Advances in Intelligent Systems and Computing 656

Jerzy Świątek
Leszek Borzemski
Zofia Wilimowska *Editors*

Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017

Part II

Springer

Advances in Intelligent Systems and Computing

Volume 656

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Jerzy Świątek · Leszek Borzemski
Zofia Wilimowska
Editors

Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017

Part II

Springer

Editors
Jerzy Świątek
Department of Computer Science,
Faculty of Computer Science
and Management
Wrocław University of Science
and Technology
Wrocław
Poland

Leszek Borzemski
Department of Computer Science,
Faculty of Computer Science
and Management
Wrocław University of Science
and Technology
Wrocław
Poland

Zofia Wilimowska
Department of Management Systems,
Faculty of Computer Science
and Management
Wrocław University of Science
and Technology
Wrocław
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-67228-1 ISBN 978-3-319-67229-8 (eBook)
DOI 10.1007/978-3-319-67229-8

Library of Congress Control Number: 2017952321

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This three volume set of books includes the proceedings of the 2017 38th International Conference on Information Systems Architecture and Technology (ISAT), or ISAT 2017 for short, which will be held on September 17–19, 2017 in Szklarska Poręba, Poland. The conference was organized by the Department of Computer Science and Department of Management Systems, Faculty of Computer Science and Management, Wrocław University of Technology, Poland.

The International Conference on Information Systems Architecture has been organized by the Wrocław University of Technology from the seventies of the last century. The purpose of the ISAT is to discuss a state of the art of information systems concepts and applications as well as architectures and technologies supporting contemporary information systems. The aim is also to consider an impact of knowledge, information, computing and communication technologies on managing the organization scope of functionality as well as on enterprise information systems design, implementation, and maintenance processes taking into account various methodological, technological, and technical aspects. It is also devoted to information systems concepts and applications supporting exchange of goods and services by using different business models and exploiting opportunities offered by Internet-based electronic business and commerce solutions.

ISAT is a forum for specific disciplinary research, as well as for multidisciplinary studies to present original contributions and to discuss different subjects of today's information systems planning, designing, development, and implementation. The event is addressed to the scientific community, people involved in variety of topics related to information, management, computer and communication systems, and people involved in the development of business information systems and business computer applications.

This year, we received 180 papers. The papers included in the three proceedings volumes published by Springer have been subject to a thoroughgoing review process by highly qualified peer reviewers. At least two members of Program Committee or Board of Reviewers reviewed each paper. The final acceptance rate was 56%. Program Chairs selected 101 best papers for oral presentation and

publication in the 38th International Conference on Information Systems Architecture and Technology 2017 proceedings.

The papers have been grouped into three volumes:

Part I—discoursing about topics including, but not limited to, Artificial Intelligence Methods, Knowledge Discovery and Data Mining, Big Data, Knowledge Discovery and Data Mining, Knowledge Based Management, Internet of Things, Cloud Computing and High Performance Computing, Distributed Computer Systems, Content Delivery Networks, Service Oriented Computing, and E-Business Systems, Web Design, Mobile and Multimedia Systems.

Part II—addressing topics including, but not limited to, System Modelling for Control, Recognition and Decision Support, Mathematical Modeling in Computer System Design, Service Oriented Systems and Cloud Computing and Complex process modelling.

Part III—dealing with topics including, but not limited to, Modeling of Manufacturing Processes, Modeling an Investment Decision Process, Management of Innovation, Management of Organization.

We would like to thank the Program Committee and external reviewers, essential for reviewing the papers to ensure a high standard of the ISAT 2017 conference and the proceedings. We thank the authors, presenters, and participants of ISAT 2017; without them, the conference could not have taken place. Finally, we thank the organizing team for the efforts in bringing the conference to a successful scientific event.

We devote ISAT 2017 to our friend and former ISAT Chair, Professor Adam Grzech.

September 2017

Leszek Borzemski
Jerzy Świątek
Zofia Wilimowska

In Memory of Our Friend and Past ISAT Chair

Professor Adam Grzech

November 2016

Professor DSc. Adam Grzech was born in 1954 in Dębica (Poland). He was graduated at the Wroclaw University of Technology 1977—M.Sc. Electronic Engineering. He did his PhD at the Institute of Technical Cybernetics in 1979 and D.Sc. in technical sciences in the field of computer science in 1989 and received the title of full Professor in 2003 from Wroclaw University of Technology.

He was an Assistant Professor at the Institute of Technical Cybernetics (1979–1982), an Assistant Professor at the Institute of Control and Systems Engineering (1982–1989), Associate Professor at the Institute of Control and Systems Engineering (1989–1993), a Professor at the Institute for Technical Informatics (1993–2006), and a Full Professor at the Institute of Computer Science, Faculty of Computer Science and Management at Wroclaw University of Technology (since 2006).

He was an author and co-author of over 350 published research works. His areas of research were as follows: analysis, modeling, and design information and communication systems and networks. Since 2002, he was a Chief of Telecommunication Department. He was a leader of Scientific School on Information and Communication Systems and Networks. He was a promoter of the 10 completed Ph.D. theses.

During the years 1991–1993 and 1999–2002, he was a Director of the Institute of Control and Systems Engineering. Since 2002, he was a delegate of the Rector for Information Technology, and then during the years 2003–2005, he was a Vice President for Development at the Wroclaw University of Technology and in 2002–2008 a member of the Senate of the Wroclaw University of Technology. During the years 2005–2012, he was a Vice Dean of Computer Science and Management Faculty.

He was active in the work at the national, international committees of conferences and scientific journals. He was Co-chair of Information Systems Architecture and Technology (ISAT) and Systems Science International Conferences. Since 1982, he served as Scientific Secretary, and since 2006 Editor-in-Chief of the Science Systems journal published quarterly.

Professor Adam Grzech was a member of the Wroclaw Scientific Society, the Polish Informatics Society, the Council of Information Technology, the Committee on Informatics of the Polish Academy of Sciences, and Technical Committee TC6 (Communication Systems) IFIP.

Leszek Borzemski
Jerzy Świątek
Zofia Wilimowska

ISAT 2017 Szklarska Poręba, September 17–19, 2017

ISAT 2017 Conference Organization

General Chair

Leszek Borzemski, Poland

Program Co-chairs

Leszek Borzemski, Poland
Jerzy Świątek, Poland
Zofia Wilimowska, Poland

Local Organizing Committee

Leszek Borzemski (Chair)
Zofia Wilimowska (Co-chair)
Jerzy Świątek (Co-chair)
Mariusz Fraś (Conference Secretary, Website Support)
Arkadiusz Górski (Technical Editor)
Anna Kamińska (Technical Secretary)
Ziemowit Nowak (Technical Support)
Kamil Nowak (Website Coordinator)
Danuta Seretna-Sałamaj (Technical Secretary)

International Program Committee

Leszek Borzemski, Poland (Chair)
Jerzy Świątek, Poland (Co-chair)
Zofia Wilimowska, Poland (Co-chair)
Witold Abramowicz, Poland

Dhiya Al-Jumeily, UK
Iosif Androulidakis, Greece
Patricia Anthony, New Zealand
Zbigniew Banaszak, Poland
Elena N. Benderskaya, Russia
Janos Botzheim, Japan
Djallel E. Boubiche, Algeria
Patrice Boursier, France
Anna Burduk, Poland
Andrii Buriachenko, Ukraine
Udo Buscher, Germany
Wojciech Cellary, Poland
Haruna Chiroma, Malaysia
Edward Chlebus, Poland
Gloria Cerasela Crisan, Romania
Marilia Curado, Portugal
Czesław Daniłowicz, Poland
Zhaohong Deng, China
Małgorzata Dolińska, Poland
Milan Edl, Czech Republic
El-Sayed M. El-Alfy, Saudi Arabia
Peter Frankovsky, Slovakia
Naoki Fukuta, Japan
Bogdan Gabryś, Poland
Piotr Gawkowski, Poland
Manuel Graña, Spain
Wiesław M. Grudzewski, Poland
Katsuhiro Honda, Japan
Marian Hopej, Poland
Zbigniew Huzar, Poland
Natthakan Iam-On, Thailand
Biju Issac, UK
Arun Iyengar, USA
Jürgen Jasperneite, Germany
Janusz Kacprzyk, Poland
Henryk Kaproń, Poland
Yury Y. Korolev, Belarus
Yannis L. Karnavas, Greece
Ryszard Knosala, Poland
Zdzisław Kowalczuk, Poland
Lumír Kulhanek, Czech Republic
Binod Kumar, India
Jan Kwiatkowski, Poland
Antonio Latorre, Spain
Radim Lenort, Czech Republic

Gang Li, Australia
José M. Merigó Lindahl, Chile
Jose M. Luna, Spain
Emilio Luque, Spain
Sofian Maabout, France
Lech Madeyski, Poland
Zygmunt Mazur, Poland
Elżbieta Mączyńska, Poland
Pedro Medeiros, Portugal
Toshiro Minami, Japan
Marian Molasy, Poland
Zbigniew Nahorski, Poland
Kazumi Nakamatsu, Japan
Peter Nielsen, Denmark
Tadashi Nomoto, Japan
Cezary Orłowski, Poland
Sandeep Pachpande, India
Michele Pagano, Italy
George A. Papakostas, Greece
Zdzisław Papir, Poland
Marek Pawlak, Poland
Jan Platoš, Czech Republic
Tomasz Popławski, Poland
Edward Radosinski, Poland
Wolfgang Renz, Germany
Dolores I. Rexachs, Spain
José S. Reyes, Spain
Leszek Rutkowski, Poland
Sebastian Saniuk, Poland
Joanna Santiago, Portugal
Habib Shah, Malaysia
J.N. Shah, India
Jeng Shyang, Taiwan
Anna Sikora, Spain
Marcin Sikorski, Poland
Małgorzata Sterna, Poland
Janusz Stokłosa, Poland
Remo Suppi, Spain
Edward Szczerbicki, Australia
Eugeniusz Toczyłowski, Poland
Elpida Tzafestas, Greece
José R. Villar, Spain
Bay Vo, Vietnam
Hongzhi Wang, China
Leon S.I. Wang, Taiwan

Junzo Watada, Japan
Eduardo A. Durazo Watanabe, India
Jan Werewka, Poland
Thomas Wielicki, USA
Bernd Wolfinger, Germany
Józef Woźniak, Poland
Roman Wyrzykowski, Poland
Yue Xiao-Guang, Hong Kong
Jaroslav Zendulka, Czech Republic
Bernard Ženko, Slovenia

ISAT 2017 Reviewers

Małgorzata Adamska, Poland
Patricia Anthony, New Zealand
Katarzyna Antosz, Poland
Zbigniew Antoni Banaszak, Poland
Jan Betta, Poland
Grzegorz Bocewicz, Poland
Leszek Borzemski, Poland
Janos Botzheim, Hungary
Djallel Eddine Boubiche, Algeria
Krzysztof Brzostowski, Poland
Anna Burduk, Poland
Udo Buscher, Germany
Wojciech Cellary, Poland
Haruna Chiroma, Malaysia
Witold Chmielarz, Poland
Grzegorz Chodak, Poland
Kazimierz Choroś, Poland
Piotr Chwastyk, Poland
Gloria Cerasela Crisan, Romania
Anna Czarnecka, Poland
Mariusz Czekała, Poland
Grzegorz Debita, Poland
Jarosław Drapała, Poland
Tadeusz Dudycz, Poland
Fic Maria, Poland
Mariusz Fraś, Poland
Naoki Fukuta, Japan
Joanna Furman, Poland
Piotr Gawkowski, Poland
Krzysztof Goczyła, Poland
Arkadiusz Gola, Poland

Arkadiusz Górski, Poland
Jerzy Grobelny, Poland
Joanna Helman, Poland
Bogumila Hnatkowska, Poland
Katsuhiro Honda, Japan
Grażyna Hołodnik-Janczura, Poland
Zbigniew Huzar, Poland
Artur Hłobaż, Poland
Biju Issac, UK
Katarzyna Jach, Poland
Małgorzata Jasiulewicz-Kaczmarek, Poland
Jerzy Józefczyk, Poland
Ireneusz Jóźwiak, Poland
Magdalena Jurczyk-Bunkowska, Poland
Krzysztof Juszczyszyn, Poland
Robert Kamiński, Poland
Yannis Karnavas, Greece
Włodzimierz Kasprzak, Poland
Grzegorz Kołaczek, Poland
Mariusz Kołosowski, Poland
Jacek Korniak, Poland
Zdzisław Kowalczuk, Poland
Damian Krenczyk, Poland
Lumír Kulhánek, Czech Republic
Binod Kumar, India
Jan Kwiatkowski, Poland
Sławomir Kłos, Poland
Antonio LaTorre, Spain
Zbigniew Lipiński, Poland
Wojciech Lorkiewicz, Poland
Andrzej Loska, Poland
Sofian Maabout, France
Lech Madeyski, Poland
Rafał Michalski, Poland
Miroslava Mlkva, Slovak Republic
Marian Molasy, Poland
Zbigniew Nahorski, Poland
Peter Nielsen, Denmark
Cezary Orłowski, Poland
Donat Orski, Poland
Michele Pagano, Italy
Agnieszka Parkitna, Poland
Iwona Pisz, Poland
Jan Platoš, Czech Republic
Grzegorz Popek, Poland

Tomasz Popławski, Poland
Sławomir Przyłucki, Poland
Dolores Rexachs, Spain
Maria Rosienkiewicz, Poland
Stefano Rovetta, Italy
Jacek, Piotr Rudnicki, Poland
Dariusz Rzońca, Poland
Anna Saniuk, Poland
Joanna Santiago, Portugal
José Santos, Spain
Danuta Seretna-Sałamaj, Poland
Anna Sikora, Spain
Marcin Sikorski, Poland
Dorota Stadnicka, Poland
Malgorzata Sterna, Poland
Janusz Stokłosa, Poland
Grażyna Suchacka, Poland
Remo Suppi, Spain
Joanna Szczepańska, Poland
Edward Szczerbicki, Australia
Jerzy Świątek, Poland
Błażej Święcicki, Poland
Kamila Urbańska, Poland
José R. Villar, Spain
Bay Vo, Vietnam
Anna Walaszek-Babiszewska, Poland
Leon Shyue-Liang Wang, Taiwan
Krzysztof Waśko, Poland
Jan Werewka, Poland
Zofia Wilimowska, Poland
Bernd Wolfinger, Germany
Jozef Woźniak, Poland
Ryszard Wyczółkowski, Poland
Jacek Zabawa, Poland
Krzysztof Zatwarnicki, Poland
Jaroslav Zendulka, Czech Republic
Maciej Zięba, Poland

ISAT 2017 Special Session

Modeling of Manufacturing Processes
Anna Burduk (Chair), Poland

Contents

System Modelling for Control, Recognition and Decision Support

Use of the Modified EPSILON Decomposition for the LTI Models Reduction

Damian Raczyński(✉) and Włodzimierz Stanisławski

University of Applied Sciences in Nysa, ul. Chodowieckiego 4,
48-300 Nysa, Poland
{damian.raczynski,
wlodzimierz.stanislawski}@pwsz.nysa.pl

Abstract. For the purposes of representing the characteristics of physical systems, the LTI (Linear Time Invariant) models are often used. Even in the case where the physical system is described by a set of nonlinear differential equations, the linearization process is able to determine a set of linear differential equations e.g. at the working point. The main problem that arises when designing the system model is the order of the resulting matrices. The model order reduction of the LTI models is a common operation which led to obtain a model of smaller size. The large system size issue often make impossible to perform real-time simulation or designate the control system - for example, determining the LQG control system is possible only for relatively small models. In order to achieve high accuracy of the reduced model with the original model, it is necessary to perform a reduction in the specified range of frequency. The problem arises when the resulting model has to match both - low and high frequencies. In addition, models that describe the slow and fast phenomenon are difficult to numerical analyze. The authors propose the use of a methodology, which bases on separating the model into two parts - the fast part and the slow part. The modified Epsilon decomposition is proposed do achieve this goal. The obtained results confirm that the presented methodology is correct.

Keywords: Epsilon decomposition · Model order reduction · Frequency Weighted Model Reduction · Stiff models reduction

1 Introduction

The paper presents a method of reducing the LTI models expressed by the set of Eq. (1).

$$\sum : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \tag{1}$$

where: **A** - state matrix, **B** - input-to-state matrix, **C** - state-to-output matrix, **D** – Feed through matrix.

The LTI model of a system can be determined using, for example, the finite element method. The finite element method is very popular today, but the discretization of the

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_1

continuum may lead to a complex system of ordinary differential equations. Analyzing the properties of the obtained model or determining its control system may be impossible. In such situations, the model order reduction should be carried out.

The Frequency Weighted Model Reduction (FWMR) method [1], used in the examples in this article, was developed to reduce the order of models with minimizing the approximation error expressed by the formula (2).

$$\|\mathbf{G_{WO}}(\mathbf{s})(\mathbf{G_O}(\mathbf{s}) - \mathbf{G_r}(\mathbf{s}))\mathbf{G_{WI}}(\mathbf{s})\|_\infty \tag{2}$$

Where **G**'s denote transfer matrices of: $\boldsymbol{G_{o(S)}}$ – the original model, $\boldsymbol{G_r(s)}$ – the reduced model, $\boldsymbol{G_{WO}(s)}$ – the input filter, $\boldsymbol{G_{WI}(s)}$ – the output filter.

The presence of input and output filters causes the reduced model more accurately approximate the frequencies for which the output filter and/or input filter have higher singular values. This method was introduced in the works of D. Enns. Modification of the Enns method, which allows to maintain the stability of the reduced model using both filters, is proposed in [2].

The LTI models can describe both fast and slow processes running simultaneously within the modeled physical system. A model in which both fast and slow modes occur is called as a stiff model. The model stiffness coefficient can be determined from the formula (3).

$$\eta(\mathbf{A}) = \frac{Max\{|\mathrm{Re}(\lambda_i)|\}}{Min\{|\mathrm{Re}(\lambda_i)| \neq 0\}} \tag{3}$$

It determines the ratio of the absolute value of the real part of the largest eigenvalue of the matrix A to the smallest. Models with significant stiffness are difficult to analyze by numerical methods. The solution to this problem may be division of the system into two parts - one for fast processes and the other for slow processes. The method of model decomposition in which the fast part and the slow part are already separated is described in detail in the literature [3–5]. The authors would like to point out that the literature known to them does not include a method of decomposing a model into a fast and slow parts if both types of equations are mixed together.

By using the finite element method to determine the model of a physical object, usually the equations describing both types of processes are mixed together. It is necessary to renumber state variables in such a way that equations describing fast dynamics fall into one part of the model, while the equations associated with the slow processes in the second part. The purpose of this article is to develop a general methodology that will allow to divide the model into two parts (slow and fast) without knowing the physical meaning of the individual equations. The epsilon decomposition will be used for this purpose. The designated parts will be subject to independent reductions using the FWMR method.

