, one has to check if there are any unrated movies remaining. In consequence, the process should be repeated if the test set still contains any unrated movies. Otherwise the root mean square error can be finally calculated.

3.2. EXPERIMENTS AND RESULTS

To determine the accuracy of the selected metric, four test cases were prepared: last1p (1% of the last movies were in test set), last5 (last 5 movies were in test set), first1p (1% of the first movies were in test set), first5 (first 5 movies were in test set). At each test scenario user ratings were sorted ascending order by date. Three methods were compared: user average rate, collaborative filtering which uses Pearson similarity, proposed method with semantics.

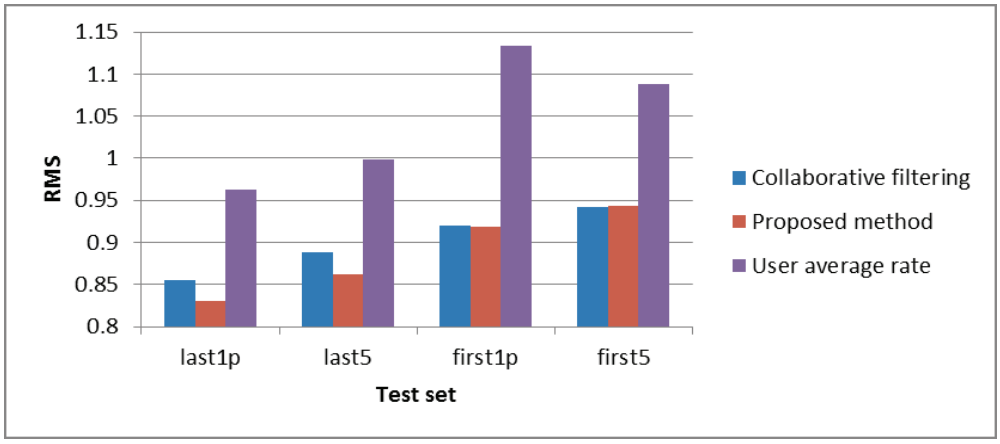

Fig. 2. Value of RMS error for different test sets

It can be seen that evaluating the last movies by the method with and without semantics performed better than the collaborative filtering. Furthermore, the results were slightly better when the semantics was not used. On the other hand, when the initial movies were evaluated, adding semantics did not work well. However, in general the results were more similar to the collaborative filtering. Getting worse results, when the ratings of initial movies were predicted, may be due to many reasons for instance the changing tastes of users, unserious treatment of the initial survey or rating the movie that was watched a long time ago. In each test scenario giving the average user rate resulted in getting the worst results.

4. CONCLUSIONS

In this paper, we have demonstrated a hybrid filtering method, which was a crossover of collaborative filtering and content based filtering. Degree of similarity between movies was calculated using coincidence matrices of genres and actors. As a quality metric of automatic recommendation was chosen root mean square error. The results obtained by proposed method were mostly better than the other tested methods. Next steps include testing proposed method for different sizes of the user profiles.

In general, the proposed approach gives the most benefits when the user profile is small as it helps to partially deal with a cold start problem. In particular, finding similar users based on initial ratings helps with obtaining a better outcome. Results (conveyed in the paper in a very limited form due to available space) confirm that increasing a size of the profile improves recommendations and show that better results are obtained if the profile consists mostly of recent movies.

ACKNOWLEDGEMENTS

This study was financially supported by Wrocław University of Technology, Institute of Informatics (*Badania Statutowe – zlecenie wewnętrzne Nr S40111 I-32*).

REFERENCES

[1] CODINA V., CECCARONI L., *Taking Advantage of Semantics in Recommendation Systems*, [in:] CCIA, R. Alquézar, A. Moreno, J. Aguilar-Martin (eds.), IOS Press, 2010, 163–172.

[2] JELEŃ K., KALINOWSKI P., MIANOWSKA B., *Rekomendacja w elektronicznym systemie rekomendacji zdjęć*, [in:] R.P. Katarzyniak, W. Rekuć (eds.), *Informatyczne narzędzia zarządzania wiedzą*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2013, 283–301.

[3] KOBSA A., KOENEMANN J., POHL W., *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*, The Knowledge Engineering Review, 2007, Vol. 15, No. 2, 111–155.

[4] MIDDLETON S., SHADBOLT N.R., DE ROURE D.C., *Ontological User Profiling in Recommender system*, ACM Trans. Inf. Syst., 2004, Vol. 22, No. 1, 54–88.
[5] SOBECKI J., *Rekomendacja interfejsu użytkownika w adaptacyjnych webowych systemach informacyjnych*, Oficyna Wydawnicza PWr., Wrocław 2009.
[6] WANG C., LIN D.Y., *A Movie Recommendation Mechanism Based on User Ratings*, [in:] HCI, C. Wang, D.Y. Lin (eds.), Springer, 2011, 610–614.
[7] www.devlicio.us/blogs/billy_mccafferty/archive/2006/11/07/netflix-memoirs-using-the-pearson-correlation-coefficient.aspx
[8] www.grouplens.org/datasets/hetrec-2011

multiagent systems, knowledge integration,
clustering, fuzzy-linguistic approach,
modal statements

Łukasz MODLIŃSKI, Grzegorz POPEK*

REPRESENTING RESULT OF KNOWLEDGE INTEGRATION WITH MODAL LINGUISTIC STATEMENTS

Generation of meaningful summaries of data is almost always about tradeoffs between precision, length of summaries, and a transparency of the system. While pursuing a goal of compact yet meaningful summaries, this paper focuses on linguistic presentation of a result of a knowledge integration process. The compactness is achieved by describing an additional dimension of the result – its degree of domination – using modal operators. The research follows a two-stage approach to knowledge integration and focuses on a translation of an output of the integration process into linguistic form. In order to present a setting, a multi-agent system equipped with a language based on fuzzy-linguistic approach is described. In particular, the paper presents – and attempts to successfully solve – a problem of modality propagation between different stages (layers) of abstraction.

1. INTRODUCTION

Multi-agent systems – the pinnacle of decentralized approaches – do not cease to stay in the spotlight as even the smallest of modern devices exhibit processing capabilities unmatched by the computers of past decades. Solving distributed problems often comes with a need for distributed processing and, in consequence, in exactly such areas multi-agent systems (MAS) win greatly over other approaches.

A particular application of MAS can be seen in an area of knowledge integration. Information gathered and pre-processed by autonomous agents may result in enormous and inconsistent body of distributed knowledge. Abstracting from such knowledge in order to obtain a meaningful summary is not an easy task not only due to potential inconsistencies but also due to usual technical limitations which usually make it impossible (or at least costly) to send "all" the knowledge to a hypothetical central processing unit. A seemingly easy solution is to create local summaries and further

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

aggregate them at higher levels. Such a solution needs establishing some means of proper communication between agents [3, 6].

Incorporating linguistic capabilities into MAS is particularly important for the task of knowledge integration as the "raw" result of the integration process may need some domain knowledge for proper interpretation [6] (e.g. information that there are 54 cars parked in front of a hotel tells a user nothing unless she knows that there is a total of 60 spaces available or is given an information that most parking spaces are taken). Such a problem can be circumvented by using a fuzzy-linguistic (FL) approach [4, 6, 7, 9] which is claimed to be able to provide statements naturally understood by external users of the system by providing transparency at the cost of some harmless imprecision [6].

It is also important [6] to convey the information about incompleteness and inconsistency of the initial knowledge to an external user of the system. Standard approaches would incorporate listing of all fitting summaries and further assigning them with probabilities or some sort of degrees of belief. The problem has been already analysed for a task of integration of fuzzy structures and an approach for coping with inconsistencies emerging during the integration process has been proposed [7]. However, for a more general case of dealing with incompleteness and inconsistency, one can follow a promising approach – a theory for grounding of language statements with auto-epistemic operators of possibility, belief, and knowledge [2, 6].

In consequence, this paper deals with MAS equipped with basic linguistic capabilities based on a combination of FL approach and the theory of grounding. The paper focuses on an aggregation of local summaries (linguistic statements generated by particular agents) into a meaningful (and proper) textual summary of distributed knowledge to an external user of the system. The research builds on a broader problem of knowledge integration and – in particular – on a multi-stage approach to knowledge integration [3, 4].

In an original approach [3] modal operators are used to express an interplay between "strengths" of particular clusters of consistent chunks of knowledge. However, due to a need of a translation of the result using FL model, modal operators are also used on a higher level of the process. In consequence, operators have to propagate between two layers of the process and various interpretational problems need to be analyzed and eliminated.

Paragraph 2 defines a basic MAS is order to present the initial requirements for the system. Paragraph 3 introduces the data integration process and its definitions. In paragraph 4 the result of research is presented regarding the representation of the result in linguistic form. Paragraph 5 presents a basic approach to define and apply modal operators on the result. Paragraph 6 presents a proposed approach to a propagation of modal operators. Finally, a general summary of the paper is provided.

2. SYSTEM DEFINITION

The main process in multi-agent system oriented toward a provision of integrated summaries is gathering data – that is – observing of a particular environment (consisting of just a single object in the case of this paper). System internals are defined based on fuzzy models which base on the definition of linguistic variable widely described in [9] and adjusted to the similar multi-agent system in [4], [6] and [3]. Selection of such data model lies in the very nature of natural language and underlying cognitive processes located within autonomous entities. More information on this topic can be found in [4] and [6].

System's agents are divided into two groups. There is a large group of agents responsible for gathering observation on given object and a single agent (called the main agent) responsible for conducting the integration process. Agents responsible for observing communicate with main agent sending their observations on the selected features of the given object.

Proceeding to a system's internal structure, let us recall (after Zadeh [9]) a quintuple

$$(\chi, T(\chi), U, G, M) \tag{1}$$

defining the linguistic variable in which χ denotes the name of the variable; $T(\chi)$ is the term-set of χ meaning it is the set of linguistic values of the variable; U is the universe of discourse of the variable χ; G denotes syntactic rule for generating names $\overline{\chi}$ of values χ and M is a semantic rule associating each $\overline{\chi}$ with its meaning.

Let the observed object in system be defined as:

$$O = \{\varphi_1\} \tag{2}$$

where φ_1 denotes a single feature the object can exhibit. The fact is in complicated systems objects have multiple features but for the sake of simplification considered object have only one feature. Presented single feature of the object is a linguistic variable with its structure as defined in (1). Feature φ_1 is related to its two properties:

$$c_{\overline{\chi}} : U \to [0, 1] \tag{3}$$

$$\lambda_{\overline{\chi}} \in [0, 1] \tag{4}$$

where (3) denotes a compatibility function required to translate selected linguistic values to internal system representation and – in further processes – to translate the

data integration result back to its linguistic form. The definition (4) presents an activation value for the linguistic value $\overline{\chi}$ which is required to translate the observations provided by the observing agents to internal representation. Activation of the linguistic value takes place the moment the equation:

$$c_{\overline{\chi}}(x) \leq \lambda_{\overline{\chi}} \tag{5}$$

is fulfilled where x denotes the observed value of feature $\overline{\chi}$ over the universe of discourse $x \in U$. Observations come in a form of one dimensional vector:

$$v_n = (\psi_{n,1}) \tag{6}$$

where $\psi_{n,1} \in T(\chi)$ denotes linguistic value for feature φ_1 of the object O observed by agent A_n. Transformation to internal representation is conducted using compatibility function and is based on the main agent's internal activation values for each linguistic value. Result of this process is in a form of vector with translated linguistic value to bracket:

$$\overline{v_n} = (\overline{\psi_{n,1}}) = [b_{*n,1}; b^*_{n,1}] \tag{7}$$

where $b_{*n,1}$ is lower bound and $b^*_{n,1}$ is the upper bound of bracket.

3. DATA INTEGRATION

Main goal of the presented multi-agent system is to integrate gathered data and present the result in a form of linguistic summary. Integration process was widely defined in [4] and [7]. Main sub-processes of the integration process are: translation to an internal representation, clustering using hierarchical clustering methods widely described in [1], and finding consensus based on an achieved result. These simple steps are valid whenever the integration result is consistent. Taking into account the cognitive and autonomous nature of agents observing the given object, the integration process result is often inconsistent. By inconsistency it is understood that the result contains at least two clusters presenting different (incompatible) groups of observations on given subject. Throughout the published papers many techniques for copying with inconsistent data were presented. Referring to the human nature few methods may be easily mentioned. Easiest to grasp is the method that relies on rejecting inconsistent observations. It is also possible to keep designated clusters and conduct the process of finding consensus on all of them.

3.1. DISTANCE FUNCTION

To conduct the process of data integration a few tools are required. One of these tools is a definition of a distance function between observations described using the internal representation. Many methods depend on a valid distance function making the definition really important. There are many existing distance functions tailored for particular cases: [5] presents three different distance functions while [4] and [7] use a distance function defined as follows:

$$d^2(x,y) = |x^* - y^*| + |x_* - y_*| \tag{8}$$

where the input x and y are two intervals for which the distance is computed.

3.2. CONSENSUS

Integration process result – regardless of its nature – requires the definition of consensus and algorithm of its finding for a given set of observations. Consensus definitions are widely explained in [5] and its definition is:

$$C_1 = [C_{*,1}; C_1^*] \tag{9}$$

adapted to the presented system definition in Paragraph 2. C_1 is a consensus for feature φ_1 of observed object O. Algorithm for consensus computing was presented in [1] and [7] while its author defined it in [5].

3.3. CLUSTERING METHODS

To achieve the result of data knowledge integration process a clustering method requires to be defined to prepare observations for the further processes. A dendrogram containing all possible clusterings is the result of this sub-process. Different approaches to clustering process have been presented in papers as they emphasize different properties of observations. In [4] and in [7] a hierarchical, agglomerative, bottom-up approach is implemented on a multi-agent system using similar data model. Hierarchical clustering algorithm was widely described and presented in [1].

Hierarchical clustering is based mainly on merging two closest clusters at time unless single cluster has been emerged. In [4] this algorithm's definition is defined as follows:

$$K_1^N, K_2^N, ..., K_N^N \tag{10}$$

where N denotes number of clusters in given clustering. Algorithm follows further merging two closest clusters measuring the distance between the consensuses of these two clusters:

$$K_1^{N-1}, K_2^{N-1}, ..., K_{N-1}^{N-1} \tag{11}$$

until all clusters are merged into single one containing all the observations gathered in the initial processes of observations:

$$K_1^1 \tag{12}$$

Such structure is crucial to perform further processes of estimating number of clusters.