2 The Epsilon Decomposition

Epsilon decomposition uses the fact that complex LTI models have a weak connection (or lack of connection) for most state variables. Usually, object dynamics are largely dominated by strongly interrelated state variables. It is crucial to identify groups of strongly interrelated elements. The Epsilon decomposition designates the subsystems which are strongly interconnected internally and weakly connected between each other. The ε parameter determining which of the connections between subsystems is insignificant. It is the number whose value is greater than 0 (the value is arbitrarily determined). Based on the ε value, the position of state variables is changed so that the strongly interrelated elements fall into the diagonal blocks of the state matrix **A** (each group of strongly connected state variables in a separate block). Blocks beyond the main diagonal, after performing permutation of rows and columns, will contain small values. The epsilon decomposition of the **A** matrix can be written as the sum of the block-diagonal matrix $\mathbf{A}_0$ and matrix $\boldsymbol{\varepsilon}\overline{\mathbf{A}}_1$ which contains the connections between the separated blocks [6–9].

$$\mathbf{A} = \mathbf{A_0} + \boldsymbol{\varepsilon}\overline{\mathbf{A}}_1 \tag{4}$$

The multi-level epsilon decomposition is based on performing an analogous division with respect to the designated subsystems Fig. 1.

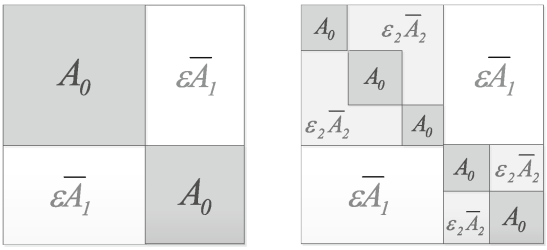


Fig. 1. A sample structure of state matrix after the one level (left side) and two levels (right side) epsilon decomposition

2.1 The Modified Epsilon Decomposition Algorithm

The first operation performed by the modified Epsilon decomposition algorithm is normalization of all elements of matrix **A** by computing absolute value of each element divided by the maximal value of the matrix. After this operation, the matrix **A_norm** will be determined. In the next step, the algorithm zeroes all elements from the matrix **A_norm**, which are smaller than the epsilon value. All elements of the matrix **A_norm** larger than the ε are set to 1. The obtained binary matrix **A_bin** defines the strong relationship of state variables. Using the CONNECTED COMPONENTS algorithm, the connections between the rows of the matrix **A_bin** are searched. The purpose of the modified Epsilon decomposition is to find the strong connections between the differential equations describing the physical phenomena. The matrix **A_bin** is treated as a

graph, where the vertices of the graph are rows. Elements equal 1 define the edges of the graph - if the n'th row contain 1 in the m'th column, this means that there is an edge from n'th to m'th vertex of the graph. The algorithm, in the form of MATLAB code, is presented below.

```
function [perm nodes]= epsdec_mod(A, e)
  A_bin=epsilon(A,e);
  nodes=connected_components(A_bin);
  perm=permutVector(nodes);
end

function A_bin=epsilon(A, epsilon)
  m=max(max(abs(A)));
  A_norm=abs(1/m*A);
  for i=1:size(A_norm,1)
    for j=1:size(A_norm,2)
      if(A_norm(i,j)<=epsilon)
        A_norm(i,j)=0;
      else
        A_norm(i,j)=1;
      end
    end
  end
  A_bin=A_norm;
end
```

The function `epsdec_mod` is a modified Epsilon decomposition algorithm. The function parameters are the state matrix of original model (`A`) and the ε value (`e`). Based on the value of the ε parameter, using the `epsilon` function a connections graph is created (the binary matrix of connections). The values returned by the function are the permutation vector and the collection identifying the connected components of the created graph. The permutation vector determines the order of state variables in such a way that the strongly connected state equations are placed in the same blocks on the main diagonal of the state matrix (`A`). Collection `nodes` is a set of vectors that contain indices of the state variables belonging to the common connected component within the created graph.

Comparing the presented algorithm to the generally available EPSDEC code [10], the main difference occurs in the matrix elements normalization. The EPSDEC normalizes every row by row absolute maximal value. For the considered model it prevents the separation of the fast and slow parts. In the presented algorithm, all matrix elements are normalized by the matrix absolute maximal value.

3 Experiment Data

As an exemplary model we choose the model of one-phase zone of evaporating tubes system in the BP-1150 steam boiler [11]. The model of the evaporating tubes system describes the slow heat processes (thermal conductivity along the tubes and connecting fins), mass transfer processes and enthalpy change of the water-steam mixture along the screen tubes, as well as very fast dynamic processes - movement of pressure along the tubes (sound processes). In order to obtain a lumped parameter model, the cumulative evaporator tube is divided along its length into a number of finite elements with dimensions small enough to assume that the individual element is a dynamic system with the lumped parameters (described by the system of ordinary differential equations). Physical meanings of the model variables are as follows: M_k – mass flow kg/s, h_k – enthalpy of the water kJ/kg, P_k – pressure MPa, q_k – heat flux kW/m, Z_k – position m, $\Theta_{t,k}$ - temperatures of the tube wall and connecting fin K (Fig. 2).

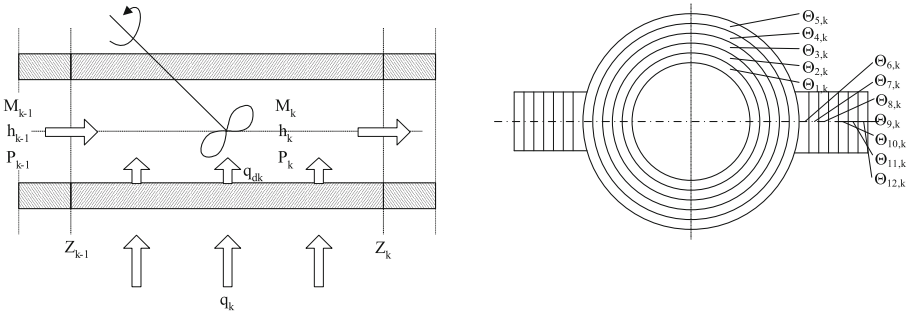


Fig. 2. A structure of single finite element of evaporator tube [11]

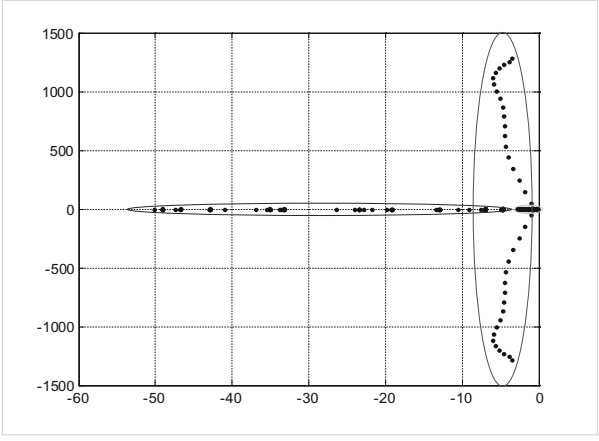


Fig. 3. Eigenvalues of one-phase zone of evaporating tubes system in the BP-1150 steam boiler

The linearized model is characterized by high stiffness ($\eta(\mathbf{A}) > 1000$). With such a large difference in the values of the eigenvalues (Fig. 3), the dynamic processes associated with the displacement of pressure changes along the tube are practically independent of the other thermodynamic processes. For this reason, two weakly interconnected sub-systems can be distinguished in the evaporator tube model:

- a model describing rapid sound phenomena,
- a model describing slow heat and thermodynamic phenomena.

4 Model Decomposition

In the analyzed case it is desirable to group into one subsystem the state equations associated with high eigenvalues (fast dynamic processes - displacement of pressure and mass flow along the tubes), while in the second subsystem – state equations associated with movement of enthalpy changes along the tubes and changes in temperature of the tubes and the connecting fins (slow dynamic processes associated with small eigenvalues).

The study was carried out on the LTI model of the one-phase zone consisting of 40 finite elements. Each finite element is related to 15 equations describing the physical phenomena occurring in the tubes. For the single k'th finite element the state variables form the vector presented in the formula (5) [11].

$$\mathbf{X}_k = \left[M_k P_k h_k \Theta_{1,k} \ldots \Theta_{12,k}\right] \tag{5}$$

The analyzed model contain 40 * 15 = 600 differential equations, which after linearization can be written in the state space: $\frac{dX}{dt} = AX + Bu, Y = CX$, where: X – state vector [600, 1], A – state matrix [600, 600], B – input matrix [600, 4], C – output matrix [3, 600], u – input vector [1, 4].

Using the original Epsilon decomposition algorithm to decompose the model, it does not produce the desirable results. For example, performing a decomposition for $\varepsilon = 0.2$ resulted in the division of the model into two parts. One decomposed system consists of elements with indexes equal: 541 556 571 573 586 588. The second system contains the remaining state equations. Changing the ε coefficient from 0 to 1 with a step 0.01 did not produce the desired results.

The modified version of Epsilon decomposition produces correct results. The algorithm was tested with epsilon values varies from 0.01 to 0.9 (with 0.01 steps). Each time the process separates two subsystems. One decomposed system consists of elements with indices equal: 2 17 32 … 587. These rows are related to the state variables P_k (displacement of pressure). The second decomposed system consists of remain elements (mass flow along the tubes and slow dynamic processes associated with small eigenvalues).

In the next step, the second system was redecomposed using the modified Epsilon algorithm. For the value of the ε coefficient from 0.04 to 0.05, the decomposition

divides the model into two parts. First decomposed system consists of elements with indices equal: 1 15 29 … 547. These rows are related to the state variables M_k (mass flow along the tubes). The second decomposed system consist of equations related to the slow dynamic processes.

5 Decomposed Models Verification

For simplicity of the presented method, we assumed that the original model is divided into two parts (the fast and the slow part). The fast part of the model is related to the Eqs. 1 and 2 of each finite element (80 equations), indices of these equations were determined in the modified Epsilon decomposition process. The slow part of the model consists of the remain equations (520 equations).

The LTI model of the one-phase zone has 4 inputs and 3 outputs (Fig. 4). Physical meanings of the input and output variables are as follows: $q^{\sim}$ – relative value of the heat flux, P_{out} – the outlet pressure of the one-phase zone MPa, h_{in} – the inlet water enthalpy kJ/kg, M_{in} – the inlet water flow kg/s, P_{in} – the inlet pressure of the one-phase zone MPa, h_{out} – the outlet water enthalpy kJ/kg, M_{out} – the outlet water flow kg/s.

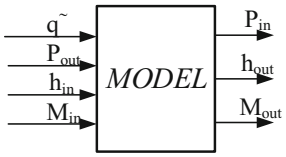


Fig. 4. Inputs and outputs of the model of one-phase zone.

The sub-models connections in the evaporation tubes model are presented in the Fig. 5.

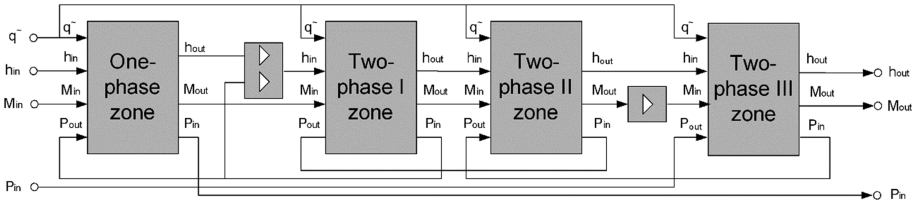


Fig. 5. The structure of the evaporation tubes model

If the presented methodology is correct then the frequency characteristics of the decomposed subsystems should satisfy the following conditions:

- fast subsystems should not differ much from the characteristics of the original model for high frequencies and fast input-output tracks,

- slow subsystem should approximate well the frequency characteristics of the original model for slow input-output tracks for slow frequencies.

Figures 6, 7, 8 and 9 compare the frequency characteristics of the selected input-output tracks of the decomposed models with respect to the original model.

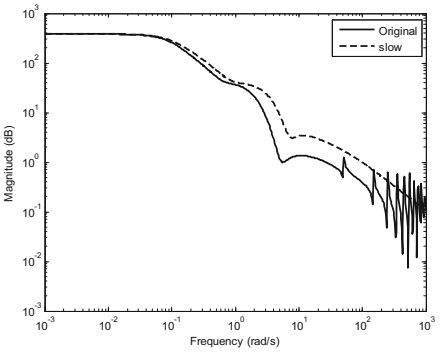


Fig. 6. Frequency response - graph of amplitude for the track $q_{in} \rightarrow P_{out}$

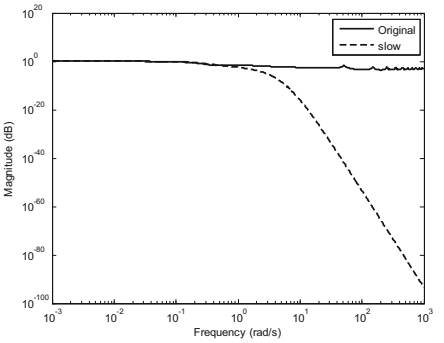


Fig. 7. Frequency response - graph of amplitude for the track $P_{out} \rightarrow P_{in}$

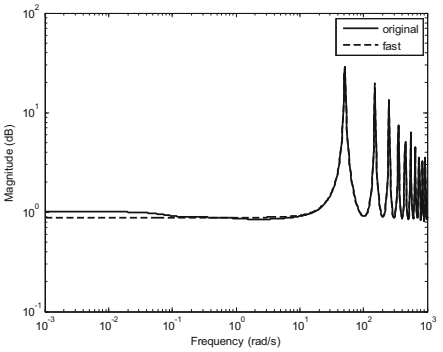


Fig. 8. Frequency response - graph of amplitude for the track $h_{in} \rightarrow h_{out}$

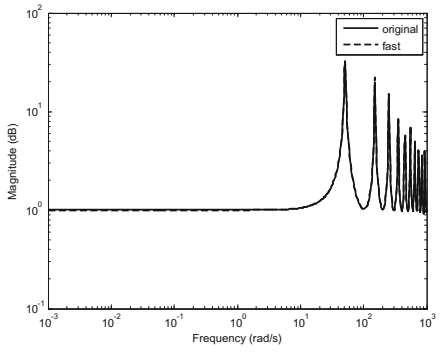


Fig. 9. Frequency response - graph of amplitude for the track $h_{in} \rightarrow h_{out}$

Figures 6, 7, 8 and 9 demonstrate the correctness of the proposed methodology. The proposed modification of the Epsilon algorithm correctly decomposes the model into parts related to fast and slow processes.

6 Decomposed Models Reduction

The original model, after permutation of state variables takes the form (6):

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 + \mathbf{B}_1\mathbf{u}\\ \dot{\mathbf{x}}_2 &= \mathbf{A}_{22}\mathbf{x}_2 + \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{B}_2\mathbf{u}\\ \mathbf{y} &= [\mathbf{y}_1\mathbf{y}_2] = [\mathbf{C}_1\mathbf{C}_2]\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}\end{aligned} \tag{6}$$

where: $\mathbf{x_1}$ – The first part of the state vector of size n_1, $\mathbf{x_2}$ – The second part of the state vector of size n_2, $\mathbf{A_{11}}$ – state matrix of subsystem 1 of size $n_1 \times n_2$, $\mathbf{A_{22}}$ – state matrix of subsystem 2 of size $n_2 \times n_2$, $\mathbf{A_{12}}$, $\mathbf{A_{21}}$ – matrices of interfaces between subsystems, determining the impact of one subsystem on another of size respectively $n_1 \times n_2$, $n_2 \times n_1$, $\mathbf{C_1}$, $\mathbf{C_2}$ – subsystems output matrices, of size respectively $l \times n_1$, $l \times n_2$, $\mathbf{B_1}$, $\mathbf{B_2}$ – subsystems input matrices of sizes respectively $n_1 \times k$, $n_2 \times k$, $n = n_1 + n_2$.

Both decomposed subsystems are subject to independent reduction, without considering connecting them interfaces. As a result, reduced subsystems expressed by formula (7) are obtained.

$$\begin{aligned}\tilde{\dot{\mathbf{x}}}_1 &= \tilde{\mathbf{A}}_{11}\tilde{\mathbf{x}}_1 + \tilde{\mathbf{B}}_1\mathbf{u}_1, \quad \tilde{\mathbf{y}}_1 = \tilde{\mathbf{C}}_1\tilde{\mathbf{x}}_1\\ \tilde{\dot{\mathbf{x}}}_2 &= \tilde{\mathbf{A}}_{22}\tilde{\mathbf{x}}_2 + \tilde{\mathbf{B}}_2\mathbf{u}_2, \quad \tilde{\mathbf{y}}_2 = \tilde{\mathbf{C}}_2\tilde{\mathbf{x}}_2\end{aligned} \tag{7}$$

where:

$$\begin{array}{llllll}\tilde{\mathbf{x}}_1 & = \mathbf{T}_1\mathbf{x}_1, & \mathbf{x}_1 & = \mathbf{L}_1\tilde{\mathbf{x}}_1, & \tilde{\mathbf{x}}_2 & = \mathbf{T}_2\mathbf{x}_2, \quad \mathbf{x}_2 = \mathbf{L}_2\tilde{\mathbf{x}}_2,\\ \tilde{\mathbf{A}}_{11} & = \mathbf{T}_1\mathbf{A}_{11}\mathbf{L}_1, & \tilde{\mathbf{B}}_1 & = \mathbf{T}_1\mathbf{B}_1, & \tilde{\mathbf{C}}_1 & = \mathbf{C}_1\mathbf{L}_1,\\ \tilde{\mathbf{A}}_{22} & = \mathbf{T}_2\mathbf{A}_{22}\mathbf{L}_2, & \tilde{\mathbf{B}}_2 & = \mathbf{T}_2\mathbf{B}_2, & \tilde{\mathbf{C}}_2 & = \mathbf{C}_2\mathbf{L}_2.\end{array}$$

The matrices $\mathbf{T_1}$, $\mathbf{L_1}$ and $\mathbf{T_2}$, $\mathbf{L_2}$ are obtained in the reduction process of both subsystems (transformation matrices generated by FWMR reduction algorithm). The transformation matrices sizes are as follows: $\mathbf{T_1}$ [$k_1 \times n_1$], $\mathbf{L_1}$ [$n_1 \times k_1$], $\mathbf{T_2}$ [$k_2 \times n_2$], $\mathbf{L_2}$ [$n_2 \times k_2$]. After considering the dependencies in the original model, substituting $\mathbf{x}_1 = \mathbf{L}_1\tilde{\mathbf{x}}_1$ and $\mathbf{x}_2 = \mathbf{L}_2\tilde{\mathbf{x}}_2$ to the original model equations, the formula (8) is obtained.

$$\begin{aligned}\mathbf{L}_1\tilde{\dot{\mathbf{x}}}_1 &= \mathbf{A}_{11}\mathbf{L}_1\tilde{\mathbf{x}}_1 + \mathbf{A}_{12}\mathbf{L}_2\tilde{\mathbf{x}}_2 + \mathbf{B}_1\mathbf{u}\\ \mathbf{L}_2\tilde{\dot{\mathbf{x}}}_2 &= \mathbf{A}_{22}\mathbf{L}_2\tilde{\mathbf{x}}_2 + \mathbf{A}_{21}\mathbf{L}_1\tilde{\mathbf{x}}_1 + \mathbf{B}_2\mathbf{u}\end{aligned} \tag{8}$$

After left multiplying (8) by $\mathbf{T_1}$ and $\mathbf{T_2}$, respectively, and after taking into account that $\mathbf{T\ L = 1}$, we obtain (9).

$$\begin{aligned}\tilde{\dot{\mathbf{x}}}_1 &= \tilde{\mathbf{A}}_{11}\tilde{\mathbf{x}}_1 + \mathbf{T}_1\mathbf{A}_{12}\mathbf{L}_2\tilde{\mathbf{x}}_2 + \mathbf{T}_1\mathbf{B}_1\mathbf{u}, \quad \tilde{\mathbf{y}}_1 = \mathbf{C}_1\mathbf{L}_1\tilde{\mathbf{x}}_1,\\ \tilde{\dot{\mathbf{x}}}_2 &= \tilde{\mathbf{A}}_{22}\tilde{\mathbf{x}}_2 + \mathbf{T}_2\mathbf{A}_{21}\mathbf{L}_1\tilde{\mathbf{x}}_1 + \mathbf{T}_2\mathbf{B}_2\mathbf{u}, \quad \tilde{\mathbf{y}}_2 = \mathbf{C}_2\mathbf{L}_2\tilde{\mathbf{x}}_2\end{aligned} \tag{9}$$

The matrices structure of the reduced model is shown in Fig. 10.