3.4. ESTIMATING NUMBER OF CLUSTERS

As it was mentioned in previous paragraphs, data integration process may lead to inconsistent result. A few standard approaches for coping with inconsistencies were mentioned while in [7] two of them were compared and one of them – Gap Statistics – was found out as more accurate. Gap Statistics was presented by the authors of [8] and their multi-agent system implementation was described in [7]. In [4] it was used to conduct the similar process of data knowledge integration. Gap Statistics is a method for cluster analysis which pays special attention to the shape of error plot on within-cluster dispersion and number of employed clusters. The main idea behind this algorithm is to compare the within-cluster dispersions on two clusterings. One performed on the input data and the second one on data coming from a null model distribution based on input data taking into consideration clustering that falls far from each other.

4. TRANSLATION PROCESS

The resulting clustering from the data knowledge integration process is in a form of internal structure – regardless of its consistency. To produce a summary that is meaningful for external user it has to be translated to an understandable form such as linguistic form. Problem of translating the result back to linguistic representation is a complicated and non-trivial one taking into consideration the nature of agents acting in the presented multi-agent system and the nature of the natural language as a part of the communication process.

To achieve the summary of a desired form the translation process is required. For that purpose it should be as intuitive as human nature allows it to. Because the transla-

tion process is a key process that is conducted in accordance with human perception, the intuitive part is crucial to provide the result with desired quality. Translation process can be thought of as of simple, reverse process to initial translation to the internal structure since its result is assumed to take a linguistic form. According to previous definitions the translation process to internal structure of the system is defined on equation:

$$c_{\overline{\chi}}(x) \leq \lambda_{\overline{\chi}} \tag{13}$$

which is based on compatibility function for linguistic value $c_{\overline{\chi}}(x)$ and the activation threshold $\lambda_{\overline{\chi}}$. Let's define the translation function as a reverse of translation to internal representation equation:

$$c_{\overline{\chi}}^{-1}(\lambda) \tag{14}$$

which takes the activation threshold as its argument and a subset of linguistic variable universe of discourse as its values. By the translation process we assume a set that fulfills the given equation:

$$\overline{\lambda}_{\chi} = \{\lambda_{\chi} \in [0;1] : \neg\exists_{\lambda^* < \lambda_{\chi}} \overline{\psi_1} \subseteq c_{\overline{\chi}}^{-1}(\lambda_{\chi})\} \tag{15}$$

which is basically to acquire $\overline{\lambda}_{\chi}$ for each linguistic value given by $T(\chi)$. Linguistic value reaching the maximum $\overline{\lambda}_{\chi}$:

$$\chi = \{\lambda : \lambda = \max\{\lambda_1, \lambda_2, ..., \lambda_n\}, n = card(T(\chi))\} \tag{16}$$

Selected linguistic value under χ is the linguistic value resulting in the translation process along with its value of λ which is used in the later parts of this paper.

5. MODAL OPERATORS

Translation process of an inconsistent result does not solve completely the inconsistency problem. In fact the inconsistencies are hidden in the linguistic translations of the summary. To convey information about these inconsistencies to the user of the system, one can use modal operators widely defined in [2].

Modal operators may be adopted on layers: whole translation result and each of the clusters in the result separately. Process of adapting operators on result layer were described id [3] while the same process on single-cluster layer was presented in [6] and adopted in [4].

After [3], three modal operators are defined and used in this paper:

$POS(x)$ – "It is possible that x is the fitting opinion for this population" – understood as "x is consistent opinion of noticeable group of agents but does not form a majority in the population".

$BEL(x)$ – "I believe that x is the fitting opinion for this population" understood as "x is consistent opinion of a major group of agents but there is at least one another noticeable group with consistent opinions"

$KNOW(x)$ – "I know that x is the fitting opinion for this population" understood as "x is consistent opinion of a major group of agents and there are no other noticeable groups with consistent opinions".

5.1 RESULT LAYER

Applying modal operators on the result layer has the meaning only when the result is inconsistent, i.e. whenever two or more clusters emerged from the data in integration process. Unless at least two clusters are present, task of applying modal operators is trivial.

Process of defining, using and applying modal operators depends on their defined relative grounding strength, epistemic satisfaction relations and modality thresholds ([2]). These three properties are required to successfully apply operators on the result layer. Let's define the relative grounding strength property:

$$\lambda_n = \frac{card(K_n^Z)}{\sum_{j=1}^{Z} card(K_j^Z)} \tag{17}$$

where K_n^Z denotes certain cluster n from result clustering K^Z where Z denotes number of clusters. Defined relative grounding strength bases on number of opinions in each cluster to propose a modal operator.

Second property – modality thresholds – define which modal operator may be used at the given level of relative grounding strength. Having three modal operators defined: POS, BEL, KNOW, these thresholds are defined using the following equation:

$$0 < \lambda_{\min Pos} \le \min\{\frac{1}{\eta}, \frac{1-\lambda_{\min Bel}}{\eta-1}\} \le 0.5 \le \lambda_{\max Pos} < \lambda_{\min Bel} < \lambda_{\max Bel} \le 1 \tag{18}$$

where by η the limit of operators POS applied is defined and an example of thresholds values may be a vector:

$$(0.1, 0.6, 0.6, 0.95) \tag{19}$$

Third property required to apply modal operator are epistemic relations containing definition of use for each operator. Let's now define a three epistemic satisfaction relations used in this paper:

- Epistemic satisfaction relation $K_l^Z \equiv_G POS(\overline{C_l})$ holds when $\lambda_{\min Pos} \le \lambda_l < \lambda_{\max Pos}$;
- Epistemic satisfaction relation $K_l^Z \equiv_G BEL(\overline{C_l})$ holds when $\lambda_{\min Bel} \le \lambda_l < \lambda_{\max Bel}$;
- Epistemic satisfaction relation $K_l^Z \equiv_G KNOW(\overline{C_l})$ holds when $\lambda_{\max Bel} \le \lambda_l \le 1$.

5.2. CLUSTER LAYER

By the cluster layer it is meant the operators are applied on the translated resulted consensus for each cluster separately. Such operation is determined by the fact the main agent may select a linguistic value for given feature but his inner interpretations made his choice vague. As defined in Paragraph 5.1, applying operators on cluster layer requires three properties definition: relative grounding strength, epistemic satisfaction relations and modality thresholds. The only difference between applying operators on cluster level and on the result level is in definition of relative grounding strength. Wider definitions may be found in [4] and in [6].

Relative grounding strength for describing operators on cluster layer is the same property as defined at the end of Paragraph 4 – $\overline{\lambda}_\chi$ – which basically is the level the given feature was translated at.

6. PROPAGATION

As for two different layers of modal operator applying, there is a need for operators' propagation because on two separate layers these operators may be opposing to each other. The propagation process shall be conducted in a direction from the cluster layer to the result layer and out of two operators defined only the one with its semantic on agent's certainty is lower.

This process has its special meaning when the observed object has more than just a single feature. Propagation lets main agent inform the outer user of its uncertainty on the given object so user will not mistake the real object state and will understand the summary using his own interpretations.

7. CONCLUSIONS

In this paper, a process of knowledge integration was described and a multi-agent system setup was presented to conduct such process. Given the fuzzy structure the system may operate on natural language statements which is a step forward the human and machine communication with understanding. A main goal of this paper was to present a method for back-translation of the processed structure in the given system to linguistic form as summaries. To improve the quality of the result in linguistic form modal operators have been used. Operators are required whenever we want to report a certain level on the given result. Without the modal statements outer user may get the information the main agent is not sure of and interpret it mistakenly. Modal statements were presented on two different yet connected layers of the result: on whole result layer and on single cluster layer. Each of the layers presents different part of the exact result. On the result layer modal statements presents the relations between the whole clusters. On the cluster layer they present the certainty on selected linguistic values for object's feature in the process of back-translation. Propagation of modalities has its meaning especially whether the observed objects have more than just a single feature.

ACKNOWLEDGEMENTS

This study was financially supported by Wrocław University of Technology, Institute of Informatics (*Badania Statutowe – zlecenie wewnętrzne Nr S40111 I-32*).

REFERENCES

[1] JAIN A.K., MURTY M.N., FLYNN P.J., *Data clustering: a review*, ACM Computing Surveys (CSUR), 1999, 31(3), 264–323.
[2] KATARZYNIAK R., *The language grounding problem and its relation to the internal structure of cognitive agents*, J. UCS, 2005, 11(2), 357–374.
[3] LORKIEWICZ W., POPEK G., KATARZYNIAK R.P., *Intuitive approach to knowledge integration.* [in:] The 6th International Conference on Human System Interaction (HSI) 2013, IEEE, 2013, 40–47.
[4] MODLIŃSKI Ł., *Wybór lingwistycznej reprezentacji zintegrowanego rezultatu klasyfikacji oraz metody ustalania momentu ponownej klasteryzacji w systemie pracy kolektywnej*, Politechnika Wrocławska, Wrocław, Poland, 2014.
[5] NGUYEN N.T., *Advanced methods for inconsistent knowledge management*, Springer, 2007.
[6] POPEK G., *Integration of Modal and Fuzzy Methods for Agent's Knowledge Representation*, PhD Thesis, Swinburne University of Technology, VIC, Australia, Wrocław University of Technology, Wrocław, Poland, 2013.
[7] SKORUPA G., KATARZYNIAK R.P., MODLIŃSKI Ł., MULKA M., *Multi-agent platform for fuzzy structures integration task*, ISAT, 2013.
[8] TIBSHIRANI R., WALTHER G., HASTIE T., *Estimating the number of clusters in a data set via the gap statistic.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001, 63(2), 411–423.
[9] ZADEH L. A., *The concept of a linguistic variable and its application to approximate reasoning-III*. Information Sciences, 1975, 9(1), 43–80.

text classification
frequent sequences, n-grams

Piotr OŻDŻYŃSKI*

TEXT DOCUMENT CATEGORIZATION BASED ON WORD FREQUENT SEQUENCE MINING

In the paper a method of text document classification based on word frequent sequences is considered. In the presented technique the process of appropriate sequence selection and class assignments is automatic, but manual manipulation and modification of the acquired data is also possible if better adaptation to the test set is necessary. As the main part of the proposed methodology an effective algorithm for discovering all frequent sequences in document sets is presented. The proposed document classification method is evaluated by experiments carried out for the Reuters document collection. Obtained results were compared with those received by using known classification techniques. Result analysis showed that the proposed method is of better or of comparable efficiency, taking into account k-NN or Naive Bayes classifiers.

1. INTRODUCTION

The classification of text documents is one of the important tasks of text mining. As its main applications there should be mentioned: spam recognition, thematic list group determination or sentiment analysis. It is also used in filtering of selected kind of content (content-control software) [10].

Many methods have been developed to categorize documents. The techniques are mostly based on algorithms, which use reference training sets that contain documents classified by an expert. These sets are used for classifier to learn and then to assign relevant classes to examined documents.

A large part of the text mining algorithms has its origin in data mining. Among them frequent sequences discovery is derived from the techniques associated with the market basket analysis, where frequent itemsets in the transactions are searched. The algorithms used in this task are the GSP (Generalized Sequential Pattern) [1, 15], PrefixSpan [12].

* Lodz University of Technology, Institute of Information Technology, Wólczańska 215, 90-924 Łódź.

In the paper **F**requent **S**equences **B**ased **C**lassification (FSBC) is considered. It is based on the heuristic that some sequences of words in the text may be so significant that on the basis of their presence one can define a document class apart from other text content. This action resembles the mechanism used by the expert to classify. The human expert not need to read the entire document and he knows when he has sufficient information to be able to classify the document. Therefore, it seems legitimate to use such sequences for classification. Sequences satisfying such properties are called emerging patterns. Their use has been the subject of various studies [6, 13]. This paper presents the method of selection of emerging patterns from the frequent sequences.

The third part will discuss in details frequent sequences search algorithm. The fourth section describes the methods of searching the most representative sequences (emerging patterns). Part five presents obtained results.

2. RELATED WORK

The searching for frequent sequence in the text is a problem transferred from the scope of database exploration. Various adaptations of sequence search algorithms based on the existing solutions are proposed. Ahonen-Myka proposed the search method of frequent sequences taking an acceptable interval between the words[3].

García-Hernández et al point out the significant difference between search patterns in transaction databases and text databases. Algorithms are oriented to search for sequences in the small set of attributes from a large number of transactions. Database of text documents can have a small number of items, but with a significant amount of attributes. Hence, the algorithms generating all possible candidates and testing the presence of the pattern are not very efficient. The proposed algorithms DIMASP-C and -D (Discover all the Maximal Sequential Patterns) seek the maximum length frequent sequences respectively in the collections of documents (DIMASP-C) and in a single document (DIMASP-D) [7, 8].

There are many methods for classifying text documents. Some of them are based on the vector representation of document called *bag of words*. The document is converted to a vector whose components represent each instance of a particular word in it. The use of the VSM representation (Vector Space Model)[14] allows to use a measure of similarity based on the distances of points designated by the vector of the document. The problem with such an approach is the large size of the feature vector, which vector is also sparse. This enforces the need for a reduction of dimension. Methods that fulfill this requirement are the use of stop-list and the word stemming. Subsequent phases of vector construction are designed to select and emphasize the features that better than any other represent the membership of a document to the class. As used in this method is to ignore words that occur less

frequently than the predetermined threshold or the rating of information gain for individual [17].

Another approach is to use the characteristic for the class of document text phrases. This approach seems to be more similar to natural: an expert evaluating a document under the terms of belonging to a certain class is able to take a decision finding the phrase which is characteristic for a given group of documents. Representation of the document as a vector of maximum length sequences was used to documents clustering[9].

Tan, Wang and Lee use bigrams of words to categorize [16]. The feature vector of the document is created from single words and bigrams. Naive Bayes and the maximum entropy methods are used as classifiers. A number bigrams is limited in order to avoid significant enlarge the size of the feature vector. It has been shown that this method improves the quality of categorization.

Antonie and Zaiana [4] present an approach based on the use of association rules known from the basket analysis. Evaluation of the results is carried out on a set of Reuters-21578, which allows to compare the results.

Kotagiri and Bailey in their work [13] used emerging patterns to the classification. The emerging pattern is defined as a set of features that support varies considerably across classes. For the purpose of text classification this features may be frequent sequences. This approach will constitute the subject of this study.