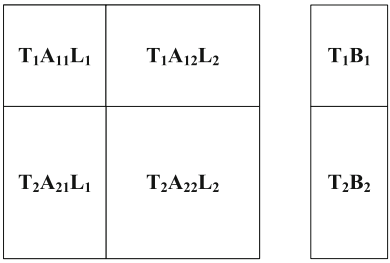


Fig. 10. Structure of matrices **A** and **C** of reduced model

7 Reduced Model Verification

As a result of decomposition of the original model into the fast part (large eigenvalues) and the slow part (small eigenvalues), two obtained subsystems are separated from each other significantly. For some input and output tracks, and a suitable frequency range, they are almost independent (what can be noticed by analyzing the Figs. 6, 7, 8 and 9). Both designated subsystems have been reduced with use of the eAMOR MATLAB application [12, 13]. Table 1 shows the reduction parameters.

Table 1. The FWMR reduction parameters for the slow and fast part (k – rank of whole reduced model, k fast – rank of the fast reduced part of model, k slow – rank of the slow reduced slow of the model, high frequency range – frequency range for determining a reduction error for a fast part, low frequency range – frequency range for determining a reduction error for a slow part)

k	k fast	k slow	High frequency range	Low frequency range
10	5	5	0.01–20 rad/s	0.01–1000 rad/s
20	10	10		
190	110	80		

Figures 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22 show the frequency characteristics of the reduced models (reduced and connected parts fast and slow).

The average square relative approximation error of the reduced models, given by Eq. (10)

$$\rho = \sqrt{\frac{1}{pmN}\sum_{i=1}^{N}\sum_{x=1}^{p}\sum_{y=1}^{m}\left(\frac{\left|G_{(x,y)}(j\omega_i) - G_{r(x,y)}(j\omega_i)\right|}{\left|G_{(x,y)}(j\omega_i)\right|}\right)^2} \tag{10}$$

where: p, m - numbers of inputs and outputs of the model, N - number of approximation points in frequency domain ($N = 231$) for the frequency scope $\langle 10^{-3} - 1000\,\mathrm{rad/s}\rangle$ is presented in the Table 2.

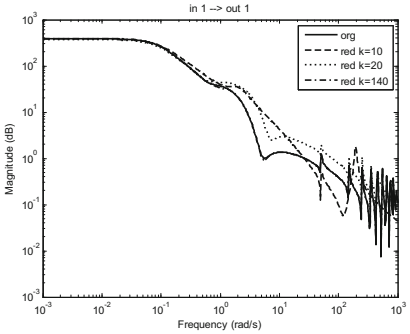


Fig. 11. Frequency response - graph of amplitude for the track $q_{in} \rightarrow P_{in}$

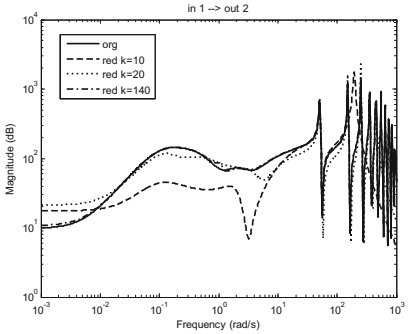


Fig. 12. Frequency response - graph of amplitude for the track $q_{in} \rightarrow h_{out}$

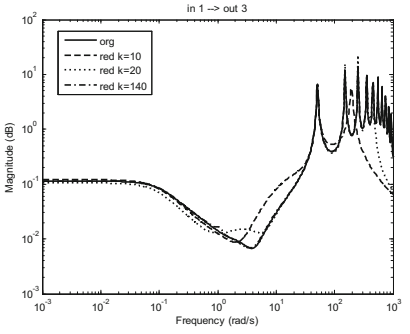


Fig. 13. Frequency response - graph of amplitude for the track $q_{in} \rightarrow M_{out}$

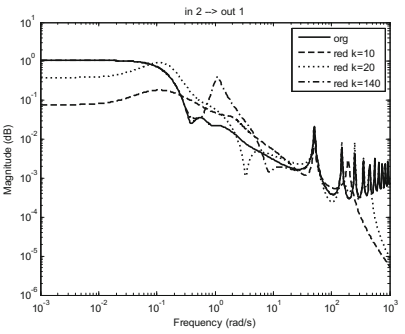


Fig. 14. Frequency response - graph of amplitude for the track $P_{out} \rightarrow P_{in}$

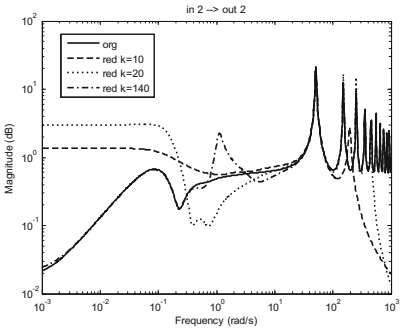


Fig. 15. Frequency response - graph of amplitude for the track $P_{out} \rightarrow h_{out}$

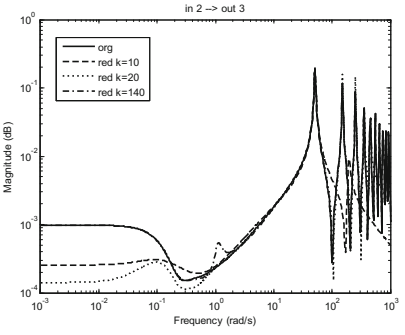


Fig. 16. Frequency response - graph of amplitude for the track $P_{out} \rightarrow M_{out}$

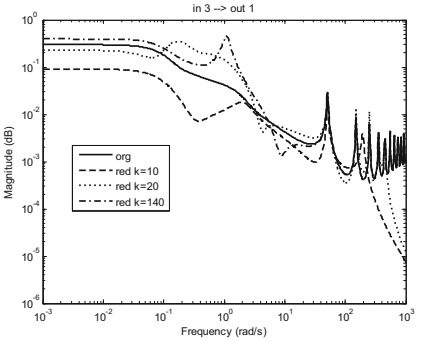


Fig. 17. Frequency response - graph of amplitude for the track $h_{in} \rightarrow P_{in}$

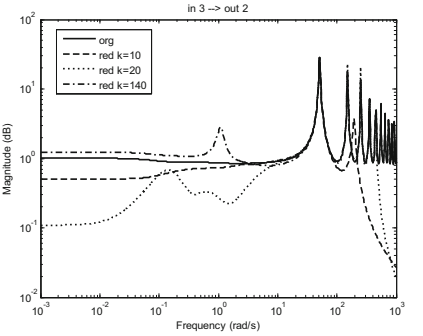


Fig. 18. Frequency response - graph of amplitude for the track $h_{in} \rightarrow h_{out}$

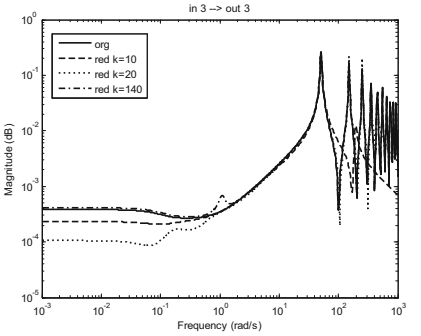


Fig. 19. Frequency response - graph of amplitude for the track $h_{in} \rightarrow M_{out}$

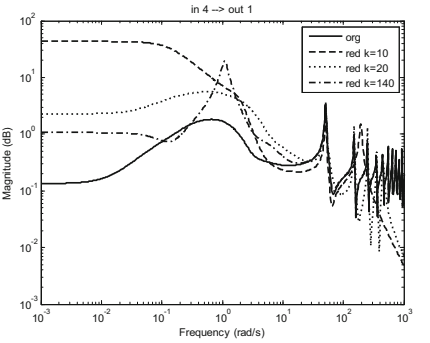


Fig. 20. Frequency response - graph of amplitude for the track $M_{in} \rightarrow P_{in}$

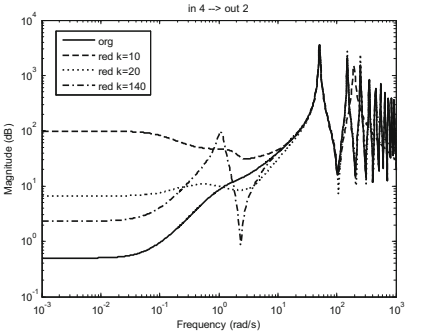


Fig. 21. Frequency response - graph of amplitude for the track $M_{in} \rightarrow h_{out}$

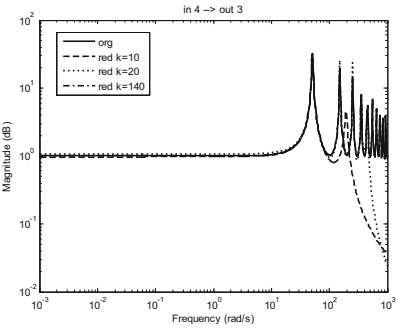


Fig. 22. Frequency response - graph of amplitude for the track $M_{in} \rightarrow M_{out}$

Table 2. The average square relative approximation error of the reduced models

k (rank of whole reduced model)	ρ
10	51.6616
20	10.2439
190	1.7159

Independent reduction of the decomposed subsystems gives good results, which are getting better with increasing of order. For the reduced model of the order of 190 (the slow part reduced to the order of 110 and the fast part unreduced) the frequency characteristics almost overlap the original model (for most tracks there is no visible difference).

8 Conclusion

Many physical objects are represented by both slow and fast physical phenomena simultaneously. The example shown in the article is the one-phase zone of the tubes of the power boiler. Models that describe both phenomena are called stiff, and they are difficult to analyze using a computers.

The paper presents a method allowing to divide the stiff model into a fast and slow part. It should be noted that the method makes it possible to divide the model without any knowledge of the physical phenomena taking place in the model.

The authors propose a modification of the Epsilon decomposition to achieve this goal. The original Epsilon decomposition was not able to decompose the model into the two – slow and fast parts.

The correctness of the presented methodology has been proved experimentally. The authors chose an exemplary physical model, applied the Epsilon decomposition and reduced the model. After combining the decomposed and reduced parts, the frequency characteristics of the reduced model almost matched the frequency characteristics of the original model. It should be noted here that the reduction was carried out on both separated parts completely independently. Reducing models in which the subsystems differ even more in their eigenvalues can produce even better results.

The presented methodology can have many practical applications. For example, the reduction of the stiff model for determining the control system or the reduction of the stiff model to perform the real-time simulations.

References

1. Enns, D.: Model reduction with balanced realizations: an error bound and a frequency weighted generalization. In: Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, pp. 127–133 (1984)
2. Lin, C., Chiu, T.: Model reduction via frequency weighted balanced realization. Control Theory Adv. Technol. **8**, 341–451 (1992)

3. Heck, B., Haddad, A.: Singular perturbation in piecewise - linear systems. IEEE Trans. Autom. Control **34**, 87–90 (1989)
4. Naidu, D., Price, D.: Singular perturbations and time scale in the design of digital flight control systems. NASA Technical Paper 2844 (1988)
5. Kokotovic, P., Khalil, H., O'Reilly, J.: Singular Perturbations Method in Control: Analysis and Design. Academic Press, London (1986)
6. Zečević, A., Šiljak, D.: Control of Complex Systems. Structural Constraints and Uncertainty. Springer, London (2010)
7. Šiljak, D.: Decentralized Control of Complex Systems. Academic Press, New York (1991)
8. Sezer, M., Šiljak, D.: Nested epsilon decompositions of linear systems: weakly coupled and overlapping blocks. SIAM. J. Matrix Anal. Appl. **12**(3), 521–533 (1991)
9. Sezer, M., Šiljak, D.: Nested ε decompositions and clustering of complex systems. Automatica **22**, 321–331 (1986)
10. Miljkovic, D.: https://sourceforge.net/projects/epsd/files/latest/download. Accessed 16 July 2017
11. Stanisławski, W.: Modelowanie i symulacja komputerowa parowników przepływowych kotłów energetycznych [Modeling and computer simulation of flow evaporators of power boilers], Studia i Monografie, z. 124, Oficyna Wydawnicza Politechniki Opolskiej, Opole (2001)
12. Rydel, M.: Zredukowane hierarchiczne modele złożonych obiektów sterowania na przykładzie kotła energetycznego [Hierarchical reduced models of complex control objects on the example of the power boiler], Ph.D. thesis, Faculty of Electrical Engineering, Automatic Control and Informatics, Opole University of Technology, Opole (2008)
13. Raczyński, D.: Redukcja modeli obiektów sterowania z zastosowaniem obliczeń równoległych na przykładzie modelu kotła energetycznego [Model order reduction of control systems with use fo the parallel computing on example of the power boiler], Ph.D. thesis, Faculty of Electrical Engineering, Automatic Control and Informatics, Opole University of Technology, Opole (2014)

Modified HALS Algorithm for Image Completion and Recommendation System

Tomasz Sadowski(✉) and Rafał Zdunek

Faculty of Electronics, Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{tomasz.sadowski,rafal.zdunek}@pwr.edu.pl

Abstract. The paper is concerned with the task of reconstructing missing values in an observed incomplete matrix, assuming its low-rank approximation. The problem has important applications, especially in image processing and social sciences. In our approach, we focus on the problem of recovering missing pixels in images perturbed with impulse noise in a transmission channel as well as estimating unknown ratings in a recommendation system. For solving these problems, we used the modified version of the Hierarchical Least Squares Algorithm (HALS), including the smoothed version, and compared them with other algorithm, such as the SPC-QV. The numerical experiments are carried out for various cases of incomplete data. For image processing, the incomplete images are obtained by removing random pixels and regular grid lines from test images. For recommendation systems, we used real rating matrices from the MovieLens database that contains five-star movie recommendation ratings. The best performance is obtained if nonnegativity and smoothing constraints are imposed onto the estimated low-rank factors.

Keywords: Nonnegative Matrix Factorization · HALS algorithm · Image completion · Recommendation systems

1 Introduction

The problem of image reconstruction occurs in many areas of science and is most often understood as reconstruction of tomographic images with the use of backprojection [1]. This term can be used in a broader sense as reconstruction of missing or distorted parts of an image, which is concerned in this paper. The topic discussed here deals with the task of reconstructing missing pixels in an image. It involves estimating the value of the missing pixels based on the available pixels and certain a priori knowledge about the characteristics of the reconstructed image. This is usually a task that is part of a complex process of noise reduction or image correction.

Currently, the issue of image completion is analyzed by many researchers, as evidenced by recent publications, e.g. [2–5]. Various methodological approaches are known depending on the size and distribution of disorder in an image. If the disturbance is caused by the excision of a certain coherent, relatively large area from a distorted image, the statistical parametric methods [6, 7] that synthesize a texture structure are more effective. The synthesis is based on the parametric models that match

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_2

the statistics of the area disturbed to the rest of the object, by analyzing histograms of color distribution in different resolutions or distribution of other parameters, such as the wavelet coefficients. This approach, however, has significant limitations, because it allows only to synthesize the textures with high stochastic similarity.

Another group of image completion methods are exemplar-based methods [8–15], which fill the missing region with a copy of undisturbed image. Single pixels or so called patches are copied for texture synthesis. This approach is most effective in texture synthesis of large missing areas, but is computationally complex as it is usually implemented with heuristic optimization algorithms. The task of image modeling with patches can also be expressed by a directed graph, unfortunately whose the edges are also associated with a high computational cost.

If image distortions are not coherent, and for example, are defined as randomly missing pixels or multiple pixel groups, then the task of image completion can be reduced to extraction of low-rank features from the analyzed image. Their product approximates the distorted image according to the appropriate criterion. In literature, various algorithms can be found to perform this task, which is usually based on matrix completion algorithms [16–19] or tensor completion [20–22]. In the case of pixel completion in monochrome images, matrix decomposition methods are most often used. For color images, matrix methods can be used separately for each RGB color map or represent a color image with a tensor and directly decompose it into low-rank factor matrices. It should be noted that this approach to image restoration is quite flexible due to a wide choice of: models, constraints imposed on the estimated factors, cost function expressing the measure of similarity of the distorted image and the model, and optimization algorithm.

This paper presents an application aspect of the selected methods for low-rank factor image decomposition, which were used for the task of estimating missing pixels in images with high-missing data rates. Various algorithms were studied and various attempts were made to improve them. In particular, we focus on the usage of the Hierarchical Alternating Least Squares (HALS) algorithm [20] and its modifications [21]. This methodology is used for decomposition of the "damaged" image into the product of non-negative low-order factors. In this sense, it is closely related to the standard Nonnegative Matrix Factorization (NMF) model [21, 22] which has found numerous applications in many areas of science, and is still being developed. One of the main advantages of this model is its great flexibility in choosing the character of the desired solution. The local smoothness of the estimated image also implies the local smoothness of the estimated factors. Enforcing such characteristics of the NMF model is most often accomplished by an additional regularization of the cost function, which determines the specific nature of the solution. In this paper, the imposition of smoothness on the estimated factors is intentional, but the task was performed in a different way than in the work [5]. Smoothness was forced in feature vectors by means of using averaging weighting functions, where low

weighting factors were attributed to those vector features that were perceived to be elements far removed from the outliers.

This papers also shows the application potential of the discussed computational tools for recommendation systems, where the computational problem is very similar to the image completion problem. The aim of a recommendation system is to predict the ratings that a user would give to some items based on its preference determined by the ratings of the selected items. The items can be understand in a wide sense, for example, movies, books, music, games, restaurants, etc. In this paper, we used the data from the Movie Lens group [24]. The idea of the recommendation system can be algebraically illustrated in Fig. 1. The user-movie matrix $\mathbf{M} = [m_{ij}] \in \mathbb{R}^{I \times J}$ should be decomposed into low-rank factors $\mathbf{A} = [a_{ir}] \in \mathbb{R}^{I \times R}$ and $\mathbf{B} = [b_{jr}] \in \mathbb{R}^{J \times R}$. Then, every ratings (observed or not) m_{ij} can be represented by the dot product of the user vector $\underline{\boldsymbol{a}}_i \in \mathbb{R}^{1 \times R}$ (row vector from $\mathbf{A}$) and the movie vector $\underline{\boldsymbol{b}}_j \in \mathbb{R}^{1 \times R}$ (row vector from $\mathbf{B}$). For low-rank decomposition, the model only roughly fits the data, which is preferred for recovering the missing entries in the matrix $\mathbf{M}$.

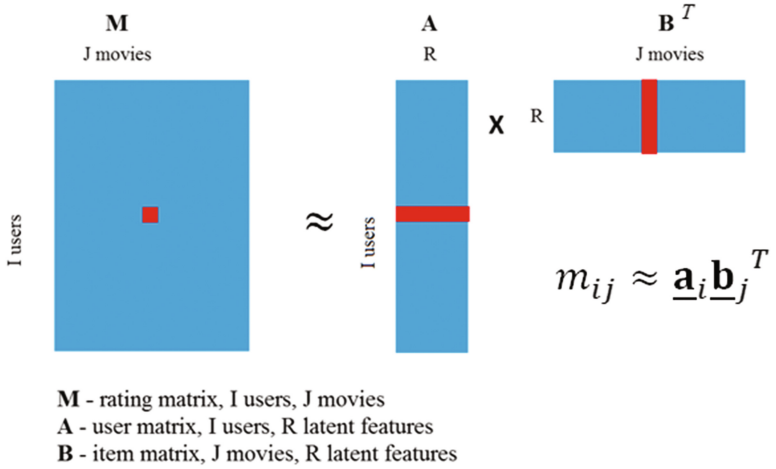


Fig. 1. Idea of representing the missing ratings in a recommendation system. Rating m_{ij} can be approximated by the dot product of user vector $\underline{\boldsymbol{a}}_i \in \mathbb{R}^{1 \times R}$ and item vector $\underline{\boldsymbol{b}}_j \in \mathbb{R}^{1 \times R}$.

This paper is organized as follows: The next chapter shows the mathematical model of the incomplete image reconstruction task. In the next section, the selected methods for a matrix completion problem have been presented. The results of the numerical experiments are presented in Sect. 4. The last section contains summary and conclusions.

2 Problem Formulation

This problem can be formulated as follows [3, 4]:

$$\min_{\mathbf{X}} \operatorname{rank}(\mathbf{X}), \quad \text{s.t. } (\mathbf{X})_{ij} = (\mathbf{M})_{ij}, (i,j) \in \Omega \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{I \times J}$ is the approximated matrix, $\mathbf{M} \in \mathbb{R}^{I \times J}$ is the original incomplete matrix, and Ω is the set (logical matrix) of indexes of the known elements in $\mathbf{M}$. It is assumed that the set is known or can be easily estimated by image segmentation.

The main goal of the missing matrix completion is to find such a minimum-rank matrix $\mathbf{X}$ that has the same set of elements as the matrix $\mathbf{M}$ in the items indicated by the set. This algebraic approach can be applied to the approximation of missing pixels in grey scale (2D matrix) or color (3D matrix or tensor).

3 Selected Methods

All the described methods are based on the Hierarchical Alternating Least Squares (HALS) algorithm, introduced by Cichocki et al. in [20]. The algorithms was demonstrated by many researchers to be one of the most efficient algorithms for solving NMF problems. In comparison to the standard Alternating Least Squares (ALS) algorithm, the HALS demonstrate monotonic convergence, and a greater flexibility for handling penalty and regularization terms.

3.1 HALS for Matrix Completion

The standard HALS sequentially minimizes the Euclidean cost function, where in each iteration a block of variables is updated as in the standard block-coordinate algorithm. Let the cost function be given by the squared Euclidean distance:

$$\mathrm{J}(\mathbf{a}_1, \ldots, \mathbf{a}_R, \mathbf{b}_1, \ldots, \mathbf{b}_R) = \frac{1}{2} \left\| \mathbf{Y} - \mathbf{A}\mathbf{B}^T \right\|_F^2 \tag{2}$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}$, R is the rank of factorization, and $\|\cdot\|_F$ denotes the Frobenius norm. Let $[\tau]_+ = \max\{0, \tau\}$ be the projection onto the set of nonnegative real numbers. Using this information and the straightforward mathematical operations [20–22], the following modified HALS algorithm can be created for solving a matrix completion problem:

Algorithm 1: HALS for matrix completion

Input: $\mathbf{M} \in \mathbb{R}^{I \times J}$ – incomplete matrix, R - rank of factorization,
Ω - set of indexes of the known elements in $\mathbf{M}$, δ – approx. error
Output: $\mathbf{X} \in \mathbb{R}^{I \times J}$ – completed matrix
Initialization: Random nonnegative initialization for $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{B} \in \mathbb{R}^{J \times R}$,
$\mathbf{Z} = \mathbf{0} \in \mathbb{R}^{I \times J}$, $\varepsilon_1 > \delta$, $\varepsilon_2 = 0$
For each $\mathbf{a}_j$ of $\mathbf{A}$ do $\mathbf{a}_j \leftarrow \frac{\mathbf{a}_j}{\|\mathbf{a}_j\|_2}$,
Do while $|\varepsilon_1 - \varepsilon_2| > \delta$
$\tilde{\mathbf{Y}} = \mathbf{Z} + \mathbf{A}\mathbf{B}^T$, $\varepsilon_1 = ||\mathbf{Z}||_F^2$,
$y_{ij} = \begin{cases} m_{ij} \textbf{ if } (i,j) \in \Omega \\ \tilde{y}_{ij} \textbf{ otherwise} \end{cases}$, $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{I \times J}$
Repeat
$\mathbf{W} = \mathbf{Y}^T\mathbf{A}$; $\mathbf{V} = \mathbf{A}^T\mathbf{A}$,
For j=1 to J
$\mathbf{b}_j \leftarrow [\mathbf{b}_j + \mathbf{w}_j - \mathbf{B}\mathbf{v}_j]_+$,
end for j
$\mathbf{P} = \mathbf{Y}\mathbf{B}$; $\mathbf{Q} = \mathbf{B}^T\mathbf{B}$,
For j=1 to J
$\mathbf{a}_j \leftarrow [\mathbf{a}_j q_{jj} + \mathbf{p}_j - \mathbf{A}\mathbf{q}_j]_+$,
$\mathbf{a}_j \leftarrow \frac{\mathbf{a}_j}{\|\mathbf{a}_j\|_2}$,
end for j
until the stopping criterion is satisfied
$\tilde{\mathbf{Z}} = \mathbf{Y} - \mathbf{A}\mathbf{B}^T$, $z_{ij} = \begin{cases} \tilde{z}_{ij} \textbf{ if } (i,j) \in \Omega \\ 0 \textbf{ otherwise} \end{cases}$, $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{I \times J}$, $\varepsilon_2 = ||\mathbf{Z}||_F^2$,
end while
$\tilde{\mathbf{X}} = \mathbf{A}\mathbf{B}^T$, $x_{ij} = \begin{cases} m_{ij} \textbf{ if } (i,j) \in \Omega \\ \tilde{x}_{ij} \textbf{ otherwise} \end{cases}$, $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{I \times J}$

Note that Algorithm 1 iteratively updates the unknown entries in the incomplete image by approximating them with the corresponding dot product of the estimated factor vectors using the HALS algorithm. The iterative process is terminated if the absolute difference between the Frobenius norm of the known entries and the reconstructed ones (determined by the set Ω) drops below a given threshold.

The computational complexity of one iterative step of algorithm is low, and it might be approximated by $\mathcal{O}(IJR)$.

3.2 Smooth NMF for Matrix Completion

In this section, we present the HALS-based algorithm with the factor smoothing for solving a matrix completion problem, which we refer to as the SmNMF-MC. It is based on the concept of the SPC-QV algorithm [5] which was efficiently used for tensor

completion. However, it does not use the CP decomposition, but only the NMF model. The optimization problem is:

$$\min_{\lambda_r,\mathbf{a}_r,\mathbf{b}_r} \frac{1}{2}\left\|\mathbf{M}-\sum_{r=1}^{R}\lambda_r\mathbf{a}_r\mathbf{b}_r^T\right\|_F^2+\sum_{r=1}^{R}\frac{\lambda_r}{2}\left(\rho\|_1\mathbf{L}_1\mathbf{a}_r\|_2^2+\rho_2\|\mathbf{L}_2\mathbf{b}_r\|_2^2\right) \tag{3}$$

where $\mathbf{a}_r \in \mathbb{R}^I$ and $\mathbf{b}_r \in \mathbb{R}^J$ are respective non-negative feature vectors (the r-th column vectors of the corresponding matrices $\mathbf{A}$ and $\mathbf{B}$) that correspond to the vertical and horizontal directions in the image $\mathbf{M}$, λ_r is a scaling factor (estimated from the data), and $(\rho_1, \rho_2) \geq 0$ are the penalty parameters. HALS [21] is used to solve the problem (3), as in the SPC-QV algorithm. The cost function in (3) also contains similar functions forcing smoothness of the estimated factors (penalties). Moreover, additional functions were implemented in this algorithm. We used a moving weight averaging filter that was performed by the smooth (u, 5, 'rlowess') function in Matlab. Therefore, by accepting $\mathbf{L}_1 = \mathbf{L}_2 = \mathbf{I}$, the penalty factors stabilizes the solution in case of ill-conditioning and the smoothing is performed by the direct filtration.

Algorithm 2: SmNMF-MC

Input: $\mathbf{M} \in \mathbb{R}^{I \times J}$, Ω, R, $\theta = [(\rho_1, \rho_2), (\mathbf{L_1}, \mathbf{L_2})]$, δ
Output: $\mathbf{X}$
Initialization: $\lambda \in \mathbb{R}$, $\mathbf{a} \in \mathbb{R}^I$, $\mathbf{b} \in \mathbb{R}^J$
$\mathbf{X} = \lambda\mathbf{a}\mathbf{b}^T$;

$$y_{ij} = \begin{cases} m_{ij} \textbf{ if } (i,j) \in \Omega \\ x_{ij} \textbf{ otherwise} \end{cases}, \quad \mathbf{Y} = [y_{ij}] \in \mathbb{R}^{I \times J}$$

$\mathbf{Z} = \mathbf{Y} - \mathbf{X}$;
Do while $|E_2 - E_1| \geq \delta$
  For $r = 1, \ldots, R$
    $E_1 = \|\mathbf{Z}\|_F$
    $\mathbf{Y}_r = \mathbf{Z} + \lambda_r\mathbf{a}_r\mathbf{b}_r^T$;
    $\mathbf{a}_r = (\mathbf{I}_1 + \rho_1\mathbf{L}_1\mathbf{L}_1^T)^{-1}\mathbf{Y}_r\mathbf{b}_r$; $\mathbf{a}_r \leftarrow \frac{\mathbf{a}_r}{\|\mathbf{a}_r\|_2}$;
    $\mathbf{a}_r \leftarrow smooth(\mathbf{a}_r, 5, \text{'rlowess'})$;
    $\mathbf{b}_r = (\mathbf{I}_2 + \rho_2\mathbf{L}_2\mathbf{L}_2^T)^{-1}\mathbf{Y}_r^T\mathbf{a}_r$; $\mathbf{b}_r \leftarrow \frac{\mathbf{b}_r}{\|\mathbf{b}_r\|_2}$;
    $\mathbf{b}_r \leftarrow smooth(\mathbf{b}_r, 5, \text{'rlowess'})$;
    $\lambda_r = \frac{\mathbf{a}_r^T\mathbf{Y}_r\mathbf{b}_r}{1+\rho_1\|\mathbf{L}_1\mathbf{a}_r\|_2^2+\rho_2\|\mathbf{L}_2\mathbf{b}_r\|_2^2}$;

$$z_{ij} = \begin{cases} [\mathbf{Y}_r - \lambda_r\mathbf{a}_r\mathbf{b}_r^T]_{ij} \textbf{ if } (i,j) \in \Omega \\ 0 \qquad\qquad\qquad\quad \textbf{otherwise} \end{cases}, \quad \mathbf{Z} = [z_{ij}] \in \mathbb{R}^{I \times J},$$

    $E_2 = \|\mathbf{Z}\|_F$
  end for
end while

$$\widetilde{\mathbf{X}} = \mathbf{A}\mathbf{B}^T, \quad \mathbf{X} = \begin{cases} m_{ij} \textbf{ if } (i,j) \in \Omega \\ \tilde{x}_{ij} \textbf{ otherwise} \end{cases}$$

4 Numerical Tests

The experiments have been run on the computer with the following parameters: Windows 10, Intel i7-4790K 4.00 GHz, 8 GB, Matlab R2016a.

The image completion tests were performed on the real world image from Positron Emission Tomography (PET) of the resolution (285 $\times$ 277 pixels), which is presented on Fig. 2. SPC-QV algorithm were taken from the authors' pages [23]. The tests were performed on incomplete images that were generated by removing from the original image: (a) randomly selected 50, 70 and 90% of pixels, (b) a single line of pixels forming a regular grid of 10 pixels wide. The same algorithms were used with the MovieLens 20M Dataset [24] for analyzing a recommendation system. This set contains 20 million ratings and 465000 tag applications applied to 27000 movies by 138000 users. The estimated results were validated with the randomly selected ratings (101 users, 87 movies). These tests were carried out similarly to the image completion problem.

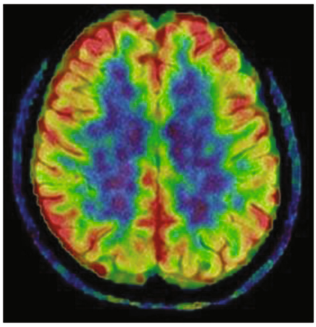


Fig. 2. Tested image: PET-scan.

The original and estimated images are shown on Fig. 3. The Signal-to-Interference Ratio (SIR) measure was used to evaluate the quality of estimation [21]. The SIR results are presented in Table 1, and the computational time in Table 2.

The results do not show clearly which of the methods is absolutely the best. The tensor methods in most cases show faster convergence than the matrix ones. In order to achieve the same quality of reconstruction, the SmNMF-MC requires about 5000 iterations while the SPC-QV only about 800. However, the SmNMF-MC provides higher SIR values than the others if the data are not highly incomplete. For the test with grid disorder, the HALS method does not provide satisfactory results; the grid still exists on the recovered image and additionally the image is blurred. The SmNMF-MC method removes the grid lines, but one can see their shadow on the reconstructed image. In the case of the SPC-QV, the lines are removed, however, the background noise is still visible.

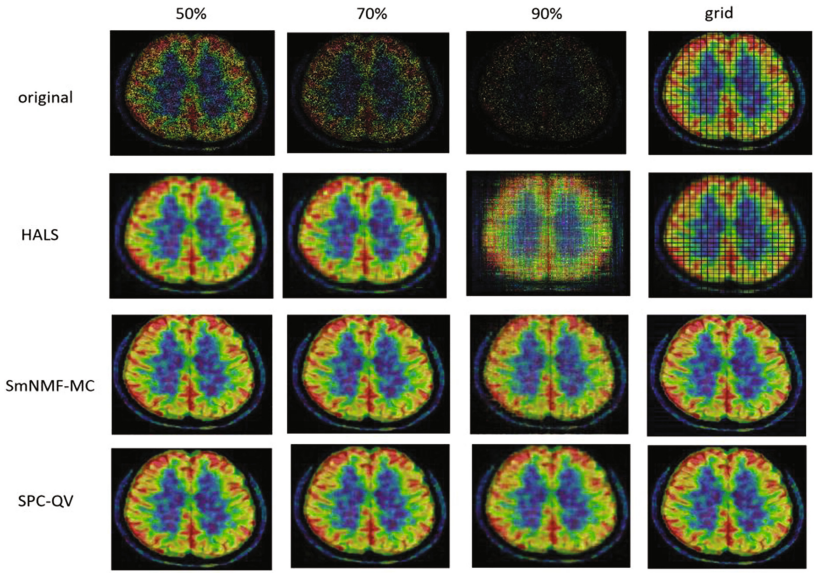


Fig. 3. Results for image completion

Table 1. SIR [dB] results for image completion

SIR [dB]			
Missing image	HALS	SmNMF-MC	SPC-QV
50%	17.06	26.11	24.43
70%	15.44	22.71	21.71
90%	5.29	15.37	17.12
Grid	7.23	11.39	28.46

Table 2. Time [s] for image completion

Time [s]			
Missing image	HALS	SmNMF-MC	SPC-QV
50%	34.188	131.559	70.017
70%	64.996	224.511	72.483
90%	164.706	101.164	78.557
Grid	8.685	66.445	43.442

The results for the recommendation system are listed in Tables 3 and 4. For this test, the ranking of the tested algorithms is slightly different. All the algorithms are very efficient, but the HALS given by Algorithm 1 seems to work the best. It is also the fastest one, which is even more important in this case. The results in terms of the SIR values show that we are able to recover the users' preference with a high accuracy. Hence, Algorithm 1 seems to be the best for recommendation systems.

Table 3. SIR [dB] results for recommendation system

SIR [dB]			
Missing data	HALS	SmNMF-MC	SPC-QV
50%	21.25	20.81	21.19
70%	17.45	17.69	17.95
90%	12.33	13.03	12.86
Grid	34.07	18.13	28.80

Table 4. Time [s] for recommendation system

Time [s]			
Missing data	HALS	SmNMF-MC	SPC-QV
50%	0.205	4.004	35.581
70%	0.153	0.478	31.139
90%	0.151	0.394	23.343
Grid	0.088	0.388	33.568

5 Conclusion

Three HALS-based methods are compared in terms of their efficiency for solving various matrix completion problems. The results strongly depends on the application. For the image completion, the best results were obtained using the SPC-QV method with tensor decomposition. For the recommendation system, the best is the simplest, i.e. the modified HALS given by Algorithm 1. It gives the best performance and in the shortest time. However, further optimization is necessary for algorithms to make them more efficient for larger datasets [24]. The methods studied might be useful in a wide range of research areas, including recognition and image compression [2], DNA sequencing [25], hyperspectral images analysis [26], etc.

Acknowledgment. This work was partially supported by the Grant 2015/17/B/ST6/01865 funded by National Science Center (NCN) in Poland. The other part is supported by the statutory Grant 0401/0098/16.