3. FINDING MOST FREQUENT SEQUENCES

A sequence is an ordered list of consecutive words . The sequences A and B are equal if they have the same length and in both sequences the same words at the same positions. The length of the sequence is the number of words within.

Definition: A sequence $B = (b_1, b_2, b_3, ..., b_m)$, $m = |B|$ is contained in the sequence $A = (a_1, a_2, a_3, ..., a_k)$, $k = |A|$, (B is a subsequence of A) if there exists a number d and

$$0 \le d \le k - m \wedge \forall i \in \{1, ..., m\} : b_i = a_{i+d} \tag{1}$$

The sequence consisting of n words will be referred to the name of the n-gram. In particular, the bigram and trigram will mean the sequence of the two and three words. In this paper the word sequence will be used interchangeably with the word phrase.

Example: the sequence of words $A = (a, c, d, f, c)$ in the document is represented as the text "*a c d f c*". Thus, according to the definition, the sequences B_1 = "*a c d f*", B_2 = "*d f c*", B_3 = "*c d f*" are subsequences of A with values of d respectively equal to 0, 2, 1. Sequences C_1 = "*a c d c*", C_2 = "*d c*" do not satisfy the condition therefore are not subsequences of the sequence A.

3.1. CONSTRUCTION OF THE ALGORITHM

The algorithm is based on the principle of building the new frequent sequence of length $n + 1$ on the basis of the existing sequence of length n and information about bigrams location. This approach is derived from the observation that if a sequence of length $n + 1$ is frequent, all subsequences of the sequence are also frequent. It is used in the algorithm *a priori* [2], which is a reference for other searching frequent patterns algorithms. Thus, searching for frequent sequences of length $n + 1$ can be assumed that it consists of frequent sequences of length n. Additionally the new sequence has on the position n a bigram from the set of frequent sequence (of length 2).

```
Input: The map of bigrams, the map of n-grams
Output: the map of (n+1)-grams
for each n-gram
    for each bigram which starts with last word of an n-gram
        (compare n-grams locations)
        if bigram is consecutive to n-gram
            add (n+1)-gram to the result set
```

Algorithm 1. A frequent sequences searching algorithm

A set of analyzed documents is initially normalized. This stage is limited to the removal of punctuation marks and numbers and convert uppercase to lowercase.

In the first step of the algorithm a unique identifier is assigned to each word in the whole set. The identifier is an integer number. A collection of text documents is converted into a set of numerical sequences. Each next occurrence of the word is replaced with the identifier given to the first occurrence.

The second step of the algorithm is to build a data structure that stores all pairs of consecutive words and additional information about their position. The occurrence of each pair is associated with a specific document and position of the pairs being an offset from the beginning of the document. To simplify the notation both identifiers are stored as a single integer. Using binary notation older bytes store the document index and younger bytes store a bigram position. This structure will be stored as a map. The key will be a pair of numbers – the bigram, and the value is a sequence of integers. Because bigrams are indexed the structure is called *the inverted bigram index*. Since longer n-grams will be represented by the same data structure it is required to remember also a sequence length.

This method of storing bigrams positions is equivalent to store document as a sequence of words. Both forms can be converted without loss of information. It is also possible to combine the first and second step to produce the word's indexes and bigram indexes in one pass.

The presented algorithm applies to search frequent sequences, therefore it is necessary to determine the support threshold from which the sequence may be considered as frequent. Further steps of the algorithm will be performed for only these *n*-grams for which the number of occurrences is greater than or equal to the threshold. For this reason it is practical to sort bigram keys in order from the most frequent.

3.2. ANALYSIS OF THE DATA SIZE

After creating a structure that stores the bigram's location number of keys will depend on the data contained in the text. The number of bigrams can be equal to the square of the number of unique words in a set of documents. However, assuming that the set consists of k words and each is unique the maximum number of bigrams is less than k because each bigram has its location in the document. Thus the number of possible bigram's positions depends on the size of the documents set.

In real documents words have different frequencies. The same is true for bigrams and larger sizes of n-grams. This distribution is determined empirically by the *Zipf's law* [11] in the form:

$$f(s) = \frac{A}{s^{\alpha}} \tag{2}$$

Equation 2 that the number of words specified by the index s is inversely proportional to the index raised to the power α wherein the exponent is a number slightly larger than 1 and A is the factor of proportionality. Words are arranged in descending order of number of occurrences.

3.3. THE ITERATIVE SEQUENCE GROWTH

The starting set of frequent sequences is the set of bigrams so $n = 2$. For each sequence of length n (denoted as $S_i(n)$) a list of candidate sequences of length $n + 1$ ($S_j(n + 1)$) is created. All bigrams whose first word is the same as the last word of the starting n-gram are searched. Since such an operation will be repeated often it is reasonable to hold such a map of bigrams in memory. The key of such a map would be the word and the value would be an array of bigrams starting with this word.

The list of potential sequences longer by one is formed by adding a bigram on the end of this sequence. The new sequence has to be more frequent than the specified threshold. It is therefore necessary to count the number of times the sequence occurs in the text. There is no need to search all the set of documents. It is enough to compare an array of positions of the sequence $S_i(n)$ and the position of the bigram that expands this sequence. If a candidate sequence $S_j(n + 1)$ occurs in the text the position of the bigram at the end of this sequence has an index position greater by $\delta = n - 1$ than the index of the beginning of the sequence $S_i(n)$.

Example: let us consider the relation presented at a figure 1.

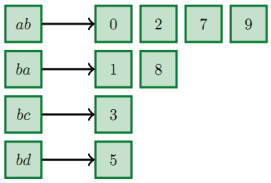

Fig 1. Relation between sequences and their positions

Sequence *ab* occurs in the set four times. There are three bigrams that start with the word *b*. It is possible that there are sequences *aba*, *abc* and *abd*. In fact, the documents contain two sequences *aba* in positions 0 and 7 because there are bigrams *ba* on positions 1 and 8. In addition there is one sequence *abc* on position 2. There is no sequence *abd*, because the bigram *bd* on position 5 does not satisfy the condition being one step after *ab* (there is no *ab* on position 4). Taking the support threshold value equal 2 there is one 3-gram *aba* satisfying this condition.

Thus the construction of sequences based on the initial subsequence can be reduced to the comparison of two ordered arrays of numbers A and B. There are searched pairs $\{a, b\}$ such that $a \in A$, $b \in B$ and $\$b = a + \delta$. Since the arrays are ordered an efficient algorithm using binary search can be designed.

After completion of the cycle of the algorithm for the next n data is stored in the structure like this prepared for bigrams. Each n-gram is defined by specifying the array of starting indices, the sequence length, the first and the last word.

The searching ends if it is not any longer frequent sequence found. Recreating of the sequence of words is possible by reading from a document or from a reverse bigram index.

4. FREQUENT SEQUENCES BASED CLASSIFICATION ALGORITHM

The search for emerging patterns in the among frequent sequence has its legitimate in the definition. It is necessary to achieve a certain level of a pattern support to be able to say that in other classes that support is much lower. On the other hand the sequences will often be present in all groups of documents carrying no information about class membership. The essence of the problem is to select these frequent sequences that strongly satisfy the condition of differentiation.

The resulting set of frequent sequence has a structure that allows to extract information on which classes of documents a phrase is applied and how many times. Phrase position in inverted index determines both the document and the

offset in the document. Thus for each sequence a list of pairs containing number of occurrences in a class and this class name can be assigned. The list is ordered descending by the number of occurrences. Due to the different number of instances of different phrases in the training set it is necessary to normalize the list. The number of instances in each class is divided by the total number of occurrences of a given sequence.

So now it will be the ratio of the number of occurrences of the phrase in the selected class to the total number of instances. This is referred to as the frequency of occurrences of the phrase in a class of documents. The sum of the frequencies for all classes will be equal to one.

Sequences which are present in a minimum number of classes and also in a certain class number of occurrences is much higher are preferred. Such a sequence is associated with class that has greatest frequency. Proposed function that determines the weight of a sequence association is given in equation 3.

$$w = \sum_{i=1}^{k} f_i^{\alpha} \tag{3}$$

The exponent α has a value greater than 1. It may be noted that if $\alpha = 2$ the score will be the sum of the squares of the frequencies. This sum will have a maximum value of 1 where there is only one class of frequency 1. If there are two classes the minimum value is 0.5 if the frequencies are equal and grows to the limit of 1 where the proportion changes.

Classification performance can be measured using *precision* (π) and *recall* (ρ). Since these values are dependent on each other a combination of both is essential. Factor F_1 is the harmonic mean of π and ρ [5].

$$F_1 = \frac{2\pi\rho}{\pi + \rho} \tag{4}$$

This value is calculated for each class of documents. The weighted average value of F_1 for each class is called *micro-avg*.

5. EXPERIMENT RESULTS

Tests were done for the Reuters-21587 ata set. The selected subset ModApte [5] was used to test the classification algorithm. It contains 3299 test documents and 9603 training documents. The set of 10 classes with the greatest number of documents has been selected. In addition in the process of learning and classification documents without content have been omitted. Obtained results were compared with the available results of other classification algorithms [10] in table 1.

Since the developed algorithm assigns only one class to test the document but used set of documents has assignment to more than one class there are three different approaches to the problem. In column FSBA DJ (disjoined) only those documents that were assigned to one class were selected for learning and testing. The results in column FSBA MT (multi topic) refer to documents that can belong to several classes. In this case, the classifier assigns only one class. This increases the number of documents classified as a false negative. Column FSBA FT (first topic) contains the results for documents that has been assigned to only one class and the others are omitted.

Table 1. Effectiveness of FSBA and other classifying algorithms (%)

	Naïve Bayes	k-NN	Decision tree	SVM	FSBA (DJ)	FSBA (MT)	FSBA (FT)
Earn	96	97	98	98	96	93	94
Acq	88	92	90	94	92	87	91
money-fx	57	78	66	75	77	69	71
grain	79	82	85	95	0	38	28
crude	80	86	85	89	84	69	77
Trade	64	77	73	76	76	66	72
interest	65	74	67	78	67	49	67
Ship	85	79	74	86	28	32	39
Wheat	70	77	93	92	–*	23	26
Corn	65	78	92	90	–*	4	13
Micro-avg	82	82	88	92	91	80	85

* – there are no documents of this class in the selected subset.

Due to the lack of information how the problem of belonging to multiple classes is resolved in the individual studies an additional comparison was performed on a subset of the set Reuters-21578. Documents containing text in the BODY tag and additionally in the tag PLACES have only one of the selected location: *canada*, *france*, *japan*, *uk*, *usa*, *west-germany* are selected. Thus formed a subset containing 13442 entries. The test group consists of 5000 randomly selected documents and the training group contains 8442 documents. Naive Bayes and k-NN method were used for the classification. Words from the stop-word list have been removed and the others were stemmed. The feature vector consisted TF * IDF coefficients calculated for each term in the document. This method of extracting features gave the best results for both methods and, therefore, was selected for comparison. Obtained results for measures of micro-avg are follows: 89% for Naïve Bayes, 87% for k-NN and 90% for FSBC. Obtained results show that the performance of the proposed method is comparable to or better than the method of Naive Bayes, k-NN and decision trees. Could not get as good of performance indicators such as using the SVM method.

6. SUMMARY AND CONCLUSIONS

The **F**requent **S**equences **B**ased **C**lassification (FSBC) that uses frequent sequences and their weights is presented. A document is classified to the class in which the most frequent is the sequence of the highest importance.

The paper proposes a method to discover frequent sequences of words in a set of documents. Sequences are formed by combining the existing ones with bigrams. The possibility of a connection determines the position of the first and last bigram in the sequence. A simple method of finding a significant weight of sequence is proposed too. This method lets to find emerging patterns in a set of sequences.

Investigated classification results were compared with the ones of other methods available in the literature and with own results. The proposed method gives results comparable to or better than the method of Naïve Bayes, k-NN and decision trees. The efficiency is worse than this in SVM method.

An important feature of the proposed method is the fact that the applying this does not require processing characteristic for the language of the document. This allows use the method regardless of the language of the. Hence there is no need for matching the correct stemmer and a dedicated stop-words list. Frequently repeated words are in frequent sequences which appear in all classes and then have a low weight therefore do not participate in the decision process.

This approach allows for easy analysis of each incorrectly classified document for opportunities to improve efficiency. Only the sequence with the highest weight is selected to classify a document but the others, with lower weights are also known. This sequences can be used to construct a k-NN like classifier to improve the method efficiency.

REFERENCES

[1] AGRAWAL R., SRIKANT R., *Mining sequential patterns*, [in:] Proceedings of the Eleventh International Conference on Data Engineering, 1995, Mar 1995, 3–14.

[2] AGRAWAL R., SRIKANT R., *Fast algorithms for mining association rules in large databases*, [in:] Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1994, 487–499.

[3] AHONEN-MYKA H., *Discovery of frequent word sequences in text*, [in:] Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery, Springer-Verlag, London, UK, UK, 2002, 180–189.

[4] ANTONIE M.-L., ZAIANE O.R., *Text document categorization by term association*, [in:] IEEE International Conference on Data Mining, 2002, ICDM 2003. Proceedings, 2002, 19–26.

[5] DEBOLE F., SEBASTIANI F., *An analysis of the relative hardness of Reuters-21578 subsets: Research articles*, J. Am. Soc. Inf. Sci. Technol., April 2005, 56 (6), 584–596.

[6] DONG G., LI J., *Efficient mining of emerging patterns: Discovering trends and differences*, [in:] Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, New York, NY, USA, 1999, ACM, 43–52.

[7] GARCÍA-HERNÁNDEZ R., TRINIDAD J., CARRASCO-OCHOA J., *A new algorithm for fast discovery of maximal sequential patterns in a document collection*, [in:] Alexander F. Gelbukh, (ed.), CICLing, Springer, Lecture Notes in Computer Science, 2006, Vol. 3878, 514–523.
[8] GARCÍA-HERNÁNDEZ R., TRINIDAD J., CARRASCO-OCHOA J., *Finding maximal sequential patterns in text document collections and single documents*, Informatica (Slovenia), 2010, 1, 93–101.
[9] HERNÁNDEZ-REYES E., TRINIDAD J., CARRASCO-OCHOA J., GARCÍA-HERNÁNDEZ R., *Document representation based on maximal frequent sequence sets*, [in:] Proceedings of the 11th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications, CIARP '06, Springer-Verlag, Berlin, Heidelberg, 2006, 854–863.
[10] MANNING C., RAGHAVAN P., SCHÜTZE H., *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2008.
[11] MONTEMURRO M., *Beyond the Zipf–Mandelbrot law in quantitative linguistics*, Physica A: Statistical Mechanics and its Applications, 2001, 300(3–4), 567–578.
[12] PEI J., HAN J., MORTAZAVI-ASL B., WANG J., PINTO H., CHEN Q., DAYAL U., HSU M-C., *Mining sequential patterns by pattern-growth: the prefixspan approach*, IEEE Transactions on Knowledge and Data Engineering, Nov 2004, 16(11), 1424–1440.
[13] RAMAMOHANARAO K., BAILEY J., *Discovery of emerging patterns and their use in classification*, [in:] Tamás (Tom) Domonkos Gedeon and Lance Chun Che Fung (eds.), AI 2003, Advances in Artificial Intelligence, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003, Vol. 2903, 1–11.
[14] SALTON G., WONG A., YANG C., *A vector space model for automatic indexing*, Commun. ACM, November 1975, 18(11), 613–620.
[15] SRIKANT R., AGRAWAL R., *Mining sequential patterns: Generalizations and performance improvements*, [in:] Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '96, Springer-Verlag, London, UK, UK, 1996, 3–17.
[16] TAN C-M., WANG Y-F., LEE C-D., *The use of bigrams to enhance text categorization*, Information Processing and Management, 2002, 38(4), 529–546.
[17] YANG Y., PEDERSEN J., *A comparative study on feature selection in text categorization*, [in:] Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1997, 412–420.

ontologies, inconsistency of ontology,
ontology matching

Daniel HALIKOWSKI*
Adriana HALIKOWSKA**

RESOLVING INCONSISTENCIES IN ONTOLOGY INTEGRATION

Ontology integration is a basic and very important subtask in many tasks of knowledge management. This article describes issues related to the themes of ontology, which will allow you to understand the causes of problem of inconsistency of knowledge with which information systems have to deal. There are also shown elements we must pay attention to during ontology matching process. Levels of inconsistency in a process of ontology matching, and a methods for resolving problems occurring in ontology integration are presented. Methods described in this work are the basis for further research on the fuzzy ontology integration in the process of building knowledge management systems.

1. INTRODUCTION

The main goal of information exchange is to enable an understanding of meaning and content of the information. The problem of organizing and structuring information in a knowledge-based society is a real challenge. Therefore there are many methods that have been developed to deliver content and significance of the expression of natural language sentences [4]. To have the possibility of usage of the stored knowledge as a basis for decision-making systems, it is necessary to create the a suitable tool that will properly process and store valuable information. However, in order to create a computer system that is able to formalize the analyzed content, it was necessary to create a method of formalizing contained information. So it seems necessary to create an intermediate element, which in the right way will analyze and formalize knowledge, regardless of

* University of Applied Sciences in Nysa, Institute of Computer Science, e-mail: daniel.halikowski@pwsz.nysa.pl
** University of Applied Sciences in Nysa, Institute of Finance, e-mail: adriana.halikowska@pwsz.nysa.pl

these factors. It would allow to present available data, information and knowledge in useful way from the point of view of knowledge-based systems. Ontologies are indicated to fulfill the role of mediator, they have been developed to provide automatic transformation semantics of information sources. The idea of ontologies is to allow users to organize information in concepts, with their own attributes and relationships between the concepts. When data are represented by the ontologies, IT systems can better understand their content. Unfortunately, during the creation of ontology we have to deal with the problem of heterogeneity of data or many different ways in which you can model the same fragment of reality. This in turn causes problems while integrating ontology, because the same fragment of real world can be defined by other ontology, in different way.

This paper presents the problem of ontology integration and the methods of solving the problem of inconsistency of knowledge during the process of ontology integration. The described methods are essential in the process of ontology matching and provide the base for subject fuzzy ontologies that are the target of my research.

2. ONTOLOGY INTEGRATION PROBLEM

The concept of the ontology was originally associated with the department of philosophy, that explores the ownership of items and their ways of the existence [2], [3]. In computer science, the most frequently quoted definition of ontology is that formulated in the publication of Tom Gruber in 1993. He described the ontology as a formal, explicit specification of a shared conceptualization ("A specification of a representational vocabulary for a shared domain of discourse – definitions of classes, relations, functions, and other objects – is called an ontology" [5], [6]).

General idea of conceptualization is on the figure 1.

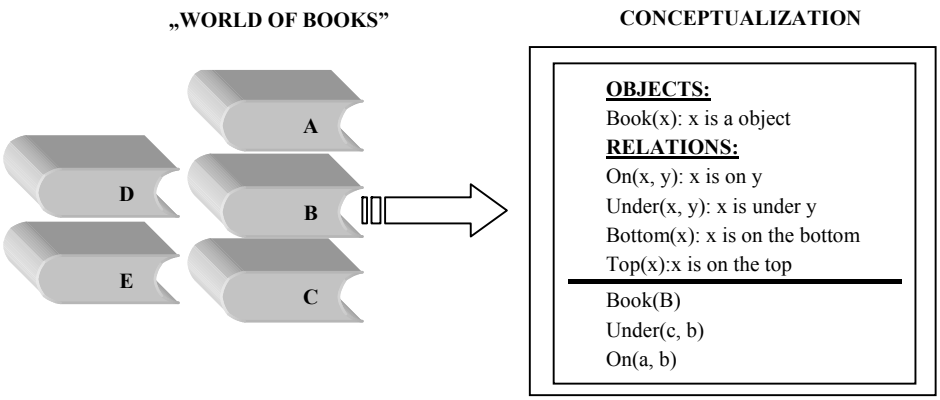

Fig. 1. Idea of the conceptualization [2]

Above definition shows that ontologies are created to the appropriate data representation. This in turn will be the basis for understanding the meaning of information and correct data processing by communicating agents (software and human). This approach creates a common conceptualization of reality. So understood conceptualization refers to an abstract model of a certain phenomenon or being, which identifies the relevant concepts of the real object. The conceptualization of the ontology, it is necessary to uniquely identify and interpret knowledge. The term "shared" in the above definition means that that the use of ontology is not confined to a single program, but can be used by a larger group of programs. Ownership of knowledge sharing has become a starting point to create ontology matching techniques and allows to use concepts and relationships defined in a variety of component ontologies. The idea of ontology is shown in fig 2.

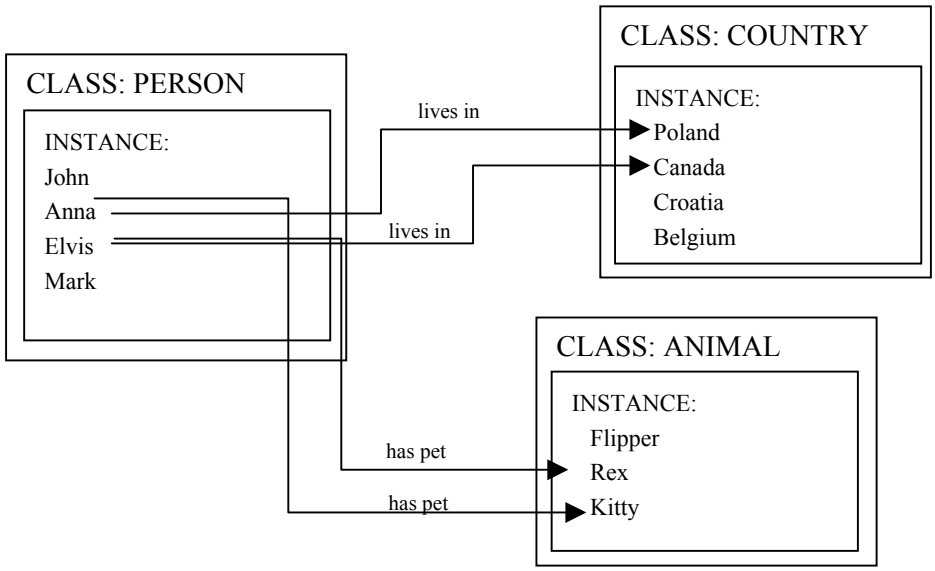

Fig. 2. Ontology as an example of objects, concepts and interrelationships – self study

Most often ontology is defined by the following elements [8]:

- **C** – Set of concepts (classes)
- **I** – Set of instances of concepts
- **R** – Set of binary relations defined on **C**
- **Z** – Set of axioms which are formulae of first-order logic and can be interpreted as integrity constraints or relationships between instances and concepts, and which cannot be expressed by the relations in set **R**, nor as relationships between relations included in **R**.

From the formal point of view, ontologies can be thought of as a tool for modeling the selected slice of reality, including its decomposition on the basics objects (classes) and definition of possible dependencies between them. Given the above, we can define ontologies according to the formula [7]:

$$O = (C, R, Z, I). \tag{1}$$

To understand how the ontologies refer to the real world requires its definition. We assume a real world ($\boldsymbol{A}$, $\boldsymbol{V}$) where $\boldsymbol{A}$ is a finite set of attributes and $\boldsymbol{V}$ is the domain of $\boldsymbol{A}$, that is $\boldsymbol{V}$ is a set of attribute values, and V_a is the domain of attribute a [9].

$$\boldsymbol{V} = \bigcup\nolimits_{a \in A} V_a \,. \tag{2}$$

Ontologies referring to the real world ($\boldsymbol{A}$, $\boldsymbol{V}$) are called ($\boldsymbol{A}$, $\boldsymbol{V}$)-based. A concept of an ($\boldsymbol{A}$, $\boldsymbol{V}$)-based ontology is defined as a triple (c, A^c, V^c) where c is the unique name of the concept, defined as $c = (Id^c, A^c, V^c)$. Id^c is a unique class identifier, $A^c \subseteq \boldsymbol{A}$ is a set of attributes describing the concept, and $V^c \subseteq \boldsymbol{V}$ is the attributes domain:

$$\boldsymbol{V}_c = \bigcup\nolimits_{a \in A^c} V_a \,. \tag{3}$$

Each ontology meeting the following criteria: $\forall_{c \in C} A^c \subseteq A$, $\forall_{c \in C} V^c \subseteq V$ will be referred to as an ontology based on the world ($\boldsymbol{A}$, $\boldsymbol{V}$). Pair (A^c, V^c) is called the *structure* of concept c. Set $\boldsymbol{C}$ in the ontology definition is a set of ontology names. An instance of a concept c is described by the attributes from set A^c with values from set I^c and is defined as a pair *instance* = (i, v) where i is the unique identifier of the instance in world ($\boldsymbol{A}$, $\boldsymbol{V}$), v is the value of the instance, a tuple of type A^c, and can be presented as a function: v: $A^c \rightarrow V^c$ such that $v(a) \in V_a$ for all $a \in A^c$. Then, i is defined as $i = (id, A_i, v_i)$ where id is the unique identifier of the instance, A_i is a set of attributes instance, v_i is a function that assigns values to attributes of an instance from the collection of domains.

$$v_i : A_i \rightarrow \bigcup\nolimits_{a \in A_i} V_a \,. \tag{4}$$

Ontology defined in this way assumes that part of reality which is to be modeled by using ontologies, can be decomposed on a finite set of component objects. This approach can creates problems associated with a variety of ways of representing the same part of reality by ontologies systems, because different knowledge bases describing the same part of reality can define it in different, sometimes conflicting with each other way. So the question is how to find the similarity between ontologies and

thus provide a way to easily integrate data between two knowledge bases using ontologies as a way to store information. In general, the issue of integration of ontology is based mainly on combining and matching ontology, however, the main ways of integrating ontologies can be divided into three groups. Ontology integration is a process which may be done on one of the following three levels [7], [10]:

- Building a new ontology reusing other available ontologies: This approach involves creating a completely new ontology by using some other ontologies which have been built.
- Merging different ontologies about the same domain into a single one that "unifies" all of them: One wants to build an ontology merging ideas, concepts, distinctions, axioms, and the like, that is, knowledge, from other existing ontologies about exactly the same domain. This approach to the ontology integration consists of modifying all the input ontologies and creating a final ontology which is the sum of the components of structures undergoing to integration.
- Introducing ontologies into applications: In this case, several ontologies are introduced into an application; they underlie and are shared among several software applications. It is also possible to use several ontologies to specify or implement a knowledge-based system on the basis of distributed resources.

3. INCONSISTENCY OF ONTOLOGIES

The idea of ontologies is to allow users to organize information on the taxonomies of concepts, with their own attributes, to describe relationships between the concepts [9]. Their integration involves determining the relationship between the elements of the integrated ontologies, relating to the same real object [12], [11]. The final system needs to have the ontology which is the result of integration of component ontologies (Fig. 3).

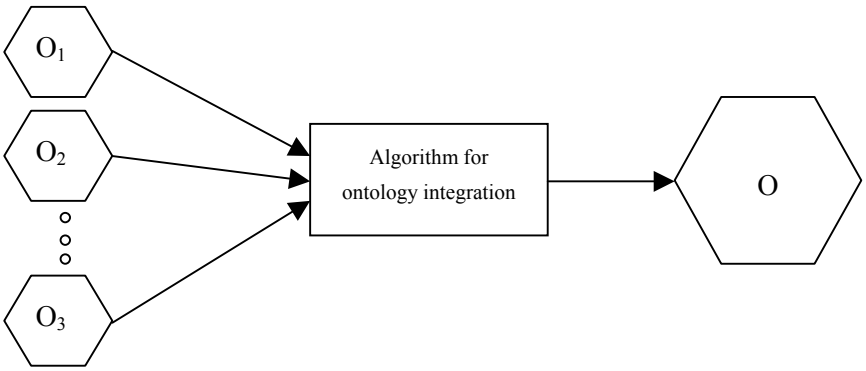

Fig. 3. Scheme for ontology integration [7], [1]

However, this task requires to deal with the problems of conflicts arising during the integration process. Conflicts can occur when the ontologies refer to the same part of the real world and its objects are differently reflected. The integration process will therefore rely on the proper classification of objects.

Inconsistency between ontologies is considered on the following levels [8]:

- Inconsistency on the concept level: The concept integration condition is not satisfied for some concept. In other words, there are several concepts with the same name having different structures in different ontologies.
- Inconsistency on the relation level: Between the same two concepts there are inconsistent relations in different ontologies.
- Inconsistency on the instance level: The same instance belonging to different ontologies does not satisfy the instance integration condition.