References

1. Herman, G.T.: Fundamentals of Computerized Tomography: Image Reconstruction from Projections, 2nd edn. Springer, New York (2009)
2. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
3. Guo, X., Ma, Y.: Generalized tensor total variation minimization for visual data recovery. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
4. Phan, A.H., Cichocki, A., Tichavsky, P., Luta, G., Brockmeier, A.: Tensor completion through multiple Kronecker product decomposition. In: Machine Learning for Signal Processing, ICASSP (2013)

5. Yokota, T., Zhao, Q., Li, C., Cichocki, A.: Smooth PARAFAC decomposition for tensor completion. IEEE Trans. Sign. Process. **64**(20), 5423–5436 (2016)
6. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, pp. 229–238. ACM, New York (1995)
7. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int. J. Comput. Vis. **40**(1), 49–70 (2000)
8. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proceedings of the International Conference on Computer Vision, ICCV 1999, vol. 2, pp. 1033–1038. IEEE Computer Society, Washington, DC (1999)
9. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, pp. 479–488. ACM Press/Addison-Wesley Publishing Co., New York (2000)
10. De Bonet, J.S.: Multiresolution sampling procedure for analysis and synthesis of texture images. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, pp. 361–368. ACM Press/Addison-Wesley Publishing Co., New York (1997)
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graph. **22**(3), 277–286 (2003)
12. Liang, L., Liu, C., Xu, Y.Q., Guo, B., Shum, H.Y.: Real-time texture synthesis by patch-based sampling. ACM Trans. Graph. **20**(3), 127–150 (2001)
13. Wu, Q., Yu, Y.: Feature matching and deformation for texture synthesis. ACM Trans. Graph. **23**(3), 364–367 (2004)
14. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceeding the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, pp. 327–340. ACM, New York (2001)
15. Shu, H., Senhadji, L., Chen, Y., Wang, L., Wu, J., Han, X.: Linear total variation approximate regularized nuclear norm optimization for matrix completion. In: Abstract and Applied Analysis, no. 765782 (2013)
16. Ji, H., Liu, C., Shen, Z., Xu, Y.: Robust video denoising using low rank matrix completion. In: CVPR, pp. 1791–1798. IEEE Computer Society (2010)
17. Li, W., Zhao, L., Lin, Z., Xu, D., Lu, D.: Non-local image inpainting using low-rank matrix completion. Comput. Graph. Forum **34**(6), 111–122 (2015)
18. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **35**(1), 208–220 (2012)
19. Chen, Y.L., Hsu, C.T., Liao, H.Y.M.: Simultaneous tensor decomposition and completion using factor priors. IEEE Trans. Pattern Anal. Mach. Intell. **36**(3), 577–591 (2014)
20. Cichocki, A., Zdunek, R., Amari, S.I.: Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization, pp. 169–176. Springer, Heidelberg (2007)
21. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Chichester (2009)
22. Zdunek, R.: Nieujemna Faktoryzacja Macierzy i Tensorów: Zastosowanie do Klasyfikacji i Przetwarzania Sygnałów. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2014)
23. https://sites.google.com/site/yokotatsuya/home/software/smoothparafac-decomposition-for-tensor-completion. Accessed 19 May 2017

24. Harper, F.M., Konstan, J.A.: The movie lens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2015)
25. Troyanskaya, O.G., Cantor, M.N., Sherlock, G., Brown, P.O., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics **17**(6), 520–525 (2001)
26. Zhang, H., He, W., Zhang, L., Shen, H., Yuan, Q.: Hyperspectral image restoration using low-rank matrix recovery. IEEE Trans. Geosci. Remote Sens. **52**(8), 4729–4743 (2014)

Proposal of Input Shaper in Real Applications

Peter Šarafín[1(✉)], Juraj Miček[1], Ondrej Karpiš[1], and Judith Molka-Danielsen[2]

[1] University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia
{Peter.Sarafin, Juraj.Micek, Ondrej.Karpis}@fri.uniza.sk
[2] Molde University College, PO. Box 2110, 6402 Molde, Norway
Judith.Molka-Danielsen@himolde.no

Abstract. In control applications, we often encounter systems that respond to the control signal change by vibration at their output. These vibrations are undesirable and various techniques are used to remove them. One of the approaches designed to suppress undesired vibrations is input shaping. This paper is devoted to discrete input shapers and to the identification and simulation of weakly damped discrete systems. Experimentally verified results are also presented.

Keywords: Input shaping · Accelerometer · Input shaper

1 Introduction

To control weakly damped dynamic systems, input shaping method is often used. Input shaping is a method that has been used mainly to control the movement of gantry cranes.

This method was designed to ensure that the crane movement is controlled so that the suspended load does not break. In general, it can be stated that with the need of input shaping, we always encounter in the control applications of the positioning systems with the flexible elements. With the development of mechatronic systems, input shaping comes to the fore. In Fig. 1, the response of a weakly damped system is shown. In practice, however, there are often limitations that need to be taken into account in the theoretical design of the input shaper. These limitations include in particular the limitation of the action quantity and the relatively low resolution of the output power elements [1].

Over the time, we have encountered other interesting applications, including the control of fast moving elevators or the control of the movement of conveyor belts in production lines, especially in the food industry [2]. The role of the input shaper is to modulate the frequency spectrum of the control signals so that in the region of the resonant elevation there are no residual vibrations in the controlled system output. On the basis of the above we can state that basically it is a proposal of a series correction member whose task is to modify the frequency characteristics of the controlled system (Fig. 2).

Input shaping has been successfully applied to the problem of maneuvering flexible structures without excessive residual vibrations. Non-negative input shapers are often

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_3

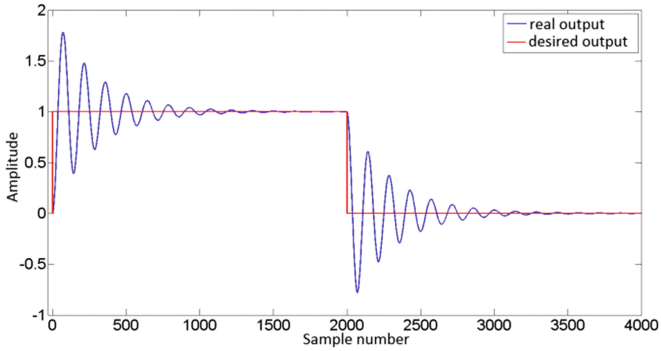


Fig. 1. Weakly damped system response

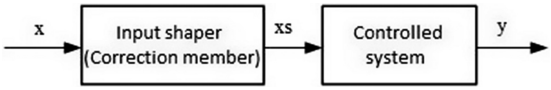


Fig. 2. The involvement of the input shaper

required because they can be used with any (unshaped) signals and do not cause the system instability (if the unshaped signals do not cause system instability).

2 The Input Shaping Principle

The reference input signal used to control the positioning systems can greatly affect the performance of the system [3, 4]. Numerical filtering and shaping of input signals are known methods of forming control signals to reduce oscillations. From the design of the robust input shaper [5, 6] the researchers found substantial arguments that input shapers are more suitable for the applications containing flexible mechanical systems with one or two dominant states than the notch and low-pass filters [5].

Digital filters, as well as shapers, generate pulse sequences, which by convolution with the input signal produce a shaped reference signal. This process is illustrated in Fig. 3. If a filter or an input shaper is designed correctly, the shaped signal will provide the desired state change without significant residual vibrations.

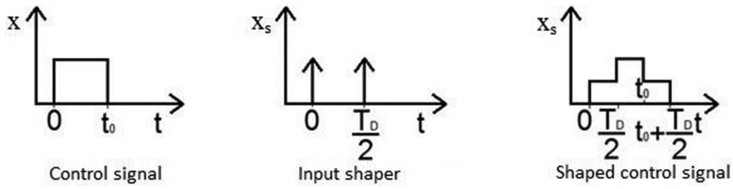


Fig. 3. Input shaping process

2.1 Theoretical Background

Input shaping is a process that modulates the control signal to prevent the resonant output of the system. In other words, the shaper filters frequency that causes resonance in the system in these signals. Parameters of the input shaper are designed so the response of the system to the input signals corresponds to the desired resonant characteristic.

Various applications have been developed for a wide range of input shapers. A commonly used input shaper is the Zero Vibration (ZV) shaper, which can be described by the relationship (1).

$$ZV = \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & T \end{bmatrix}, \; where \; T = \frac{\pi}{\omega\sqrt{1-\varsigma^2}}, \; K = e^{-\frac{\varsigma\pi}{\sqrt{1-\varsigma^2}}}, \tag{1}$$

ω stands for resonance frequency and ς represents system damping.

This input shaper has the shortest time required to execute the arithmetic operations of the system only by positive pulses. This time is important because the convolution with the input shaper extends the control time according to the transition time of the input shaper. If the ZV shaper is designed with a perfect model, it eliminates all vibrations. If there is an incorrect model, some oscillations occur [6].

If it is necessary to provide resistance to modeling errors, a Zero Vibration Derivative (ZVD) input shaper may be used, which may be described by Eq. (2). This shaper forces the derivation of the function with respect to model errors equal to zero. The tax on adding this robustness is the increased time of realization of the arithmetic operations of the input shaper, and thus the calculation delay of the system [7].

$$ZVD = \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{1+2K+K^2} & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & T & 2T \end{bmatrix}, \tag{2}$$

where T and K have the same meaning as for the ZV shaping machine. Another type of input shaper is the Extra-Insensitive (EI) shaper (3). The execution time of the arithmetic operations of the input shaper is the same as for the ZVD shaper, but its insensitivity to change of the system parameters is considerably higher. The sensitivity of the EI shaper depends on the allowed vibration magnitude in the exact model. In general, the allowed vibration magnitude is determined to a value that is equal to the upper limit of acceptable residual vibrations. The reason for this is the fact that by increasing the allowed vibration size, the insensitivity to modeling errors increases.

$$EI = \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1+V}{4} & \frac{1-V}{2} & \frac{1+V}{4} \\ 0 & T & 2T \end{bmatrix}, \; where \; T = \frac{\pi}{\omega\sqrt{1-\varsigma^2}}, \tag{3}$$

and V represents the rate of insensitivity to system oscillations.

Designing the Input Shaper in z-plane. In control of the weakly damped systems, the process of input shaping can also be solved by designing suitable discrete correction elements that modulate the frequency spectrum of input control signals to suppress

residual system oscillations. This task can be solved by appropriately locating the zeros of the transfer function of the correction elements from points of the z-plane corresponding to the poles of the controlled system [2]. The poles of the continuous system can be expressed

$$p_{1,2} = -\frac{\varsigma}{T} \pm \frac{\sqrt{1-\varsigma^2}}{T}. \tag{4}$$

Since for the complex variable applies $z = e^{sT_{vz}}$, the poles of the continuous system from the s-plane are transformed into points of the z-plane (5).

$$p_{d_{1,2}} = e^{-\frac{T_{vz}\varsigma}{T}} \cdot e^{\pm j\frac{T_{vz}\sqrt{1-\varsigma^2}}{T}}, \tag{5}$$

where T_{vz} is the sampling period of the discrete system. It is then appropriate to place the zeros of the input shaper to these points. By locating the zeros of z-transfer function of the input shaper into points corresponding to the position of the poles of the controlled system, the transfer function of input shaper has form

$$F(z) = C \cdot \left(1 - z_1 \cdot z^{-1}\right) \cdot (1 - z_2 \cdot z^{-1}), \tag{6}$$

where C represents the normalization constant. In order to preserve the causality of the system it is advisable to add to the centre of the coordinate system as many poles as zeros characterize the transfer of the input shaper [8]. The transfer function of the input shaper can be presented as

$$F(z) = C \cdot (a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}), \tag{7}$$

where

$$\begin{aligned} a_0 &= 1, \\ a_1 &= (-z_1 + z_2), \\ a_2 &= z_1 \cdot z_2, \\ C &= \frac{1}{a_0 + a_1 + a_2}. \end{aligned} \tag{8}$$

If we choose the sampling period T_{vz} so that the sum $z_1 + z_2$ equals zero, this choice corresponds to the positive ZV shaper with transfer function

$$F(z) = C \cdot \left(a_0 + a_2 \cdot z^{-2}\right). \tag{9}$$

For the sum $z_1 + z_2$ to be zero, the complex conjugate roots z_1 and z_2 must lie on the imaginary axis. The solution with the shortest transition time thus expands into the relationship (10).

$$T_{vz} = \frac{\pi T}{2\sqrt{1-\varsigma^2}} \tag{10}$$

The so-called ZV input shaper may be described by the relationship (11).

$$ZV = \begin{bmatrix} A_j \\ t_j \end{bmatrix} = \begin{bmatrix} \frac{1}{1+a_2} & \frac{a_2}{1+a_2} \\ 0 & 2T_{vz} \end{bmatrix} \tag{11}$$

3 Design and Application of the Input Shaper

The requirement to shape the control signal resulted from the need for a non-resonant control of the device that performs the measurement during the movement of the mechanical arm (Fig. 4).

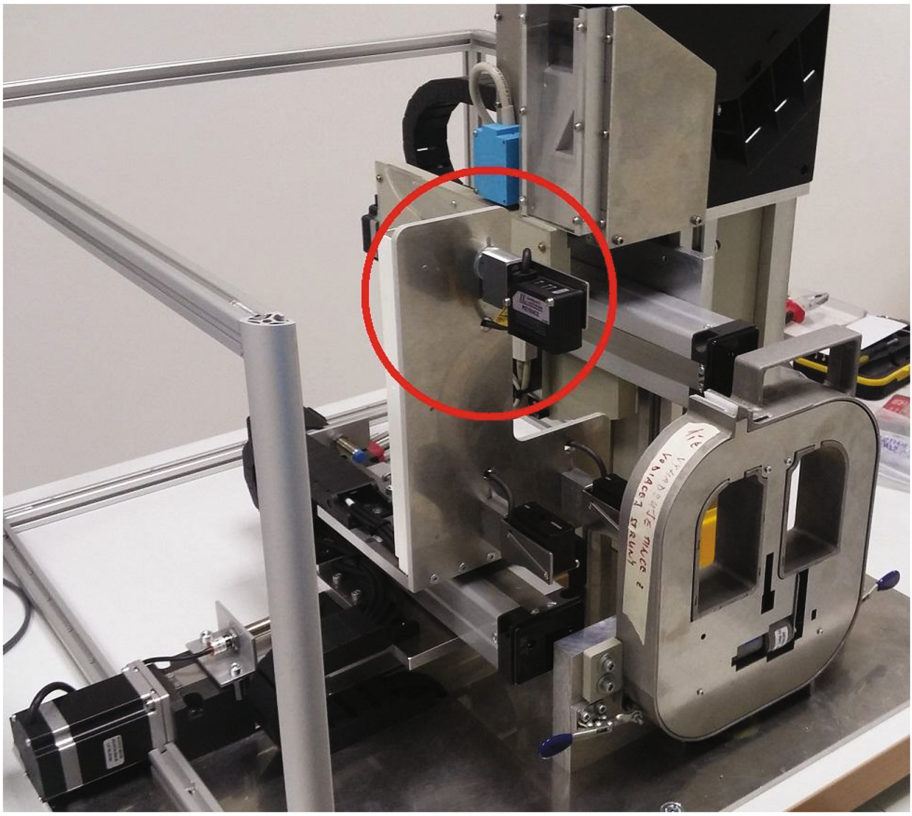


Fig. 4. Coin hopper tester device with marked critical part

The movement of the arm is ensured by a linear drive that is driven by stepping motors with the JK1545DC excitation circuit. The engine torque is 1.8 Nm and the phase current is 2.5 A. Stepper motor resolution is 1.8°, which corresponds to 200 steps per revolution.

For a particular application of the motion control of the coin hopper tester arm, stepping with the micro steps was chosen. The vibrations caused by the primary control of the motors have been minimized.

However, the movement of the arm caused vibrations, the influence of which affected significantly the accuracy of the measurements of the physical dimensions of the coin hopper [9, 10]. For this reason, we decided to investigate the dynamic properties of the system and then apply an input shaper that suppresses the residual vibrations.

When identifying the dynamic properties of the system, we used a printed circuit board (PCB) with an accelerometer. As a control element, the ATmega168 microcontroller was used. Its task was to provide the LSM303DLHC accelerometer configuration and send RF data through the RFM73 RF module to eliminate the undesirable phenomena associated with the use of wire communication. The accelerometer was set to measure acceleration in the axes x, y and z. The sampling frequency was set to 400 Hz, which corresponds to the maximum possible recording speed when the accelerometer is used.

On the receiver side, we used a PCB with ATmega8A microcontroller. The role of this microcontroller was to receive the data recorded by the RFM73 RF module and then send the results using the UART peripheral. In order to be able to continue working with this data on the computer, we used a communication interface converter to convert between USB and UART peripherals standards.

The measuring device was positioned such that the movement of the arm manifested mainly in one axis of the accelerometer (Fig. 5). On the PC, the incoming data has been aggregated into a file so it can be analyzed.

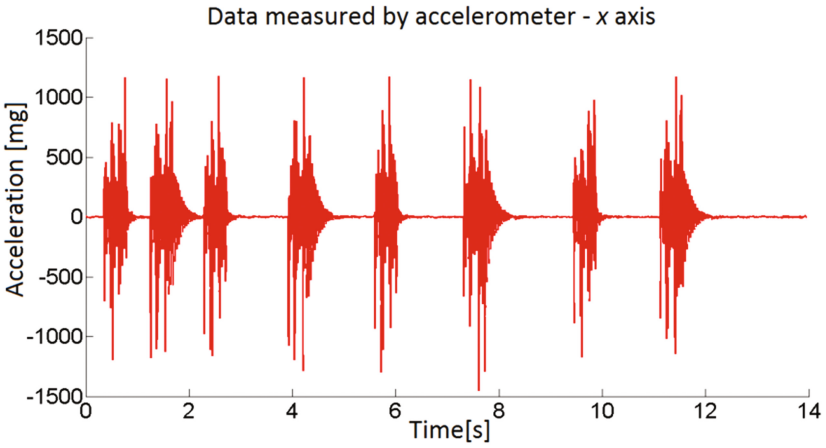


Fig. 5. Data collected from the accelerometer critical axis

Using the identification method used, we determined the zeroes and poles of the system, and thus the actual frequency and damping of the system.

The knowledge of our own period and the damping of the system allowed us to design an input shaper with the required properties. Our attention has been drawn to the ZV shapers, how much it can generally be considered appropriate in knowing the system parameters. By applying the ZV input shaper, we have acquired a new, shaped, input signal to control the movement of the motors. Figure 6 shows shapers and modifications of the control step signal when applied.

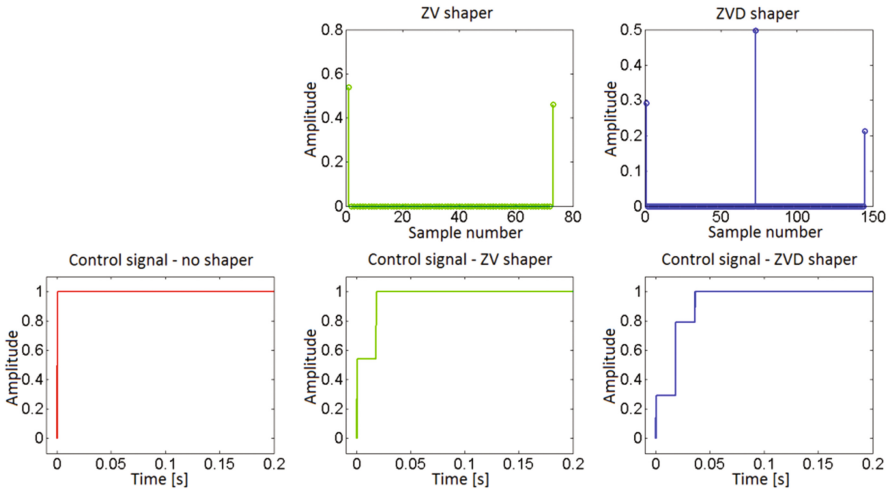


Fig. 6. Shapes and corresponding modifications of the control step signal

Since we know that the system was excited by a step signal, for illustration purposes, besides the basic set of ZV shapers, we have defined the corresponding control signals (Fig. 7) and system reaction set (Fig. 8). Based on our model, to radically suppress residual frequencies, the application of the designed ZV shaping machine is sufficient. The result of the control process was surprising since the signal did not eliminate the resonance recorded at the output of the system so markedly. This was due to the fact that we did not take into account the limitations of action elements, stepper motors. By defining the minimum time at which it is necessary to ensure the required rise (in our case 100 ms), we also indirectly determine the maximum admissible order of the input shaper, in which the rise time of the control signal is not longer than the determined guaranteed response time of the system. From Fig. 7 we can conclude that the maximum ZVD4 type of shaping is permissible under the given restrictions.