4. METHODS OF INCONSISTENCY RESOLUTION

4.1. INCONSISTENCY ON CONCEPT LEVEL

Inconsistency at this level is when we have ontologies with different structures, built according to the same concept. This situation is very frequent inasmuch as each ontology serves to represent knowledge of a specific object and therefore concepts are defined from the point of view of the needs resulting from the subject [7].

Definition is as follows:

Let O_1 and O_2 be $(\boldsymbol{A}, \boldsymbol{V})$-based ontologies. Let the same concept c belong to O_1 as (c, A^{c1}, V^{c1}) and belong to O_2 as (c, A^{c2}, V^{c2}). We say that the inconsistency takes place on the concept level if $A^{c1} \neq A^{c2}$ or $V^{c1} \neq V^{c2}$.

Example: We can have concept *Employee* defined by attributes:

{Name, Age, Sex, Job}

whereas in the other ontology it is defined by attributes:

{Id, Name, Surname, Date_of_birth, Sex, Occupation}

Inconsistencies can also result from the fact that in two ontologies based on the same world (A, V) there are differences in semantics of the same attributes in two concepts. The solution to this problem is as follows:

For the given set of pairs

$X = \{(A^i, V^i)$: (A^i, V^i) *is the structure of concept c in ontology Oi*

for $i = 1, ..., n\}$

it is necessary to determine a pair (A^*, V^*) *which best represents the given pairs.*

The solve problem of inconsistency on this level we can use postulates defined by NGUYEN N.T. [7].

4.2. INCONSISTENCY ON RELATION LEVEL

The ontology can define relationships between classes, and so the way in which objects of the modeled fragments of reality come together will interact. Relations between concepts are included in set $\boldsymbol{R}$ of the ontology definition, whereas the relationships between them may be included in set $\boldsymbol{Z}$ [7]. Conflict at this level occurs when in two ontologies are defined conflicting relationships between classes. We can also say that by inconsistency on the relation level we understand the situation where referring to the same two concepts in one ontology there is defined a relation which is not present in another ontology [7].
Definition is as follows:

Let O and O' be $(\boldsymbol{A}, \boldsymbol{V})$-based ontologies containing concepts c and c'. Let $r \in R_O(c, c')$ and let $r' \in R_{O'}(c, c')$. Inconsistency on this level occur if the coexisting of relations r and r' between concepts c and c' may cause nonsatisfaction of the constraints defined for ontology O or O'.

Example 9.3. Assume that in the example ontologies exists concepts *Employee* and *Course*. In the first ontology there is defined the relation *Teaching* (i.e., an employee teaches a course), and in the second ontology instead of the relation *Teaching* there is also defined the relation *Completing* (i.e., an employee has completed a course). Thus we have inconsistency on the relation level.

For solving inconsistency of ontologies on the relation level, we need to:

For a given profile
$X = \{R_{Oi}(c, c'): i = 1, ..., n\}$
it is necessary to determine set $R(c, c')$ *of final relations between c and* c', *which best represents the given sets.*

If between relations on concepts c and c' there is a lack of relationships, then the solution for this problem seems to be simple. One can determine $R(c, c')$ as the sum of sets $R_{Oi}(c, c')$, or select to $R(c, c')$ only those relations from $R_{Oi}(c, c')$, which appear most frequently. However, the solution will not be so simple if we take into account the relationships between these relations.

4.3. INCONSISTENCY ON INSTANCE LEVEL

On this level we assume that two ontologies differ from each other only in values of instances. That means they have the same concepts and relations between them [7]. Conflicts at the instance level is the most complex aspect of the discussed issues. On-

tology instances are representations of physical objects, which are expressed in the ontologies (and therefore in their respective collections instance). According to the formal definition, each such real object is described by a set of attributes and corresponding values. These attributes form a coherent description of selected characteristics of objects and instances, and are strongly dependent on the ontologies based on real world.

Definition is as follows [7]:

Let there are two instances of two ontologies based on the world ($\boldsymbol{A}$, $\boldsymbol{V}$):
$(i, v) \in Ins(O_1, c)$ *and* $(i, v') \in Ins(O_2, c')$
Inconsistency occurs when $v_a \neq v'_a$, for $A^{c1} = A^{c2}$ and $V^{c1} = V^{c2}$ and $a \in A^c \cap A^{c'}$.

Example: As an example, consider the ontologies of two systems for companies offering their services. In one of the systems we have, there is the concept *Customer* and in the second the concept of *Debtor*. The structure of the concept *customer* is as follows

$$(\{Client\_id, Name, Address, Personal\_identity\_number, Contract\_number\}, V^{\text{Client}})$$

and concept *Debtor* which has the following structure:

$$(\{Debt\_id, Name, Personal\_identity\_number, Contract\_number\ \}, V^{\text{Debtor}})$$

The value of this instance referring to concept *Client* is:

$$[10; Kowalski; Nysa; 120613425; 77/14]$$

The value of this instance referring to concept *Debtor* is:

$$[10; Kowalski; 120613425;125/10]$$

Thus there is inconsistency on the instance level because both instances referring to attribute *Contract_number* but in the first case there is a value "*77/14*" and in the second case there is value "*125/10*". The conflict arises from the fact that the same person in the first company is listed as a customer under contract No. 77/14 and in the other company as debtor of the contract number 125/10.

For solving inconsistency of ontologies on the instance level consensus methods seem to be very useful. These methods deal with determining for a set of different versions of data (so called a conflict profile) such a version which best represents the given versions. On that level of inconsistency we have to deal with a set of versions of the same instance. A version of an instance (i, x) can be represented by the value x.

The task is to determine a version (a value) which best represents the given versions. For this kind of consistency the consensus problem can be defined as follows [7].

Given a set of values $X = \{v_1, ..., v_n\}$ *where* v_i *is a tuple of type* $A_i \subseteq \boldsymbol{A}$,
that is: v_i: $A_i \to V_i$
for $i = 1, ..., n$ *and* $V_i = \bigcup_{a \in A} V_a$ *one should find a tuple v of some type*,
which best represents the given values.

For the defined structures and particular postulates several algorithms for consensus determination have been worked out. It has turned out that in this case the best criterion for determining value v is

$$\sum_{i=1}^{n} d(v, v_i) = \min_{v' \in TYPE(A)} \sum_{i=1}^{n} d(v', v_i), \tag{5}$$

$$A = \bigcap_{i=1}^{n} A_i ,$$

where and d is a distance function between tuples.

Value v can be assumed to be the value of instance i in the final ontology.

5. CONCLUSIONS

Ontologies are a very useful instrument in the conceptualization of information systems. In database systems allow to create a structure based on elements such as classes, attributes, relationships, that will be the basis for further processing. But in order to fulfill their role, they must be properly modeled. Then they become the perfect tool for performing integration of information systems. Unfortunately, there are various problems related to inconsistency processing, such as ontology mismatch, ontology conflict, ontology merging, or ontology integration. The paper presents some of the known aspects of inconsistencies during the integration of ontologies. Presented levels of inconsistencies may be one of many problems that arise during the process of ontology matching. Future work will focus on inconsistency of integration of fuzzy ontologies, for whom ontologies presented in this work are the basis.

REFERENCES

[1] EUZENAT J., SHVAIKO P., *Ontology matching*, Springer, 2007.
[2] GLIŃSKI W., *Ontologie – próba uporządkowania terminologicznego chaosu*, SBP, Warszawa 2005.
[3] GOCZYŁA K., *Ontologie w systemach informatycznych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2011.

[4] GOŁUCHOWSKI J., FILIPCZYK B., *Perspektywy wykorzystania ontologii w procesie przetwarzania języka naturalnego w systemach zarządzania wiedzą*, Wydawnictwo Akademii Ekonomicznej im. Karola Adamieckiego w Katowicach, Katowice 2006.
[5] GRUBER T.R., *A Translation Approach to Portable Ontology Specifications*, Journal Knowledge Acquisition, 1993, p. 2.
[6] GRUBER T.R., *The role of common ontology in achieving sharable, reusable knowledge bases*, [in:] J.F. Allen, R. Fikes, E. Sandewall, (eds.), KR, Morgan Kaufmann, 1991.
[7] NGUYEN N.T., *Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
[8] NGUYEN N.T., *Conflicts of ontologies – classification and consensus-based methods for resolving*, [in:] Proceedings of the 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II, KES '06, Springer-Verlag, Berlin, Heidelberg 2006.
[9] NGUYEN N.T., HAI BANG TRUONG, TRONG HAI DUONG, *A Hybrid Method for Fuzzy Ontology Integration*, Cybernetics and Systems, 2013.
[10] PINTO H.S., PEREZ A.G., MARTINS J.P., *Some issues on ontology integration*, [in:] V.R. Benjamins (ed.), Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods, CEUR Publications, Vol. 18, 1999.
[11] SHVAIKO P., EUYENAT J., GIUNCHIGLIA F., STUCKENSCHMIDT H., MAO M., CRUY I.F., (eds.), Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010, [in:] P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, M. Mao, I.F. Cruz, (eds.), OM, CEUR Workshop Proceedings, Vol. 689, CEUR-WS.org, 2010.
[12] SHVAIKO P., EUYENAT J., GIUNCHIGLIA F., STUCKENSCHMIDT H., NOY N.F., ROSENTHAL A., (eds.), Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009), Chantilly, USA, October 25, 2009, [in:] P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, N.F. Noy, A. Rosenthal (eds.), OM. CEUR Workshop Proceedings, Vol. 551, CEUR-WS.org, 2009.

data integration, data conflict handling,
Schemaless data integration

Bolesław SZAFRAŃSKI,
Grzegorz BLIŹNIUK, Walid CHERIFI*

A BROAD OVERVIEW OF DATA INTEGRATION SYSTEMS, FROM THE PAST TO THE FUTURE

Due to the significant development of the Internet, finding information from distributed sources over the network is becoming increasingly difficult. Indeed, due to the revolution of new information technology, companies as well as individuals have a huge amount of data. These data are stored in sources that are almost all the time heterogeneous and autonomous. Data integration systems harmonize data from different independent sources into a single coherent representation. It aims to provide a unified access to a set of data sources in a specific application domain, where users can request their queries. The aim of this paper is to present an overview of data integration problem by providing a general definition to this problem, and classifying its different approaches that are proposed by the IT community. Besides, it introduces new challenging issues in order to efficiently improve the quality of data provided by data integration system, and to keep pace with new technology.

1. INTRODUCTION

Data integration is a vast research area, which is concerned with the problem of combining data located in heterogeneous, autonomous and distributed data sources, and providing the user with a uniform and transparent access to all these data sources over a single, unified and homogenous view. The emergence of the Web and the permanent enlargement of business organizations, often arising when more than two similar enterprises need to merge their data sources, impose an increasing necessity in integrating and managing huge volume of data, collected from a number of heterogeneous and disparate data sources. Such needs are also shown by many other applications, such as scientific projects, where groups of scientists are independently collecting data and trying to collaborate with one another. Moreover, integrating data sources is the key success of business intelligence systems, especially with the exponential growth of autonomous data sources over the Internet and enterprise Intranet which

* Faculty of Cybernetics, Military Unversity of Technology. Kaliskiego St. 2, 00-908 Warsaw, Poland.

oblige companies to be able to access data located over the Internet within their Intranets for better mining and effective decision making. Furthermore, information integration plays an important role in internet search. For instance, querying the deep web that is composed of a huge number of data sources accessible through Web forms. Here, the user would like to be able with a single query to find accurate data without needing to spend time in accessing all data sources separately. Finally, data integration has also raised importance in the life sciences, where the number, size, and complexity of life science database continue to grow. Therefore scientists and researchers in the field of proteomics, genomics, clinical medicine and drug discovery need a complete picture for all their data [1].

In this paper, we focus on the principal approaches proposed by the IT community to deal with the problem of data integration and, in particular, we introduce four criteria used to classify different information integration solutions. Our purpose is to present an overview of data integration issues and show some proposed solutions to solve them. Besides, we provide an outlook to new challenges in the area of data integration.

The remainder of this paper is organized as follows. Section 2 defines the problem of data integration. Section 3 presents possible classifications of integration system that have been suggested by IT researchers to address the integration issue. Section 4 presents new tendency for this problem. Finally, Section 5 concludes the paper.

2. THE PROBLEM OF DATA INTEGRATION

Due to the significant development of the Internet, finding information from distributed sources over the network is becoming increasingly difficult. Indeed, due to the revolution of new information technology, companies as well as individuals have a huge amount of data. These data are stored in sources that are almost all the time heterogeneous and autonomous. Data integration systems harmonize data from different independent sources into a single coherent representation. It aims to provide a unified access to a set of data sources in a specific application domain, where users can request their queries. This is done by resolving the heterogeneities and offering to the disparate sources an integrated view. Then, users are able to pose queries over this uniform view without having to spend a lot of time in searching the set of separate data sources.

The data sources which participate to the integration process may either belong to the same organisation intranet or just spread out all over the internet. Generally, each of the repositories is independently designed for diverse application objectives. Besides, the sources are not necessarily only databases; they can range from database systems and legacy systems (old and outdated systems that are difficult to migrate to a modern technology) to structured/unstructured files, web services and forms on the web.

The inputs of data integration problem are a set of distributed, autonomous and possibly heterogeneous data sources, where each one has probably its own schema and population that are different from others. Its output is a single integrated view of data sources, where users can submit their queries without any prior information about the structure of the disparate sources. The construction of such system remains a difficult task because of the following principal reasons:

1. **The scalability of sources:** Data integration is already a challenge for small number of sources, generally fewer than 10 data repositories, but the challenges are exacerbated when the number of sources increases [2]. Indeed, the number of dispersed data sources participated in the integration operation is still growing. This exponential growth increases the quantity of information produced in the world by 30% every year [3], mainly in domains such as Internet of things where objects like sensors and smartphones generate a huge amount of data every day, E-commerce, life sciences, social networks, etc. These mountains of data cannot be integrated manually. However, they necessitate automatic solutions which are difficult to build.
2. **The autonomy of sources:** In its current form, most data available in the world today is spread out across disparate data sources which are entirely autonomous and isolated from each other. The owners of sources may not always be aware of other sources referencing to their data, or integration system accessing to their sources. Furthermore, they are free to choose their own schemes and models, to modify their design, and to update and/or remove some data at any time, without having to notify any central administrative entity. These facts make the autonomy of sources a critical issue for the task of integration; since it engenders other difficulties such as instability and heterogeneity of sources.
3. **The instability of sources:** the instability and dynamicity of sources are due to the fact that almost all sources are autonomous. New data sources appear every day, others disappear. The format and contents of the sources may change over time. A new attribute may be added, or an existing attribute may be dropped. A powerful integration system should be able to surmount all these unexpected change and provide relevant and accurate information to the requesters.
4. **The heterogeneity of sources:** The heterogeneity of data sources is the hardest obstacle which faces the problem of data integration; it may occur in various forms, ranging from syntactic heterogeneity and differences in data structures to semantic heterogeneity and variances in the meaning.
   – *Syntactic heterogeneity* is a result of differences in data models where the information may be modelled according to different paradigms (Relational tables, XML tree...), languages (SQL for relational databases, XQuiry for XML files...) and representations which involve the problem of deferent type (e.g. char vs. string) or deferent format (e.g. 02-03-2014 vs. 02/03/14).