Another significant limitation resulting from the engines used is the minimum sampling period of the actuator (in our case 10 ms). This limit specifies the maximum

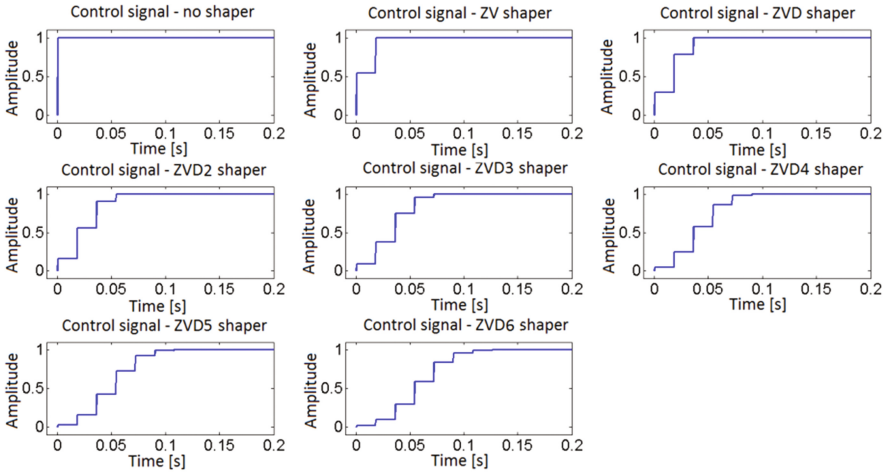


Fig. 7. Basic set of control signal modifications

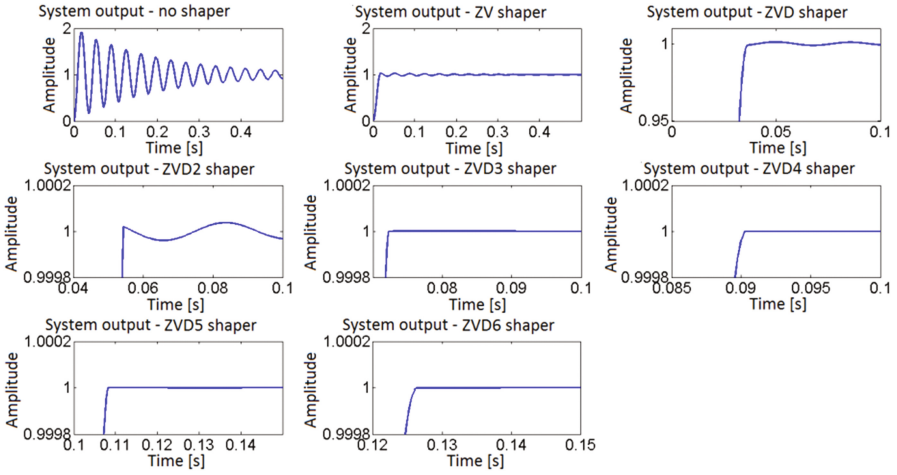


Fig. 8. Reaction of the system after the application of the corresponding input shaper

allowable number of control signal changes per second. It follows that at the given sampling period of the actuator and the time at which it is necessary to ensure that the desired change of the control signal is achieved, it clearly defines the highest permissible order of the input shaper.

The response of the moving part of the system to the control signal was also recorded for clarity by an optical sensor located directly on the device. Change of the

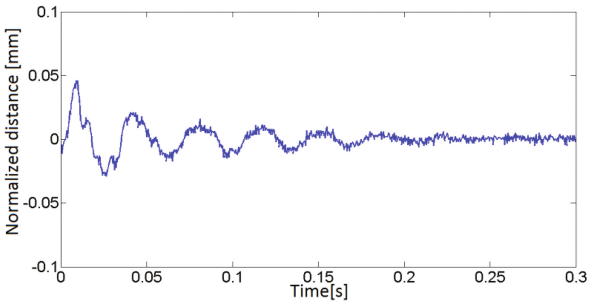


Fig. 9a. Basic set of control signal modifications

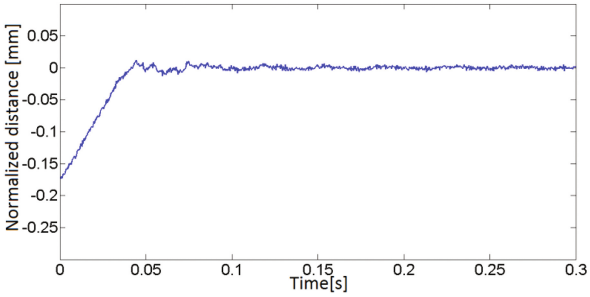


Fig. 9b. Response to the need to change the position of the arm-used ZVD3 shaper

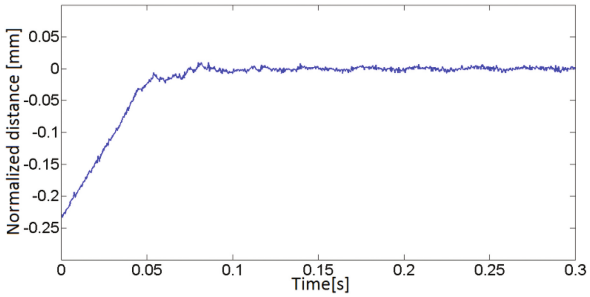


Fig. 9c. Response to the need to change the position of the arm-used ZVD4 shaper

momentary distance of the arm from the target position when applying different types of shapers can be represented graphically. Experimental results are shown in Figs. 9a, 9b, 9c and 9d.

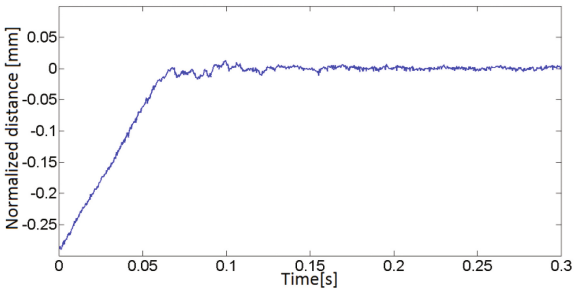


Fig. 9d. Response to the need to change the position of the arm-used ZVD5 shaper

4 Conclusion

Undesired vibrations can interfere with mechanical systems. In systems where vibrations are driven by control signals, signal shaping approaches have been shown to be appropriate and effective. Due to the design of a suitable correction element in the form of input shaper, it is necessary to define the system parameters. Knowing the exact location of the poles of the controlled system model serves as a basis for eliminating the resonant output of the system.

In practice, however, there are often limitations that need to be taken into account in the theoretical design of the input shaper. These limitations include in particular the limitations of the action quantity and the relatively low resolution of the output power elements.

The resistance to the errors of the resulting shaper can be increased by increasing the number of zeros of the transfer function of the correction member. Especially for tasks that are characterized in that the sampling period of the actuator is comparable to the required transition time, the poles are suggested to be repeated in order to achieve the desired behavior. In the case where the maximum transition time is limited, with a minimum sampling time defined based on the dynamic properties of the actuator, the method for calculating the pole multiplicity is proposed. This will ensure that the required properties are met by both the actuator and the user. Applying the highest possible order shaper that meets the design limitations is desirable because all the conditions set for the input shaper are met. At the same time, the system will be maximally resistant to model errors with regard to system time constraints.

Methods of input shaper designing have been simulated. The accuracy and suitability of the solution was also verified on a real device, using data obtained from an accelerometer and an optical distance sensor.

References

1. Miček, J., Kovář, O.: Tvarovač riadiacich signálov: poznámka k voľbe periódy vzorkovania a minimalizácia chýb spôsobených kvantovaním času, Elektrorevue 13, February 2011. ISSN 1213-1539

2. Miček, J.: Alternatívny prístup k návrhu tvarovača riadiacich signálov. AT&P J 3 (2010). ISSN 1335-2237
3. Singhose, W.: Command shaping for flexible systems: a review of the first 50 years. Int. J. Precision Eng. Manuf. **10**(4), 153–168 (2009)
4. Singhose, W., Seering, W.: Command generation for dynamic systems. Lulu (2010). 978-0-9842210-0-4
5. Singer, N., Seering, W.: Design and comparison of command shaping methods for controlling residual vibration. In: Proceedings of IEEE International Conference on Robotics and Automation, Scottsdale, AZ, vol. 2, pp. 888–893 (1989)
6. Singer, N., Seering, W.: Preshaping command inputs to reduce system vibration. J. Dynam. Syst. Meas. Control **112**, 76–82 (1990)
7. Singer, N., Singhose, W., Seering, W.: Comparison of filtering methods for reducing residual vibration. Eur. J. Control **5**, 208–218 (1999)
8. Singhose, W., Seering, W., Singer, N.: Residual vibration reduction using vector diagrams to generate shaped inputs. ASME J. Mech. Des. **116**, 654–659 (1994)
9. Tang, T.: Reduction of mechanical resonance based on load acceleration feedback for servo system. Electron. Eng. **34**(7), 15–17 (2007)
10. Wang, H.: Vibration rejection scheme of servo drive system with adaptive notch filter. In: Proceedings of 37th IEEE PESC, pp. 1–6 (2006)

A Structure-Driven Process of Automated Refactoring to Design Patterns

Anna Derezińska(✉)

Institute of Computer Science, Warsaw University of Technology,
Warsaw, Poland
A.Derezinska@ii.pw.edu.pl

Abstract. Design patterns can be introduced into an existing code by a code restructuring. It is counted to one of goals of code refactoring. This paper presents a process that automates this kind of refactoring. The approach is based on a structural code analysis aimed at design pattern relevance. The process consists of three main phases: code analysis, determination of a refactoring range, and realization of code restructuring. The latter is a complex code2code transformation, which comprises a series of code refactoring steps. A decision about a range of a design pattern application is taken by a user or can be automated. In both cases, it is supported by a software relevance metric. A framework for the whole process has been implemented as an extension of Eclipse. Code refactoring to exemplary design patterns can be performed in a prototype tool for Java programs.

Keywords: Code refactoring · Design patterns · Eclipse

1 Introduction

Design patterns are often brought into an application during not only initial design or coding phases, but also later during program evolution and maintenance [1]. Therefore, they can be a target of a refactoring process, a technique to improve maintainability of software by changing its structure without altering its external behavior [2].

Kerievsky presented many transformation scenarios that combine refactoring with the practice of designing with patterns [3]. Different kinds of refactorings: *to, towards* or *from*, describe transformations that provide an application of a given pattern, a code modification in direction to a pattern usage, or eliminate a pattern, accordingly.

Contemporary development environments (Eclipse, IntelliJ IDEA, NetBeans, Visual Studio) support realization of a limited set of basic refactorings [2] working on code elements (e.g. *rename, move, extract*), class inheritance hierarchy (e.g. *pull up, push down*), but scarcely approaching towards design patterns (*introduce factory*). They assist also creation of a script of refactorings, but are still insufficient to perform complex refactorings targeting design patterns.

Realization of complex refactorings, including towards/to many design patterns, is more difficult than the basic ones. Some solutions are based on search-based refactoring [4] and also deal with design patterns [5–7]. Another resent and the most wide-ranging

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_4

approach has been presented by Kim et al. [8]. Implementations of refactorings to selected patterns have also been proposed [9–11].

Recommendation of basic refactoring actions [2] is primarily based on recognition of code smells, i.e. catalogued code shortcomings in an object-oriented design. Refactoring is often stimulated by software quality observed by a developer, implied by measurement of software metrics [12–14] and suggested by specialized code analyzers [15–17]. However, even in case of a simple refactoring, it has been observed that many developers faced selection problems [18].

Usage of design patterns can be associated with removal of some code smells [3], but in general, it is often aimed at more complex restructuring of a program and improving its flexibility, scalability, extensibility, testability, etc. Necessity and adequacy of utilization of design patterns might be more arguable than application of simple refactorings, e.g. an increase in code size could be observed. In some automated search-based approaches [5–7], common software metrics are used to asses a software quality, and serve as a base for a fitness function to decide about a refactoring to be performed. In dependence of different quality measures, we could obtain a code with, for example, lower coupling and higher cohesion. However, there seems to be no evidence that a design pattern is introduced according to a semantically confirmed architectural concept. In the similar context, Tokuda and Batory have noticed "Relating two arbitrary classes via inheritance relations can be meaningless" [19]. In another approach [8], a focus is on scripting high-level refactorings and not on recommending when and which refactoring to apply.

It should be noted, that what is the most challenging is not a direct code manipulation in order to introduce a desired design pattern, but taking a decision which part of the code should be used and correspond to a target design pattern. Moreover, refactoring to design patterns could be performed in large and nontrivial software systems written by other developers, especially in legacy systems [20].

Therefore, in a refactoring process proposed in this paper, we take into account not only a code transformation, but also analysis of structural dependencies and pattern-related decisions. Those decisions are based on our original software metrics that evaluate relevance of a given code extract to a pattern under concern. They are not general quality metrics [12–14], as typically used in other approaches, although such quality metrics could be additionally used in further extensions of the process. The solution with selected refactorings has been implemented for Java programs working in the Eclipse environment [21].

The rest of the paper is structured as follows. The next section reviews the related work. A process overview is presented in Sect. 3. Section 4 describes an exemplary process based on a selected refactoring. A framework implementing the approach and experimental evaluation are briefly discussed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

Much research has been performed on recommending refactoring operations from the basic Fowler set [2], e.g. the mostly used transformations *move* or *extract* [15–17].

Possibility of automated refactoring versus a manual one aimed at an object-oriented program restructuring has been considered by Tokuda and Batory [19]. They suggested benefits of automated transformations, while decisions of refactoring to apply were made by a user. The most complete collection of scenarios dealing with refactoring towards/to design patterns was gathered by Kerievsky [3]. Refactoring to design patterns was also described as a series of minitransformations including their pre-, post-conditions and transformation steps [22].

Current advances in the domain can be reviewed from two perspectives. State of the art on design pattern processing has recently been studied by Mayvan et al. [23]. Trends, opportunities, and challenges of the software refactoring have been reviewed by Abebe and Yoo [24]. Both reported on some research combining the areas.

Finding of a best sequence of refactorings to be applied in a software artifact can be regarded as an optimization problem that can be solved using search techniques [4]. A fitness function of such approaches is usually based on software quality metrics [12–14], and a transformation process focuses on the improvement of a selected quality factor.

Amoui et al. [5] proposed a search-based evolutionary method, particularly Genetic Algorithm, to find a best sequence of valid high level design pattern transformations to improve software reusability. Object-oriented software metrics were used for assessing software quality. They suggested that different metrics had been appropriate for different applications. However, they did not consider transforming of an existing code to design patterns.

Jensen and Cheng [6] applied genetic programing to automate the use of software engineering metrics to generate refactoring strategies that introduce design patterns. The QMOOD metric suit [14] helped to determine the optimal set of refactorings.

Shimomura et al. applied another approach to recommendation of a design pattern [7]. Pairs of programs with and without a design pattern were prepared. A set of software metrics were calculated for these programs, treating those with design patterns as “good” solutions. An impact of a particular metric to a program quality was determined with a genetic algorithm and expressed as a weight coefficient. Comparison of weighted metric outcomes suggested whether a program should be refactored and which of design patterns should be applied. However, it was observed that differences in metrics between programs were not very big.

Several solutions have been concentrated on different selected design patterns, and therefore, they could be adjusted for code prerequisites to the more extend.

Refactoring to the Null Object design pattern was thoroughly examined by Gaitani et al. [9]. The analysis focused on special cases of null-checking conditionals that could be effectively refactored to Null Object. In a broad meaning, it is similar to our approach, although it has specialized on a different design pattern and would be not easily extensible to others. A logic programming based on use of Prolog-like predicates was applied in refactoring to Abstract Factory [10].

The Strategy design pattern was a target of automated refactoring reported in [11]. An algorithm was proposed to identify conditional statements that emulate this design pattern. In special cases of conditional statements, a technique for total replacement of conditional logic with method calls of appropriate concrete Strategy instances was suggested.

3 Refactoring Process to Design Patterns

Our aim is to automate a process of introducing design patterns to a code of an existing program. In order to efficiently use a design pattern a developer should understand a general structure of the pattern, principles of communication between objects within the pattern and between its context objects, possible benefits and disadvantages following its application. However, all decisions could also be made automatically to save up developer time and/or support less experienced programmers.

A whole process of a design pattern insertion consists of three main phases:

1. code analysis and calculation of relevance metrics,
2. determination of a refactoring range,
3. realization of a code restructuring - i.e. an adequate code transformation.

A code is analyzed in order to decide which refactoring for particular classes can be applied. Basic information about a class structure is derived from a syntax tree. Information to be collected depends on an anticipated design pattern. Different relations between code elements are identified that are important for a kind and range of the presumed refactoring. During the analysis several conditions and coefficients are determined according to selected quantitative parameters.

There are many issues considered in refactorings to design patterns, for example:

- static values assigned to class fields, which are often used to control an application,
- complex conditional instructions, which make a code incomprehensible,
- a class comprising a lot of functionality, which can result in complex methods and hiding an actual purpose of the class,
- a selection of an appropriate class constructor, which is especially important when a class has polymorphic fields.

Based on the collected data, all elements that point out at the refactoring application are recognized. Those design patterns are selected that might be introduced at the place of considered elements.

For each considered design pattern, a *relevance metric* is calculated. The metric should reflect the ability of the code to be transformed to a selected pattern. It takes positive values in the range from 0 to 100%. It is designed especially for a given design pattern. Different factors influence decisions in code refactoring, and selection of elements to be transformed. We can express an impact of selected factors in a quantitative way. For example, a final value depends on two factors, one in λ_1% and another in λ_2%, where $\lambda_1 + \lambda_2 = 100\%$. Coefficients are determined as standard values before the refactoring process, or could be defined by a user, if demanded.

In general, while specifying the first phase of the process the following issues should be given: (i) preconditions that should be satisfied by selected elements of a class under consideration in order to complete the refactoring, (ii) an algorithm how to calculate a relevance metric.

The second phase has the following goals: to take a decision which design pattern (if any) will be built into the code, to determine which parts of the pattern will be incorporated, and which parts of the code will be transformed. After a design pattern is

selected the reasoning about its features can be automated or a user can guide further decisions.

During the last phase, the application structure is modified in a way that has been decided during the previous phase. Specification of each refactoring comprises recognition of elements that are added, modified, and deleted during introduction of a new design pattern. Detailed actions realized in the process of the refactoring should also be defined.

A user can take their own decisions how to carry out the refactoring. It can be used when the user is familiar with the code and would like to adjust the range of changes to be performed in the code. In another situation the user would like to make a mock refactoring. Proposed changes are visible in the code after the refactoring, and then the user can take the final decision whether accept them or not.

Application of a design pattern can also lead, for example, to an unnecessary increase in a program complexity due to a high number of classes with small methods, as in the State design pattern. Therefore, it is important to support an "undo" facility that in an easy way could restore a code after a "big" refactoring.

4 Replace State Altering Conditional with State (RSACS)

Various transformations of code to design pattern described in [3] have been considered as subjects of the above general process. We have designed the detailed processes and proposed relevance metrics for the following refactorings: *Replace Type Code with Class* (*RTCC*), *Replace State-Altering Conditions with State* (*RSACS*), *Replace Conditional Logic with Strategy* (*RCLS*) and *Replace Constructors with Creation Methods* (*RCCM*). As an example, we discuss the second one (*RSACS*), but its realization could be proceeded by (*RTCC*). The target State design pattern is illustrated in Fig. 1.

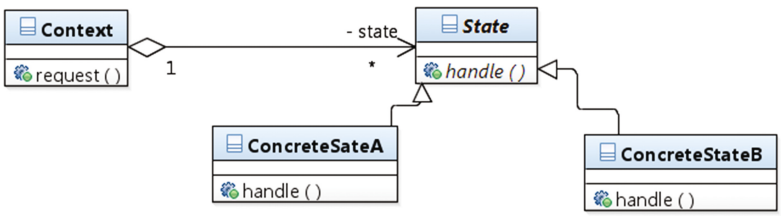


Fig. 1. Class diagram of the *State* design pattern

4.1 Code Analysis of the Refactoring

Let us consider class *X* pointed at by a user or selected automatically. We specify how an analysis is carried out of this class. Preconditions of the refactoring determine whether the class can be used as a source class in the transformation.

Preconditions. The design pattern will be considered for a class *X* only when the following conditions are met by its properties:

1. There exists a field f_X such as:
   (a) f_X can take at least k static values from a set $S(f_X)$, which may be finals or not,
   (b) values $S(f_X)$ are of a standard primitive type (e.g. in Java int, char, string, etc.),
   (c) the field can take only those values, or values of corresponding fields of other instances of class X.
2. There exists a set $M_X(y_X)$ of methods m_X under the following constraints:
   (a) method m_X includes at least one *P-type* statement *due to a field* y_X. A statement is of *P-type due to a field* y_X if it is a conditional instruction (e.g. *if-else*) and has a conditional predicate which is determined by the field y_X taking a value from $S(y_X)$, where y_X belongs to fields that satisfy the first precondition,
   (b) set $M_X(y_X)$ contains at least k methods m_X including *P-type* statements due to the same field y_X, where k is a refactoring parameter.

In case of RSACS, we found the refactoring parameter $k = 3$ as a reasonable bottom boundary. It means that there are at least three different cases corresponding to at least three concrete substates in the State pattern (Fig. 1).

Next, sets of methods $M_X(y_X)$ of class X are examined. Any such method includes one or more *P-type* statements, which are checked whether are suitable for creating a new method anticipated in the State design pattern. Two situations are distinguished in accordance to the number of local variables modified in a *P-type* statement:

- One local variable is modified at most. This change can be easily realized by taking the variable as a parameter and returning its new value at the method end.
- More than one local variable is modified. This statement is disregarded.

Metric Calculation. A relevance metric should help in assessing usefulness of transformation of appropriate parts of class X into the State design pattern. The metric can be calculated for any class X that has field f_X satisfying the first precondition and a set of methods $M_X(f_X)$ fulfilling the second precondition. At this stage it is possible, although rarely, to have in class X many fields f_X with their sets of methods $M_X(f_X)$.

The metric for RSACS is a sum of two weighed components that relate to the impact of a quantitative factor (based on numbers of elements) and a profitability factor (suitability of the pattern to a code). In case studies, we assumed the range values equal to 25 and 75% for the factors, accordingly. We assess a class X with a metric above 50% to be reasonable to consider for refactoring. The metric is also used for comparison of different fields in a class, or comparison of different classes. The detailed discussion of the metric is beyond the scope of the paper [25].

4.2 Determination of the Refactoring Range

A way of taking decisions about a refactoring was an important factor that simplify automating of the second phase of the process. In *RSACS* refactoring this decision is based on features of fields located in one class. A code located in other classes has some influence but only one class is a basic class used in analysis and reasoning. This refactoring depends also strongly on appearing of conditional instructions (*if-else*) in methods and constructors of the basic class X.

We will denote by C_i concrete subclasses that inherit from the base *State* class in the State design pattern (Fig. 1). The following decisions have to be taken:

- If *RTCC* is combined with *RSACS*, it has to be selected: which field of X is a given f_X to drive the refactoring in case many fields satisfy the preconditions (in most cases there is only one field); and shall we delete static fields in X that represent $S(f_X)$ values and are not more used after the refactoring.
- Are concrete subclasses C_i created for all possible cases, or only for indispensable ones?
- Which of class X will be modified?
- How the selected methods m_X will be modified? A whole method can be moved to the *State* class or appropriate small methods are extracted from m_X and moved to *State*. The extracted methods are based on the recognized *P-type* statements.

If the process is not driven by a user, above decisions and others are taken automatically. For each method m_X its *P-type* statements are examined. A general goal is to find extracts of the method that could be used for creation of new methods in the State pattern. A selected statement could have a simple logical predicate to be directly transformed. Otherwise; closures could be duplicated in various conditional statements and cover a limited part of the whole method.

Other decisions concern naming conventions for newly created artifacts, such as C_i classes and their methods.

4.3 Code Transformation

Realization of *RSACS* is often preceded by *RTCC*, which can be automatically included in the realization phase. Many code elements are modified or refined during the transformation. The base *State* class is created. It includes a general method that is either copied from the class X and has an object of X as an additional argument, or it is built from a closure. In the latter case its arguments are determined based on local variables used in a method m_X. If local fields of X are used in the closure, an X object also belongs to the arguments of the general method.

New C_i classes are created with their argumentless constructors that assign appropriate $S(f_X)$ values. Moreover, methods of C_i are created. Particular methods of C_i have the code specific to a given subtype. All references to elements from X are converted into a reference to an X object passed as a parameter. If there is a method based on a closure in which a local variable is modified and the variable is used in the method, the method returns a value of the modified variable.

In the class X a general method from *State* is called in the context of a field f_X. A body of the old method from X will be substituted by the new method call if the whole method has been moved to the *State* class. Otherwise, i.e. the method in *State* has been created from closures, the appropriate closures in methods m_X are substituted by a call of the new general method.

Finally, appropriate import closures are added in files with declarations of *State* and C_i classes.

5 Framework

Ideas presented in the previous sections were implemented in a framework that extends the Eclipse environment [21] and supports the Java language.

5.1 Design

In Eclipse above 20 refactoring operations are provided for a Java developer. They can be used via the GUI or by calling appropriate methods from the API. The available transformations correspond mainly to basic refactorings [2].

It was intended to make use of some simple refactorings supported in Eclipse. However, a primary requirement of the tool was to keep the whole complex refactoring reversible. A user should have a possibility to resign at any step of the refactoring process, and undo the entire refactoring recently having been completed. In order to have full control over each refactoring step, refactorings are implemented directly in the tool and a whole selected transformation can be reversed.

While building the tool, several facilities provided by Eclipse were applied [21]. An infrastructure for building the own refactoring and integrating with the Eclipse environment is the *Refactoring Framework*. It assisted in viewing code modifications before are integrated into the final code, adding an applied refactoring to the refactoring history, creating a user interface, etc. Developed plug-in and simple refactoring actions existing in Eclipse have identical kind of interface.

An application code can be viewed in a form of the *Abstract Syntax Tree* (AST). Each program file is represented by its tree. The hierarchy and links between the nodes reflect relations between corresponding code elements, e.g. inclusion a method in a class, a class in a class. Working on a tree was employed during the code analysis phase and the code manipulation.

The implemented framework consists of two parts: Complex Refactoring Framework (CRF) and Automated Refactoring to Patterns (APRT). CRF is based on the standard Refactoring Framework, but it additionally provides means for analysis and decision steps. It also specifies a framework for GUI pages responsible for user interaction. APRT realizes final Java code transformations.

The whole framework can be easily extended with other refactorings following the described process and applying appropriate relevance metrics.

5.2 Case Studies

The prototype implements two kinds of refactorings: RTCC and RSACS. They were used in case studies on small programs ($\sim$250 LOC). Program functionality was tested with test suites before and after refactorings. All transformations could also have been successfully withdrawn. All process phases were verified in experiments. A manual refactoring was compared with an automatic one. We checked whether all metric components were correctly calculated, a selected element had the highest metric value, and all required elements were taken into account in transformations. Moreover, the employed transformations matched to non-automatic decisions. The results of

relevance metrics were compared to user expectations expressed quantitatively. The values were similar in case of RTCC, and differ in 2% in case of RSACS.

The preliminary experiments appear to confirm usability of the approach, although many threats to validity have to be considered. More complex, third party developed applications, are expected to give less promising results. Moreover, the metric coefficients that were tuned after examination of a range of programs are still a question of concern. However, it should be noted that, as in many metric-driven approaches, the direct metric values are not of the most importance. The decisions are mainly based on relative metric values of different elements that can be compared and the same coefficients are used for those elements during calculations.

6 Conclusions

In this paper, we presented a process of refactoring of software code to design patterns that can be automated or driven by a program developer. In opposite to other approaches, it focuses on assessment of the code relevance to a selected design pattern. Future plans refer to design and implementation of refactorings to other design patterns [3] as well as experimental evaluation with bigger real-world applications. Further research would be required on calibration of relevance metric and on other metric variants [25]. Combining of the process with evaluation of software quality with other metrics could also be beneficial.

Acknowledgments. The author thanks their student P. Saran for implementation of the plugins and evaluation of case studies.

References

1. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston (1995)
2. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: Refactoring: Improving the Design of Existing Code. Addison Wesley, Boston (1999)
3. Kierevsky, J.: Refactoring to Patterns. Addison Wesley, Boston (2004)
4. Mariani, T., Vergilio, S.R.: A systematic review on search-based refactoring. Inf. Softw. Technol. **83**, 14–34 (2017)
5. Amoui, M., Mirarab, S., Ansari, S., Lucas, C.: A genetic algorithm approach to design evolution using design pattern transformation. Int. J. Inf. Technol. Intell. Comput. **1**, 235–244 (2006)
6. Jensen, A.C., Cheng, B.H.: On the use of genetic programming for automated refactoring and the introduction of design patterns. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 1341–1348. ACM (2010)
7. Shimomura, T., Ikeda, K., Takahashi, M.: An approach to GA-driven automatic refactoring based on design patterns. In: 5th International Conference on Software Engineering Advances (ICSEA), pp. 213–218 (2010)

8. Kim, J., Batory, D., Dig, D.: Scripting parametric refactorings in Java to retrofit design patterns. In: 31st IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 211–220. IEEE (2015)
9. Gaitani, M.A.G., Zafeiris, V.E., Diamantidis, N.A., Giakoumakis, E.A.: Automated refactoring to the null object design pattern. Inf. Softw. Technol. **59**, 33–52 (2015)
10. Jeon, S.-U., Lee, J.-S. Bae, D.-H: An automated refactoring approach to design pattern-based program transformations in Java programs. In: Proceedings of Ninth Asia-Pacific Software Engineering Conference (APSEC 2002), pp. 337–345. IEEE Computer Society (2002)
11. Christopoulou, A., Giakoumakis, E., Zafeiris, V.E., Soukara, V.: Automated refactoring to the strategy design pattern. Inf. Softw. Technol. **54**, 1202–1214 (2012)
12. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Trans. Softw. Eng. **20**, 476–493 (1994)
13. Kan, S.H.: Metrics and Models in Software Quality Engineering. Addison-Wesley, Boston (1998)
14. Bansiya, J., Davis, C.G.: A hierarchical model for object-oriented design quality assessment. IEEE Trans. Softw. Eng. **28**(1), 4–17 (2002)
15. Tsantalis, N., Chatzigeorgiou, A.: Identification of move method refactoring opportunities. IEEE Trans. Softw. Eng. **35**(3), 347–367 (2009)
16. Silva, D., Terra, R., Valente, M.T.: Recommending automated extract method refactorings. In: 22nd International Conference on Program Comprehension (ICPC), pp. 146–156. ACM, New York (2014)
17. Bavota, G., Lucia, A.D., Marcus, A., Oliveto, R.: Recommending refactoring operations in large software systems. In: Robillard, M.P., Maalej, W., Walker, R.J., Zimmermann, T. (eds.) Recommendation Systems in Software Engineering (RSSE), pp. 387–419. Springer, Berlin (2014)
18. Vakilian, M., Chen, N., Negara, S., Rajkumar, B.A., Bailey, B.P., Johnson, R.E.: Use, disuse, and misuse of automated refactorings. In: 34th International Conference on Software Engineering (ICSE), pp. 233–243 (2012)
19. Tokuda, L., Batory, D.S.: Evolving object-oriented designs with refactorings. Autom. Softw. Eng. **8**(1), 89–120 (2001)
20. Sommerville, I.: Software Engineering, 10th edn. Pearson Education, New York (2015)
21. Eclipse - an open development platform. http://www.eclipse.org
22. Cinneide, M.O.: Automated application of design patterns: a refactoring approach. Ph.D. thesis, University of Dublin, Trinity College (2001)
23. Mayvan, B.B., Rasoolzadegan, A., Yazdi, Z.G.: The state of the art on design patterns: a systematic mapping of the literature. J. Syst. Softw. **125**, 93–118 (2017)
24. Abbe, M., Yoo, C.-J.: Trends, opportunities and challenges of software refactoring: a systematic literature review. Int. J. Softw. Eng. Appl. **8**(6), 299–318 (2014)
25. Derezinska, A.: Metrics in software development and evolution with design patterns (to appear)

Modeling Autoreferential Relationships in Association-Oriented Database Metamodel

Marek Krótkiewicz[1] and Marcin Jodłowiec[2(✉)]

[1] Department of Information Systems,
Wroclaw University of Science and Technology, Wroclaw, Poland
[2] Institute of Computer Science, Opole University of Technology, Opole, Poland
m.jodlowiec@kieg.org

Abstract. The article provides a concise discussion of autoreference database design pattern (recursive relationship) which allows elements to directly reference other elements with the same type. The paper introduces modeling of autoreferential relationships in Association-Oriented Database Metamodel in regard to its semantic capabilities, i.e. freedom of defining multiplicity on both sides of relationships, connecting associations with each other and composition providing lifetime dependency.

Keywords: Database design patterns · Database modeling · Association-Oriented Database Metamodel · AODB · Autoreferential relationships · Recursive relationships

1 Introduction

Autoreferential relationship is a popular design pattern, often used in database modeling [2,5]. Its characteristics is the recursivity of relationship, i.e. situation in which modeled element holds the reference to itself [4]. Autoreference can be considered in the intensional and extensional manner. Conceptually, in terms of intensional database part, structures containing autoreferential relationships may be designated as entities connected with other entities which are of the same type as shown on the left side of Fig. 1. Whereas considering them extensionally, they define entities having references to themselves (right side). The extensional autoreference aspect is a special case of autoreferential structure and is related to actual data described within intensional autoreference.

The most general and at the same time the most representative autoreference structures are graphs. The specific case of graphs can be trees, while their generalization are hypergraphs. On the basis of listed data structures one can create abundance of graph-based solutions, e.g. binary trees, k-partite graphs, labeled graphs and other, more complex structures comprising compound of two or more elementary design patterns based on aforementioned concepts or other data representation methods. One should pay attention to the fact, that autoreferential structure is not the only way to model graphs, but each autoreferential structure might be interpreted as a graph.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_5

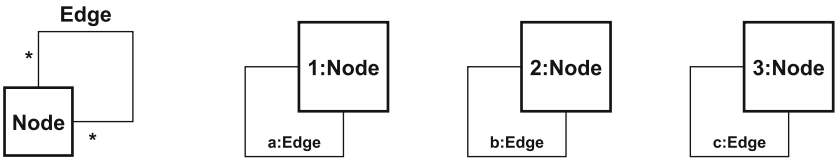


Fig. 1. Conceptual autoreferential relationships in a graph – intensional (left) and extentional (right) autoreference

In this respect, the key concept for current applications seems to be optimization of elementary database operations of graph traversal. The complexity of such operations depends mostly on selection of database metamodel and structure representing a graph. The database metamodel selection unambiguously determines the nature of relationships between modeled entities. These links may be behavioral, e.g. in relational metamodel (RDB) or structural, e.g. in OM ODMG 3.0 [3], document-oriented databases, RDF-based graph-oriented databases or databases in Association-Oriented Metamodel (AODB).

The subject of the following study is the issue of modeling of self-referencing relationships in databases. The motivation was to describe association-oriented databases modeling methodology in terms of recursive relationships. It was assumed that the separation of the data containers and relationships containers, which is fundamental property of the AODB metamodel, will significantly improve semantic capacity of autoreference structures. The AODB metamodel has been proposed by Krótkiewicz in [9] and its usefulness for modeling complex knowledge representation systems has been elaborated in [10]. The aim of this paper is to present the modeling capacity of AODB in regard to autoreferential relationships.

The Sect. 2 presents implementation of recursive structures in terms of known and popular database metamodels regarding their grammatical and semantic limitations. The main contribution of this paper has been described in Sect. 3, i.e. method of modeling and features of autoreferential structures in AODB. Next, in Sect. 4 authors evaluate presented solutions by comparing modeling features of the patterns in different metamodels. The last sections are the conclusions.

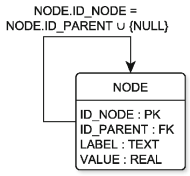
2 Autoreference in Popular Database Metamodels

The following section presents the survey of autoreferential (or recursive) relationship realized in various database metamodels. The study presents models, which implement this pattern on conceptual case study of modeling some classes of graphs. The most important is that depending on expressiveness of database metamodel, i.e. possibilities of modeling, which it posses, recursive structures will vary. The analyzed examples consider only *intensional autoreference*. That means such autoreference, which is directly supported by structures possible to be modeled within the specific metamodel. Obviously, one could try to model autoreferential relationships on the higher level of abstraction using expanded

models, however it would only leed to creation of some more complex design patterns, which conceptually represent autoreference. However, physically they would comprise of several different relationships and force introducing new, auxiliary model categories on the implementation level.

2.1 Relational Database Metamodel

The implementation of autoreference relationship in relational database metamodel (RDB) has been presented on Fig. 2. In relational metamodel relationships are stored in the database with the use of key mechanism (primary key and foreign key) together with referential integrity constraints.