- *Structural heterogeneity* is related to the choices of modelling. It includes representational heterogeneity which is due to conflicts in the way related information is modelled in different schema, and schematic heterogeneity that is the result of encoding concepts using different elements of data model.
- *Semantic heterogeneity* presents a real logical bottleneck in developing integration systems [4]. It is due to different interpretations of real world objects by designers according to their understanding. Another important reason of this heterogeneity is the fact that disparate data sources are barely created for the exact same purpose. This type of heterogeneity engenders different sorts of conflicts [5]:
  a. Naming conflicts: This kind of conflicts refers to the noticeable difference in naming schema of information. The most frequent case is the presence of synonyms where several terms are used to describe the same concept (e.g. Family_Name vs Last_Name), and homonyms when the same word is used to denote different concepts (e.g. name of person vs name of company).
  b. Scaling conflicts: occur when different reference systems are used to measure a value (for instance price of a product can be given in zloty or in euro).
  c. Confounding conflicts: These conflicts occur when information items seem to have the same meaning, but differ in reality, e.g. due to different temporal contexts.

By analysing the principal problems above-mentioned, we can figure out that constructing data integration systems requires overcoming the differences in modelling, semantics and capabilities of sources, handling the variety and the massive amount of available data, resolving the possible inconsistencies with data heterogeneity from different sources, and providing the users with a unified, reconciled view of all these repositories. In this sense, and in order to efficiently surmount these issues, we present in the next section a brief description of the different classification proposed by the IT community for the integration solutions.

3. CLASSIFICATION OF DATA INTEGRATION SYSTEM

For more than twenty years, data integration has been deeply investigated by the IT community. Researchers in different research area (database, artificial intelligence…) have been addressing data integration issues in diverse perspective. Several approaches have been proposed, various technics have been developed and numerous projects have been produced, making it hard to provide a precise classification of the previous work based on shared criteria. By exploring the literature and the proposals work in the field of data integration, we can show the following criteria proposed by the community [6] in order to classify the existing data integration systems:

3.1. DATA REPRESENTATION

This criterion shows how data is retrieved from disparate sources. There are two broad types of data representation, based on whether the data from different sources is duplicated and materialized into a central source generally called warehouse, or accessed virtually and on demand from a common logical schema. Different approaches have been proposed for this purpose, the most used are:

- **Materialized data integration:** The global schema of a data integration system can be fully materialized. It means that a new central repository, called warehouse, is developed by a data management system and a copy of all the data which correspond to the global schema is saved in this repository. The copies of data which are stored at the warehouse are pre-processed in a complex way before storage; it is usually referred to as extract, transform and load (ETL) tools that periodically extract data from the sources, and load them into the warehouse after doing some complex transformations (cleaning, aggregation, and value transformations). The data warehouse eliminates several problems of integration, mainly the excessively long server response times, the network clogging, or the sources unavailability. Queries can be also more easily optimized, and the data transformed at the user's discretion. In the other hand, the warehouse has to be updated regularly (hourly, daily, etc.), based on the specific application requirements, but since data must be extracted, transformed and loaded into the warehouse, there is an element of latency. As a consequence, between two updates, applications that depend on this data cannot be considered as being up-to-date.
- **Virtual data integration:** this approach is based on mediator system; a mediator is a software component that supports a virtual database which is characterized by its global (mediated) schema [6]. Unlike in the warehouse approach where all data from disparate sources are duplicated into central repository, the virtual approach is oriented especially towards queries rewriting, where the mediator system is responsible for reformulating and translating at real-time a query posed by a user over the mediated schema into a sub-queries on the local schema of the underlying data sources. The key to build a virtual data integration system is the semantic mappings, which relate the schemata of the data sources to the mediated schema. The semantic mappings specify how attributes in the sources correspond to attributes in the mediated schema. Moreover, they show the way how query over the global schema can be translated to queries over the local schemas. The data flow between mediators and data sources is provided by software components called wrappers, which their roles are to send queries to a data source, receive answers, and possibly apply some basic transformations on the answer. Results from subqueries are finally unified by data reconciliation (data fusion) techniques, which aim to resolve data conflict in the instance level, and provide complete and concise answers of all existing data: complete because no object or attribute is forgotten in the result; concise because each object is provided only once [7].

3.2. SENSE OF MAPPING

In data integration system, there are two types of schemas: the global schema where the goal is to describe the logic of data in the highest layer of data integration system, and the local schemas which describe the logic of data in the local data sources. Schema mapping describe the relationship between the attributes of the global schema and those of the sources' schema. When a query is formulated in terms of the global schema, we use the mappings to re-write the query into appropriate queries on the sources. The result of the reformulation is a logical query plan. There are several approaches to map between a global schema and the schemas of the local sources. *GaV* (*Global-as-View*), *LaV* (*Local-as-View*), *GLaV* (*Generalized Local-as-View*), and *BaV* (*Both-as-View*) represent the known mapping methods. *GaV* and *LaV* are the main methods; *GlaV* and *BAV* are mixed approaches.

- **Global-as-View (*GaV*):** In this approach, the predicates in the global schema are described as views over the predicates in the local schemas. In other words, each entity in the global schema is expressed as a query over the local sources [8]. *GaV* has the power of simple query answering. In fact, it is just a simple reformulation of the global query by rules unfolding. However, this approach is not scalable; adding a new source can cause enormous modification to the schema mapping. Thus, the mapping predicates must be recreated in this case.
- **Local-as-View (*GaV*):** In this approach, the mapping expresses the predicates of the local schemas as a function of the predicate of the global schema. In other words, data sources are defined as a set of views over the global schema. In this case, the mapping associates for each element of the local schema a query over the mediator [8]. The main advantage of this approach is the facility offered to add new sources. That is because sources can register independently of each other, since a source's mappings do not refer to other sources in the system. Nevertheless, query evaluation is non-trivial anymore; this is mainly due to its technical implications of query answering. Moreover, any change in the global schema can be difficult to pass at defining views sources. That is why this approach is advocated in cases where the global schema is not subject to frequent changes.

3.3. MATCHING AUTOMATION

Schema matching is the task of identifying semantic correspondences and producing matchers between two given schemas, ontologies or XML messages format, each consisting of a set of discrete entities. Solving such matching problems is a fundamental key for data integration systems. Schema matching can be done in a variety of ways:

- **Manually:** here, the designer of data integration system (generally helped by domain experts) is the one who determines the relationship between schemas

manually supported by some graphical point-and-click interface. Obviously, this kind of matching is broadly efficient when the number of sources is small and each source has a good description. Otherwise, it is a time-consuming, tedious, and error-prone process for large-scale data integration (two huge schemas or a large number of schemas).

- **Semi-automatically:** To surmount the difficulty engendered by the large number of data sources, it is preferable to use at least a partial automatic schema matching process. To do so, many researches have been done in order to determine semi-automatic methods that help designers to select similarities between schemas and reduce theirs manual works. Here, data integration systems are responsible to find the matchers under human supervision.
- **Automatically:** In the typical case, only systems shall be responsible to determine the link between schemas without human intervention. To enable automatic integration, explicit representation of data meaning is necessary to overcome the problem of semantic heterogeneity. Thus, most recent integration systems use ontologies, which are a specification of conceptualisation [6].

3.4. DATA CONFLICTS HANDLING

After the construction of the mediator and the mapping schemas for data integration system, users submit queries over the global schema, and wait for a complete, correct and concise answers coming from several different data sources. These answers which represent real-world entities may occur in various repositories redundantly, and they may also contain many conflicting and missing attributes. In order to deal with this issue, data integration systems should have data fusion capabilities, which aim to combine records that refer to the same real-world entity by merging them into a single representation and resolving possible conflicts. Many strategies were developed to handle data conflicts, some of them were repeatedly mentioned in the literature. These conflicts handling strategies can be classified into three main classes based on the way of handling data [7]:

- **Conflict-ignoring:** this strategy does not make any decision to deal with data conflicts and sometimes it is not aware of the existence of conflicts. When employing such a strategy one not even needs to look for, detect or be aware of data conflicts in data. An example for an ignoring strategy is PASS IT ON, which takes all conflicting values and passes them to the user without changing them, and then let him decide which data is the most appropriate. Another example is CONSIDER ALL POSSIBLITIES that generates all possible combinations of values, on which some of them do not even exist in the sources, and then let the user choose which data is correct. This strategy is applicable at all times and in all situations and is easy to implement.
- **Conflict-avoidance:** This strategy is aware of conflicts but do not perform individual resolution for each conflict. Instead, they handle conflicting data by

applying a unique decision equally to all data. Indeed, decisions are made before regarding the data value, which prevents any possibility of hesitation or making decisions about the values to be chosen in the data fusion step before showing the final result to the user. The most famous techniques implementing this strategy are: 1) TAKE THE INFORMATION which takes the existing information and leave aside those who have NULL value. 2) NO GOSSIPING which takes only into consideration the answers from data sources that fulfil the constraints or conditions added into the user query, and ignoring all of the inconsistent answers. 3) TRUST YOUR FRIENDS in which data are preferred to be taken from one data source over another based on the user preference in the query.

- **Conflict-resolution:** This strategy is the most expensive among conflicts handling strategies. It examines all the data and metadata before deciding on how to handle conflicts. Conflict resolution strategy can be classified into deciding and mediating strategies:
  - *Deciding Strategy:* This sub-strategy resolves conflicts by deciding which value to present among various conflicting values [9]. This decision may be based on instance values, where two techniques are well known: CRY WITH THE WOLVES on which the system selects the value that appears the most among the conflicting values, while the other technique ROLE THE DICE takes a random value among the conflicting ones. The decision may also be based on the metadata, where the famous technique is KEEP UP TO DATE which uses the timestamp metadata about the data sources, attributes and the data to select from the conflicting values.
  - *Mediating data:* This strategy present a value that may not be necessary one of the exiting conflicting values, by taking into account the data values and/or the stored metadata. As an example of a technique that implement this strategy we can cite: MEET IN THE MIDDLE which invents a new value from the conflicting values to represent them by many ways: like taking the mean of the values.

4. NEW CHALLENGES IN DATA INTEGRATION

In its current form, and with the growth in global internet use, data is generated in huge quantity whether by humans or machines. Due to this explosion of data and the arrival of new data types in such amazing speed, many companies are currently innovating in the domain of data management in order to design their own systems. Indeed, these companies need data storages to be able to store and manage big data efficiently, so the traditional relational databases are confronting some novel challenges. Principally in large scale and high-performance applications, such as Social Networks and Internet of Things, using the relational database to store and query dynamic user

data has seemed to be inadequate. In this situation, NoSQL database created and gained traction.

In fact, one of the most important reasons to use NoSQL is to help solve the big data problem which is a much larger problem than storage [10]. Big data is often referred as the "3V":

- Data volume: the generation and collection of a huge mass of data.
- Data velocity: lots of data coming in very quickly, possibly from different locations.
- Data variety: storage of data that is structured, semi-structured and unstructured.

The power of NoSQL databases consists in their scalability, availability and flexibility. The latter characteristic means that data in NoSQL databases has a flexible (schema-less) data model. Unlike relational databases, where designer must determine and declare a table's schema before manipulating data, NoSQL data storages do not enforce the structure. This flexibility facilitates the mapping of the stored data to a real-world entity.

These advantages of NoSQL databases engender new challenges in the problem of integrating several different data sources in both schema and instance level, and lead to rethinking about the integration system approaches, in order to enhance its performance and dynamicity.

Since NoSQL databases are schema-less, it is difficult to build a stable global schema for all these sources. The integration of these kind of sources necessitates to investigate the possibility of integrating sources without the schema match step (schema-less data integration), and to try to discover the relation between sources incrementally, this could be done by focusing in the instance level in order to provide data of high quality to the requester.

Another interesting issue consists in the integration of cloud data sources. Indeed, many companies are using the Cloud in some form today, with many companies embracing the hybrid IT model and running applications both in the cloud and on premise. Data integration in the cloud is a hard task, since data is hosted natively on it. Data aren't just stored in different places; they may be subject to completely different data-integrity rules, and may not be easily reconciled into a single stream without major heavy lifting. A deep investigation should be carried out in order to build a powerful system allowing hybrid integration between data storage in the cloud and those in the enterprise intranet.

5. CONCLUSION

In this paper we provided a broad overview of data integration. We started by presenting a global definition of data integration. We showed the benefit of integrating several distributed sources and the principal obstacles that face data integration process. Secondly, we introduced four proposed criteria which allow classification of the

main data integration systems. This classification facilitates the understanding of manipulated systems. Finally, we presented some new and emergence challenges in the area of data integration system.

Data integration has been investigated for more than twenty years. Although many problems have been solved, there are issues that still require to be investigated. These are harnessed by the dynamic nature of today's application domains. Future prospective of data integration is mainly in the large scale level, where the large number of sources with the huge amount of data is still growing. This dramatic increase engenders new promising challenges which are principally related to flexibility, accuracy and performance.

REFERENCES

[1] SHADBOLT N., HALL W., BERNERS-LEE T., *The semantic web revisited*, IEEE Intelligent System App., 2006, 21(3), 96–101.

[2] BUSSE S., KUTSCHE R.D., LESER U., WEBER H., *Federated Information Systems: Concepts, Terminology and Architectures*, Tech. Rep. 99-9, TU Berlin, Berlin 1999.

[3] DONG X., NAUMANN F., *Data Fusion – Resolving Data Conflicts for Integration*, PVLDB, 2009, Vol. 2, No. 2, 1654–1655.

[4] HAKIMPOUR F., GEPPERT A., *Global Schema Generation Using Formal Ontologies*, [in:] Proceedings of 21th International Conference on Conceptual Modeling (ER '02), 2002, 307–321.

[5] ANOKHIN P., ENGINEERING S., Motro A., *Use of Meta-data for Value-level Inconsistency Detection and Resolution During Data Integration.ech*, Rep. ISETR-03-06, Department of Information and Software Engineering. George Mason University, USA, 2003.

[6] BAKHTOUCHI A., CHAKROUN C., BELLATRECHE L., AĬT-AMEUR Y., *Mediated Data Integration Systems Using Functional Dependencies Embedded in Ontologies*, [in:] Recent Trends in Information Reuse and Integration 2012, 2011, 227–256.

[7] BLEIHOLDER J., NAUMANN F., *Data fusion*, ACM Computing Surveys, 2008, Vol. 411, No. 1, 1–41.

[8] MANOLESCU I., FLORESCU D., KOSSMANN D., *Answering XML queries over heterogeneous data sources*, [in:] VLDB, 2001.

[9] KATSIS Y., DEUTSCH A., PAPAKONSTANTINOU Y., VASSALOS V., *Inconsistency Resolution in Online Databases. In Data Engineering (ICDE)*, 2010 IEEE 26th International Conference on IEEE, 2010, 1205–1208.

[10] CHEN M., MAO S., Liu Y., *Big Data: A Survey*, [in:] Springer Science+Business Media, New York 2014, DOI: 10.1007/s11036-013-0489-0.

efficient and safe storage space configuration,
input/output operation performance,
disk striping with B-tree filesystem

Mateusz SMOLIŃSKI*

MODERN MULTI-DISK DATA STORAGE USING BTRFS

This paper introduces various software multi-disk storage configuration in modern GNU/Linux operating system. In all compared storage configurations new B-tree filesystem was used, which provides innovative data protection capabilities. Completed research include many multi-disk storage configurations with various management software RAID, logical volume LVM and direct management with BTRFS. The research results allow to choose the appropriate disk manager and data storage space configuration for BTRFS, that provides best I/O operations performance for fixed number of disks. Presented results allow more effective management of system resources for modern multi-disk data storage configurations in GNU/Linux operating system i.e. servers, workstations.

1. INTRODUCTION

1.1. DATA STORAGE CONFIGURATION IN OPERATING SYSTEM

The main goals of modern operating systems are data storage and data processing [1, 2]. Storage space in operating system is usually divided into zones which increases stored data security and configuration flexibility. Each zone can customize configuration of the storage space for data storage characteristic such as the choice of block for device type and selection of filesystem type and its configuration. Each data storage zone base on physical block devices like hard drives (i.e. disks) providing non-volatile memory. Increasing the number of disks used in zone configuration can affect performance of execution input/output operations in operating system.

Kernel of modern operating system GNU/Linux includes various implemented management mechanisms for many disks storage zone configuration. These mechanisms enable a creation of logical block device known as volume, whose blocks address space size can exceed the address space of any used disk. The volumes manage-

* Institute of Information Technology, Technical University of Lodz, ul. Wólczańska 215, 90-924 Łódź, Poland.

ment in each Linux kernel mechanism requires rules, that provide the organization of stored data. Data organization rules for volume affects both the sector addressing and input/output operation performance. Among the considered Linux kernel mechanisms of disk space management are: software RAID (Redundant Array of Independent Disk), LVM (Logical Volume Manager) and BTRFS (B-Tree FileSystem) [1].

This paper includes comparison of Linux kernel software disk array and volume management mechanisms, its individual characteristic and impact on I/O performance in various testing multi-disk storage configuration scenarios using BTRFS. The uniform test environment was prepared in order to ensure reliable comparison results. The obtained results facilitate the configuration of the efficient local storage space in Linux operating system.

2. STORAGE TESTING ENVIRONMENT

2.1. SOFTWARE TOOLS FOR TESTING

For research was used Linux operating system from Fedora 18 distribution for 64-bit computer architecture (kernel version 3.6.10 x86_64) [1]. Hardware used for testing: computer with i7-2600 CPU with six identical, connected SATA-2 Western Digital disks model WD500AAKX with 465GB capacity and 16MB cache. Operating system was installed on first of six disks, other five were used in I/O performance testing of different storage configuration scenarios.

Testing scenario includes fixed number of disks and one of software disk management mechanism implemented in Linux kernel. Additional parameters for scenario according to software RAID and LVM is allocation unit size (chunk size).

In each testing scenario there have been 21 cases of writing and reading 32 GB set of data to/from regular files with standard program tool spew in 1.0.8 version. In *n*-th test case all I/O operations were performed using one spew process executing I/O operation with 2n-1 block size, therefore the range of block size in single I/O operation was from 1KB up to 1GB. Test unification require to use the same BTRFS filesystem with 4KB allocation size (concern: node, leaf and block), for this purpose btrfs-progs in 0.20 version was used. In all testing scenario filesystem was mounted in standard asynchronous read-write mode and only one BTRFS subvolume was created.

Using software implemented in Linux kernel and program tools from mdadm in version 3.2.6 in testing storage configuration scenarios software disk array level 0, 5 or 6 were created [3]. The RAID with 0 level marked as RAID0 applies only striping, where stripe size depend on number of disk included in array and chunk size. Level 5 disk array uses additional metadata to increase data security and reliability, which retains data even one disk fails. RAID level 5 (marked as RAID5) has data capacity

($N - 1$)**S*, where *S* is minimal capacity of disk from all active array disks. Level 6 disk array also uses additional metadata, data are available even if two disk fails. RAID level 6 (marked as RAID6) has data capacity ($N - 2$)**S*, where *S* is minimal capacity of disk from all active array disks.

Table 1. The parameters of tested
multi-disk storage configuration scenarios managed by disk array RAID

RAID Level	Number of RAID active disks	Chunk size
0	2, 3, 4, 5	16 KB, 64 KB, 512 KB
5	3, 4, 5	16 KB, 64 KB, 512 KB
6	4, 5	16 KB, 64 KB, 512 KB

According to table 1, for the research purposes was distinguished 27 testing storage scenarios with various software RAID configuration.

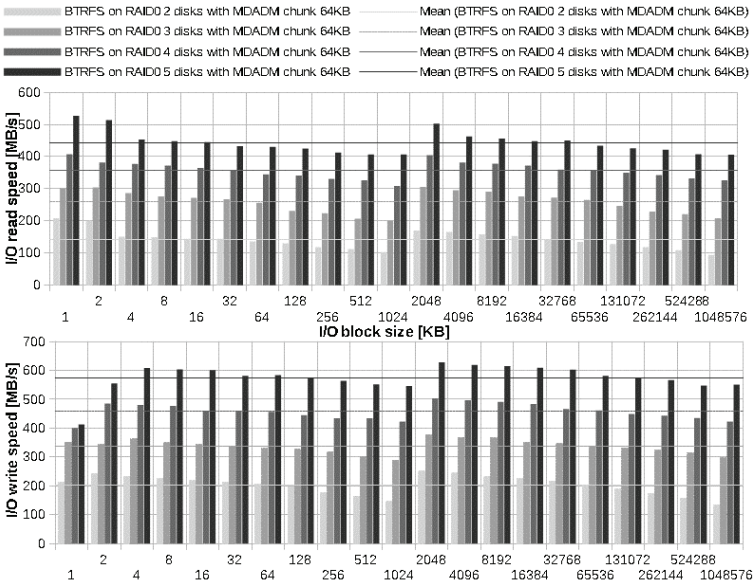

Fig. 1. I/O performance for BTRFS localized in software RAID level 0 with 64 KB chunk size

2.2. STORAGE WITH SOFTWARE RAID

The measured I/O operation performance results for storage configuration scenarios with software RAID level 0 with example chunk size 64 KB were presented in fig. 1, and for software RAID5 and RAID6 respectively in fig. 2 and 3.

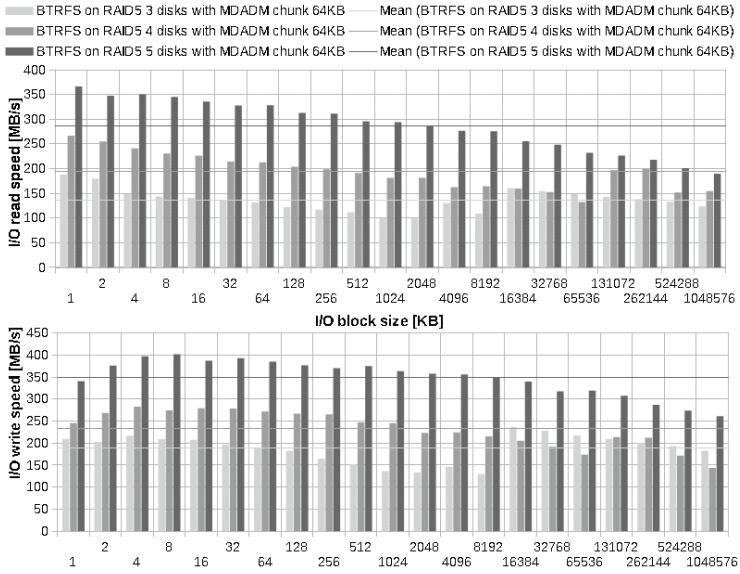

Fig. 2. I/O performance for BTRFS localized in software RAID level 5 with 64 KB chunk size

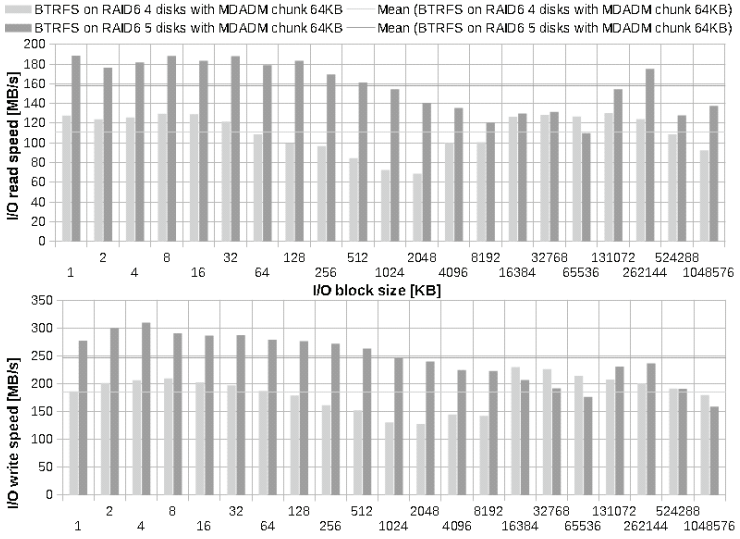

Fig. 3. I/O performance for BTRFS localized in software RAID level 6 with 64 KB chunk size

2.3. STORAGE CONFIGURATION WITH LOGICAL VOLUME MANAGER

Kernel in Linux operating system includes Device Mapper component, which allows to create block devices with complex sector addressing and additional function-

ality. External logical volume manager LVM uses Device Mapper, in research was used LVM in 2.02.98 version. Actual LVM implementation in Linux kernel assures: flexible volume size configuration, various block allocation policy to volume, volumes with special functionality (i.e. snapshots).

LVM introduce various objects: physical volume, volumes groups and logical volume. All LVM object configuration is persistent. Physical volumes store volume group data to which they were assigned. Disk as other block device can be converted to physical volume. In volume group logical volume are created, however volume size is integral multiplication of extent size, which is allocation size determined by volume group. Default LVM extent unit size is 4MB [4].

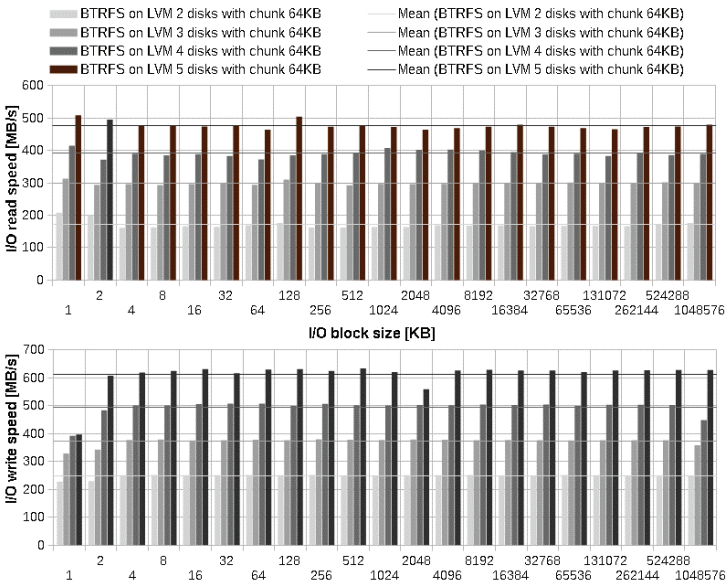

Fig. 4. I/O performance for BTRFS localized in logical volume with LVM managed striping with 64 KB chunk size

By using LVM data storage space in logical volume could be distributed between many physical volumes belonging to the same volume group. This LVM allocation policy base on striping, where data stripe size is defined by number of used physical volume and chunk size. However, this allocation policy allows increase performance of I/O operations performed by logical volume if its physical volume were placed on different disks. In each testing scenario with LVM logical volume was created with allocation policy using various number of physical volumes localized on separate disk (from 2 to 5) and different chunk size (16 KB, 64 KB or 512 KB). Results for I/O operation performance managed by single LVM volume with striping with example 64KB size was presented in fig. 4.

2.4. B-TREE FILESYSTEM

Another tested multi-disk storage scenarios include volume created and managed directly by BTRFS [5, 6]. This modern filesystem has internal block device manager, that provides management of disk storage spaces. BTRFS offers various allocation policy for managed subvolumes. When filesystem is created separate policy can be configured to its metadata and stored data [5, 7].

For research purposes only raid0 allocation policy was used, which base on data striping during I/O operation performance across all disks managed by b-tree filesystem. Unlike LVM and RAID, BTRFS directly manages disk space using internal allocation units without specifying chunk size. Figure 5 presents I/O performance results statement for directly managed disks by BTRFS in multi-disk storage configuration.

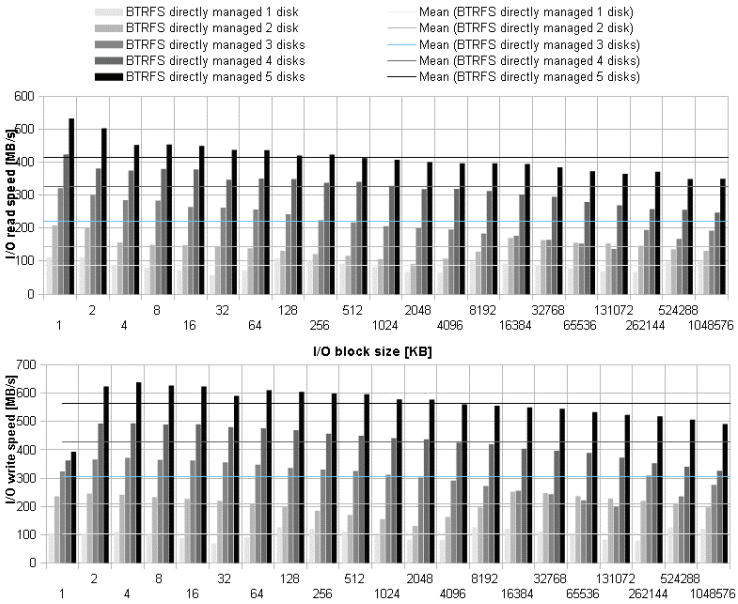

Fig. 5. I/O performance for BTRFS directly manages disks striping

Modern b-tree filesystem when directly manages of many disks has specific I/O performance characteristic to other software disk managers implemented in GNU/Linux kernel. Basic difference is I/O performance dependence to number of managed disk. For software RAID or LVM each storage configuration scenario with more disk shows better I/O performance in all testing cases. In example scenarios with BTRFS directly manages disks, in testing case with I/O operation data size 32MB read and write speed for two and three disks are similar (fig. 5).

3. IMPACT STORAGE SPACE CONFIGURATION ON I/O PEFORMANCE

3.1. I/O OPERATION PERFORMANCE RESULTS ANALYSIS FOR TESTING SCENARIOS

Choice of storage configuration has major impact on I/O performance and depends on I/O operation characteristic. Different storage configuration software from GNU/Linux kernel have various I/O performance characteristic. Generally, each I/O operation is performed on other data size, therefore in any tested storage configuration scenario was measured I/O performance for various data block size.

Research results were presented on statement charts for fixed disk numbers storing b-tree filesystem data. To increase readability each statement presents measured I/O performance results from various tested storage space configurations with identical BTRFS storage space with I/O data size from 1KB to 1MB.

Figures 6 and 7 present statements, where I/O operations performance was compared between storage space configurations with disk striping managed by RAID level 0, 5, 6, LVM and directly by BTRFS. The figure 7 presents statement, where I/O operations performance was compared between testing scenario with five active disks managed by software RAID level 0, logical volume configured with striping managed by LVM and directly by BTRFS. On statement charts each LVM or RAID configuration is presented with three fixed chunk size.

The research results presented on statements revealed that choice of chunk size also has impact on I/O performance in any test case, that has other fixed read or written data block size. In storage configuration scenarios where BTRFS directly manages disks, the gap for data write and read speeds has the highest value. Storage configuration scenario with disks managed by LVM has the lowest gap between data read and write speed in each tested scenario. The highest differences between data read and write speed according to configured chunk size in tested scenario have storage configurations with disks managed by software RAID5 and RAID6.

Averaging I/O performance results for each storage configuration scenario (fixed disks manager and chunk size) on all statement charts highlights important conclusions. Scenarios based on LVM with striping provide the best performance in almost all testing cases (except for read and write of 1KB or 2KB data blocks).

Analysis of mean data read and write speeds also show, that the storage configurations base on software RAID level 0 has I/O performance better than BTRFS and software RAID level 5 and RAID level 6 for scenarios with number of disks grater than two. The biggest 35% difference in I/O operation performance was observed in 3 disks scenarios, where mean of data read speeds in all testing case between LVM with 512KB and internal b-tree disk manager and accordingly difference in mean write speed in all testing case was 22 percent.

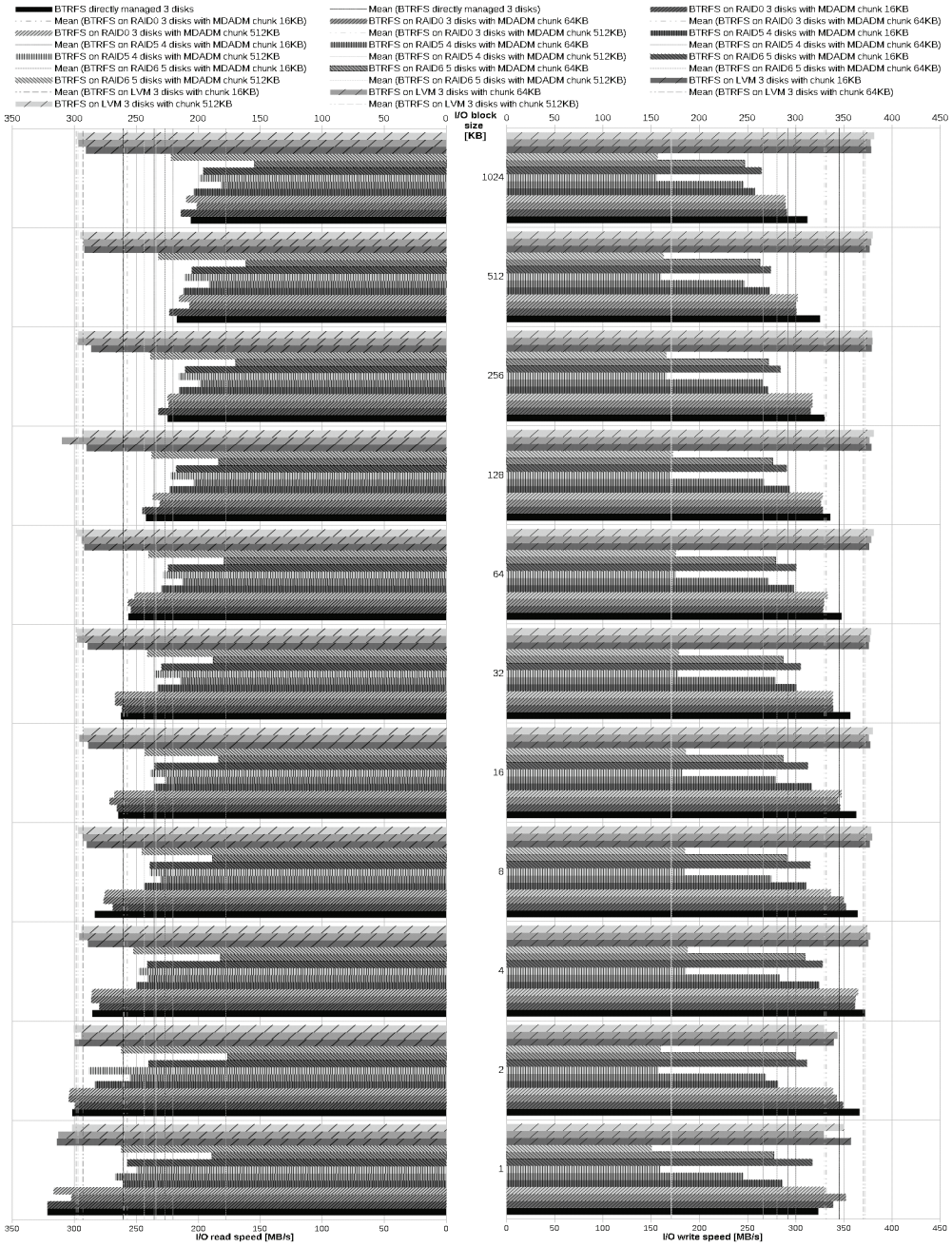

Fig. 6. Statement of I/O performance for storage configuration scenarios with BTRFS and data striping between 3 disks

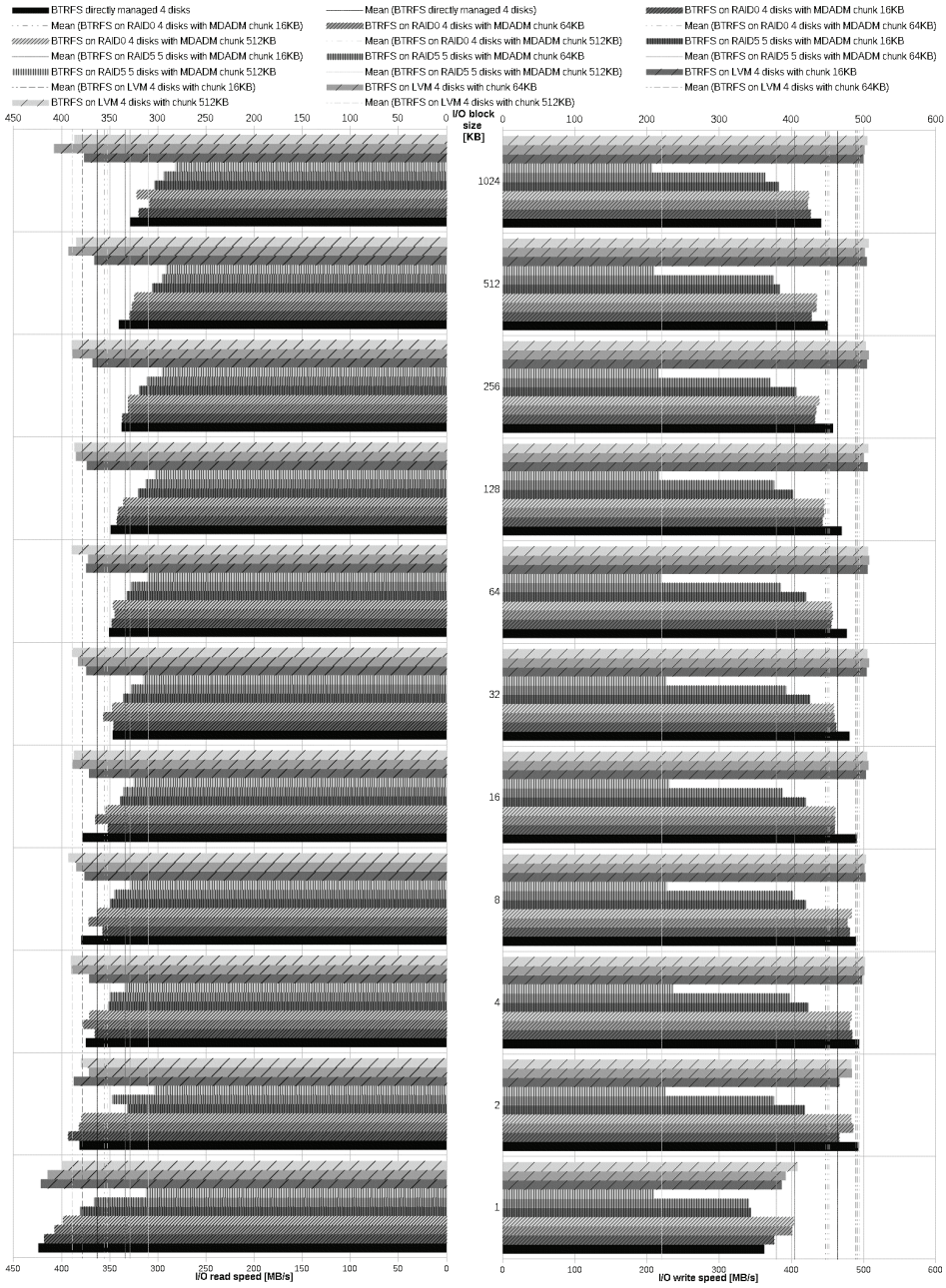

Fig. 7. Statement of I/O performance for storage configuration scenarios with BTRFS and data striping between 4 disks

4. CONCLUSIONS

Data storage space configuration in modern operating has significant impact on I/O operation performance, especially when many disks are used for data striping in storage zone configuration. GNU/Linux like other modern operating system offers disk managers implemented in kernel i.e. LVM or software RAID. Also modern filesystem implemented in Linux kernel like BTRFS includes internal disks managers. Research results of I/O performance for tested storage space configuration clearly show that striping managed by LVM is actually the most efficient multi-disk data storage space configuration. Its average data read and write speed for I/O operation block size from 1 KB up to 1 GB is greater then in storage configurations using other disks managers. Additionally each scenario storage configuration managed with LVM disk striping has lowest gap between data read and write speed in all scenario testing cases.

REFERENCES

[1] TANENBAUM A., *Modern operating systems*, 3rd ed., Prentice Hall, 2007.
[2] SILBERSCHATZ A., GALVIN P.G., GAGNE G., *Operating system concepts*, 9th ed., John Wiley & Sons, 2013.
[3] SOBELL M.G., *Fedora and RedHat Enterprise Linux*, 6th ed., Prentice Hall, 2011.
[4] VUGT S., *Red Hat Enterprise Linux 6 Administration*, John Wiley & Sons, 2013.
[5] NEMETH E., SNYDER G., HEIN T.R., WHALEY B., *UNIX and Linux Administration Handbook*, 4th ed., Pearson Education, 2010.
[6] RODEH O., BACIK J., *The Linux B-tree filesystem.*, Technical report, IBM, 2012.
[7] BACIK J., *Btrfs, The Swiss Army Knife of Storage*, USENIX, 37, 2012.

BIBLIOTEKA INFORMATYKI SZKÓŁ WYŻSZYCH

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

Information Systems Architecture and Technology. Web Information Systems Engineering, Knowledge Discovery and Hybrid Computing, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2011

Information Systems Architecture and Technology. Service Oriented Networked Systems, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2011

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 2011

Information Systems Architecture and Technology. Information as the Intangible Assets and Company Value Source, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2011

Information Systems Architecture and Technology. Web Engineering and High-Performance Computing on Complex Environments, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2012

Information Systems Architecture and Technology. Networks Design and Analysis, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2012

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 2012

Information Systems Architecture and Technology. The Use of IT Models for Organization Management, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2012

Information Systems Architecture and Technology. Intelligent Information Systems, Knowledge Discovery, Big Data and High Pefomance Computing, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2013

Information Systems Architecture and Technology. Networks Architecture and Applications, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2013

Information Systems Architecture and Technology. Knowledge Based Approach to the Design, Control and Decision Support, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2013

Information Systems Architecture and Technology. Models of Decision Making in the Process of Management in a Risky Environment, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2013

Wydawnictwa Politechniki Wrocławskiej
są do nabycia w księgarni
plac Grunwaldzki 13, 50-377 Wrocław
budynek D-1 PWr., tel. 71 320 29 35
Prowadzimy sprzedaż wysyłkową
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-855-6