ID_NODE	ID_PARENT	LABEL	VALUE
1	NULL	"N1"	1.0
2	1	"N2"	0.15
3	1	"N3"	0.2
4	2	"N4"	NULL
...	...	...	...

Fig. 2. Relational diagram (left) and relation of sample autoreferential model in relational database metamodel (right)

In case of autoreferential structures in RDB, one should be aware of a very important fact: only *one-to-many* and *one-to-one* relationships can be physically represented. It means, that in relational database metamodel *many-to-many* relationships do not exist inherently, but they can be built using well-known design patterns, e.g. intersection table. The following paper considers the issue of occurrence or absence of certain mechanisms, in contrast to the possibility to be constructed.

It is reflected in database structures, which can be stored in model shown in Fig. 2. Under the assumption of cycle detection in the layer of system logic (or specific database management system) this structure is suited to store trees, which model *parent-child* structures, i.e. hierarchies. Sample data for this structure have been shown on the right part of the Fig. 2.

2.2 XML Data Model

The XML language can sometimes be used as data model in database systems [1,8,12]. XML documents have tree structure, which is based on tags. Such a structure forces hierarchical shape of model. The XML documents can posses its own meta-structure, which can be described within XML schema or DTD (document type definition). Therefore, the structure can be treated as a metamodel in relation to the built XML documents. XML gives possibility of very natural modeling of tree-based autoreferential structures. The Listing 1.1 shows XML schema for graph.

Listing 1.1. XML Schema for exemplary autoreference hierarchy

```
<?xml version="1.0"encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="node"type="nodeType/>
  </xs:sequence>
  <xs:complexType name="nodeType">
    <xs:element name="label"
      type="xs:string"/>
    <xs:element name="value"
      type="xs:decimal"/>
    <xs:sequence minOccurs="0"
      maxOccurs="unbounded">
      <xs:element name="node"
        type="nodeType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Listing 1.2. Exemplary XML document based on schema on Listing 1.1

```
<?xml version="1.0"encoding="UTF-8"?>
<node>
  <label>N1</label>
  <value>1.0</value>
  <node>
    <label>N2</label>
    <value>0.15</value>
    <node>
      <label>N4</label>
    </node>
  </node>
  <node>
    <label>N3</label>
    <value>0.2</value>
  </node>
</node>
```

Listing 1.3. An exemplary XML Schema for *many-to-many* multiplicity autoreferential relationship

```
<?xml version="1.0"encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="graph">
    <xs:complexType>
      <xs:sequence minOccurs="0"
        maxOccurs="unbounded">
        <xs:element name="node"type="nodeType/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="nodeType">
    <xs:element name="nodeID"type="xs:ID"/>
    <xs:element name="relationships"
      type="xs:IDREFS"/>
  </xs:complexType>
</xs:schema>
```

Listing 1.4. Exemplary XML document based on schema on Listing 1.3

```
<?xml version="1.0"encoding="UTF-8"?>
<graph>
  <node>
    <nodeID>1</nodeID>
    <relationships>2 3</relationships>
  <node/>
  <node>
    <nodeID>2</nodeID>
    <relationships>3</relationships>
  <node/>
  <node>
    <nodeID>3</nodeID>
    <relationships>3</relationships>
  <node/>
  <node>
    <nodeID>4</nodeID>
    <relationships>1 2 3</relationships>
  <node/>
</graph>
```

The autoreferential relationship has been modeled very naturally, as set of nested nodes, which have the same tag type (`node`). The exemplary XML document which is described within presented schema has been shown on Listing 1.2. Hierarchical structure of this language seems to be well-suited to model tree-based structures.

XML also enables defining *many-to-many* autoreferential relationships. This can be achieved using the `xs:ID` and `xs:IDREFS` data types, which are definition of references to unique identifiers. The XML code Listings 1.3 and 1.4 show exemplary model of graph.

2.3 Object-Oriented Approach

Currently, the only official standard in regard to object-oriented databases is Object Model ODMG 3.0 [3]. Although there is no approved and popular implementation of this metamodel, which would fulfill all of its requirements in terms of this standard, it still is the reference document for this type of database metamodels.

Using the Object Model gives possibility to model relationships with the following multiplicities: *one-to-one*, *one-to-many*, *many-to-many*. In the terms of autoreferential relationships it is a huge step forward. Beyond the simple hierarchical relationships, it is possible to build typical relationships, which represent directed graphs with no cardinality constrains, which are enforced by tree-like structures. The implementation of relationship is in the responsibility of the database designer: it can be modeled by reference to one object and collection of references to objects (Listing 1.5) or two collections of references to objects (Listing 1.6).

Listing 1.5. ODL for *one-to-many* autoreference

```
class Node {
  attribute double value;
  attribute string label;
  relationship Node parent
  inverse Node::children;
  relationship set<Node> children
  inverse Node::parent;
}
```

Listing 1.6. ODL for *many-to-many* autoreference

```
class Node {
  attribute double value;
  attribute string label;
  relationship set<Node> from
  inverse Node::to;
  relationship set<Node> to
  inverse Node::from;
}
```

The extensional part of database modeled within the structure of node have been shown in the Fig. 3. On the left side of the figure is the object diagram for the structure from the Listing 1.5 and the diagram on the right side stands for the *many-to-many* relationship (Listing 1.6) respectively.

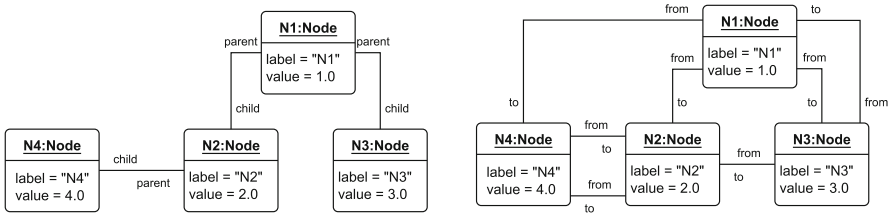


Fig. 3. *UML*® object diagrams representing sample object-oriented data

3 Proposed Solution of Autoreferential Relationships Modeling in AODB

The following section shows the main paper's contribution, namely describes semantic capability of autoreferential relationship in AODB. The metamodel

has been briefly described in the first part of this section. AODB allows to build connections with two relationships (associations) using single role. In the case, when both the destination of role and its source is the same association, one can call it autoreferential relationship. Due to this feature of the metamodel (combining associations with each other) the notion of autoreference differs from other database metamodels, where recursivity was observed on the level of modeled entity category (relation, object, XML node, etc.), not the relationship level.

The important thing is that according to the key postulate of the metamodel, which is separation between data and relationships, modeled entities (in example used in this paper: nodes) are divided into two parts, representing their specific aspects: data storage and connecting with other entities. The two parts of node are associated by the bicompositional role with *one-to-one* multiplicity. This design pattern is crucial in terms of modeling functionally complex entities (storing data with attributes and being in relationships with other model categories).

In the presented models, autoreferential role Edge represents relationships between the nodes. There are presented the cases of graph structure configuration taking into consideration such AODB role features as: multiplicities, uniqueness, compositionality and navigability. The study also contains AODB extensional diagrams which represent data described within modeled structures. In sake of simplification, the extensional diagrams show only the relationship part of database, i.e. association objects and corresponding role objects.

3.1 Brief Description of AODB

AODB Metamodel Primitives. The mentioned below belong to the main primitives of AODB: intensional primitives: database (*Db*), association (*Assoc*), collection (*Coll*), role (*Role*), attribute (*Attr*), and extensional primitives: association object (*AssocObj*), object (*Obj*), role object (*RoleObj*).

Database contains two lists: the list of collections and associations respectively. Intensionally, association contains roles, which it owns, and extensionally association is the set of association objects. Additionally, it may also contain reference to the collection, which describes it, allowing to supplement relationship with data. Intensionally the collection is the set of attributes, and extensionally it is the set of objects. Moreover, both the collection and the association have name, information about abstractness, navigability and inheritance. Role contains name, information about virtuality, inheritability, navigability, directionality, multiplicity and compositionality (the mechanism of lifetime dependency), uniqueness, describing collection and the type of element, which it can link on the side of role destination. On the side of role source, it is anchored to association, which is its owner. The attribute contains name, information about virtuality, inheritability, cardinality, type of elements and default value. Object is built of values, moreover it contains automatically assigned, unique and time-invariant identifier. Association object contains a set of role objects and reference to object, which it describes as well as an unique identifier. Role object contains

set of references to the extensional elements of database (objects or association objects) and reference to describing object.

Association-Oriented Modeling Language. This section addresses the most important aspects of grammar and semantics of selected part of Association-Modeling Language (AML). It is a graphical language used to design database schemata in AODB [7,9,11]. The graphical representation has been provided in Fig. 4. The following description maps graphical symbols to the presented AODB primitives: (1) *Association* corresponds to a semantic category (*Assoc*). (2) *Collection* corresponds to a semantic category (*Coll*). (3) *Role* corresponds to a semantic category (*Role*). The graphical form in AML depends on navigability, directionality and composability. (4) *Role Ownership*, (5) *Navigability*, (6) *Composition*, (7) *Multiplicity*, (8) *Uniqueness* are parts of the graphical form of a role (*Role*). (9) *Attribute* is a part of the graphical form o a collection (*Coll*). Attribute (*Attr*) has a name, scope of visibility, quantity, type and default value. (10) *Attribute Type* is a part of the graphical form of an attribute (*Attr*).

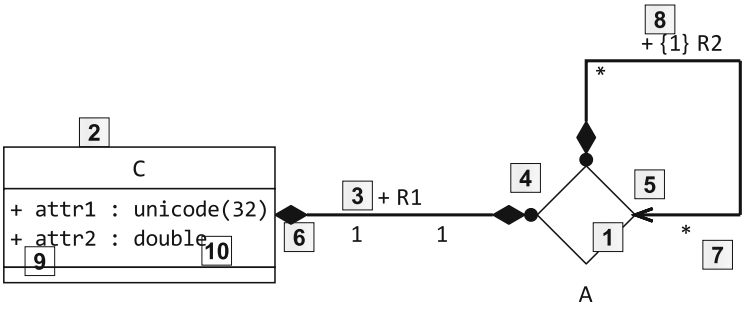


Fig. 4. Sample database schema diagram in AODB

3.2 Multiplicities of Recursive Relationships

The association-oriented modeling allows the designer to accomplish very complex definition of relationships, i.e. in terms of full latitude of their arity and multiplicity. The role multiplicity in AODB can be defined both on the side of the relationship and the related element. This has also its application when autoreferential relationships are designed.

The application of the multiplicity *many* on both sides of the role shown in Fig. 5 makes it possible to construct graphs without degree restrictions. The sample database relationship graph built on this structure has been shown on the right side of the figure.

The application of the multiplicity *one* on the side of role owner and *many* on the side of role destination helps to force a structural dependency for the number of elements, which corresponds to the terms of multiplicity. It gives

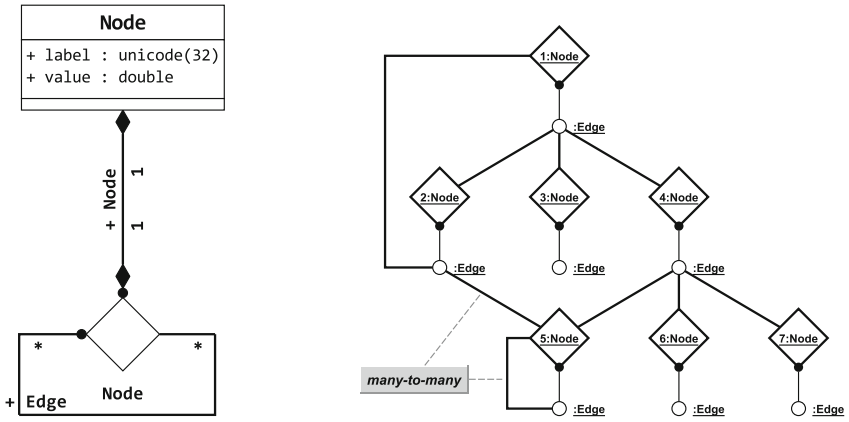


Fig. 5. AML intensional (left) and extensional (right) diagrams of model with autoreferential role with *many-to-many* multiplicity

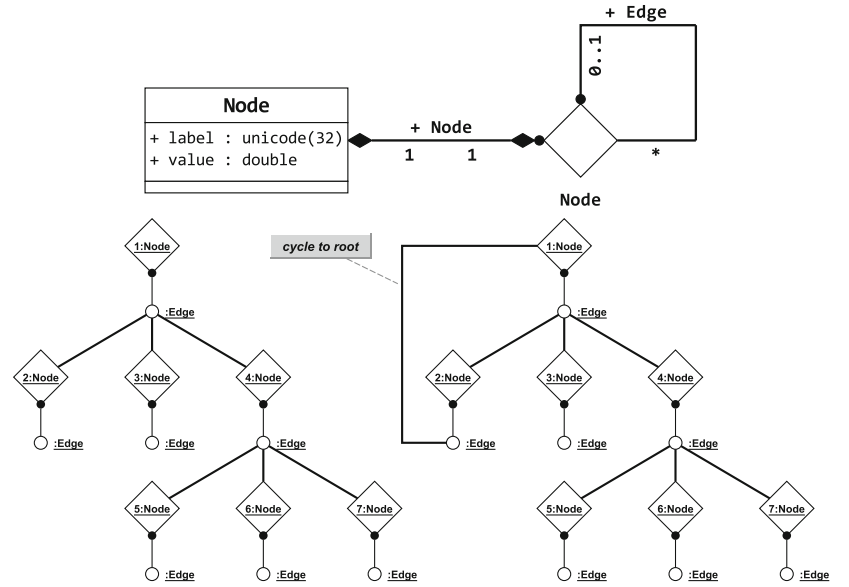


Fig. 6. AML intensional (up) and extensional (down) diagrams of model with autoreferential role with *one-to-many* multiplicity

the possibility to design tree-like hierarchies. However, one has to notice, that this solution cannot disable cycles in the structure model level, thus controlling whether the model will actually store a tree has to be performed by the system logic. The Fig. 6 shows a AML diagram of the sample *one-to-many* structure and two hypergraphs described within it. The left one does not have any cycles and in regard to given structural multiplicity constraints represents a tree structure.

The hypergraph on the right side presents situation, where association object 2:Node forms a cycle with "root" node, therefore breaking the tree constraint.

The freedom in the multiplicity modeling within recursivity can be achieved within object-oriented databases and since Association-Oriented Metamodel constitutes a development of object-oriented approach, this is possible in AODB, too. However, one shall notice the difference, that implementation of recursive relationship in OM ODMG 3.0 and $UML^{®}$ requires one relationship consisting of two roles connecting one class, and in AODB the recursivity can be achieved with only one role within one relationship. The RDB database metamodel does not support direct *many-to-many* relationships on structural level.

3.3 Autoreferences with Composition

AODB provides with possibility to model autoreferential structures which are lifetime dependent, by the use of the feature of role, namely composition. The lifetime dependency relationship relies on the fact, that when lifetime-binding element is deleted from database, all element bound by it are also deleted. The left part of the Fig. 7 shows a model of graph in AODB metamodel, in which composition has been applied on the side of the role owner.

It means, that when the Node association object becomes deleted, all connected to it *Node* association objects will be deleted as shown on the right side of Fig. 7 (marked gray). Likewise, if the composition is applied on the side of role destination as shown in Fig. 8, the lifetime dependency mechanism acts analogously, but in opposite direction. In AODB metamodel one can also model bicomposition, i.e. when the composition is applied on both sides of the role. Application of this type of role in autoreferential structure representing a graph would force deletion of the whole connected subgraph, in which deleted element appears.

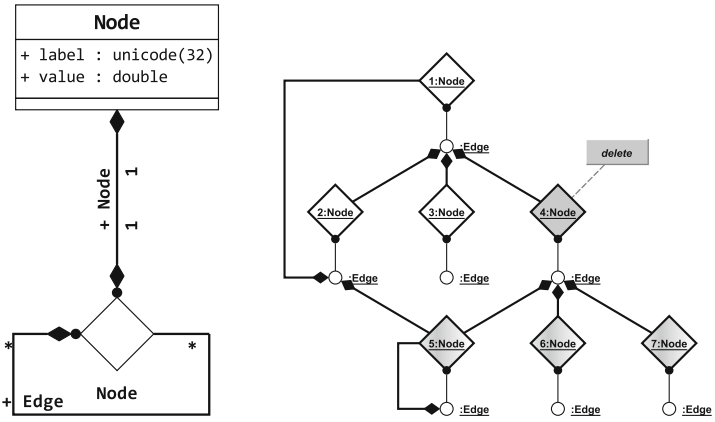


Fig. 7. AML intensional and extensional diagrams of model with autoreferential role with composition in its owner

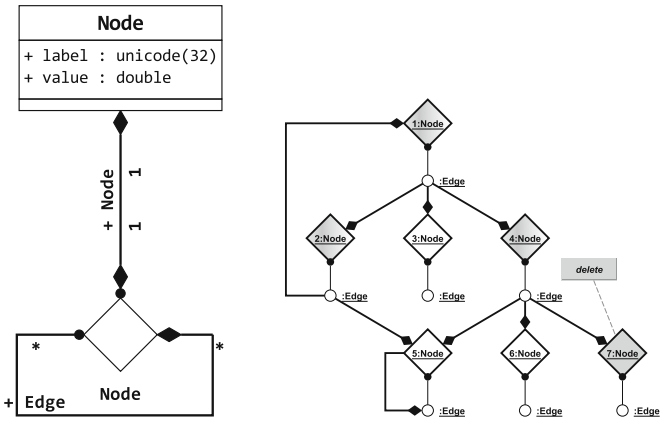


Fig. 8. AML intensional and extensional diagrams of model with autoreferential role with composition in its destination

Composition is AODB primitive unlike any other well-known database metamodels. One shall consider that the concept of strong aggregation (also known as composition) exists in UML®, however the standard of object-oriented databases does not apply any lifetime dependency customization of the relationship.

3.4 Autoreferences with Uninavigability and Uniqueness

The relationships in AODB might be also filled with information regarding uniqueness and navigability as shown in the Fig. 9. Setting the uniqueness value of the Edge role to 1 ensures, that the model of autoreferential structure cannot store multigraphs, that is such graphs, in which exists more than one relationship between two exactly the same nodes. It is worth to notice, that with AODB it is easily to model k-constrained multigraphs by setting the uniqueness value of this role to k. In the effect the graph will have maximum k direct connections with each pair of connected association objects representing nodes.

Uniqueness is AODB primitive. Beyond this metamodel, only RDB supports of modeling uniqueness within relationships. One can notice, that relationships in RDB are modeled using attributes, primary and foreign key mechanisms and referential integrity constraints. Since attributes may be unique, thus relationships can as well.

The other feature of roles in AODB is their navigability, i.e. physical ability to perform direct traversal from one node to the other. The uninavigability is the property of relationship and occurs, when given element contains direct reference to other element (or to itself), while the direct reference in opposite direction is absent. By the term direct reference, authors understand the information which unambiguously points to the specific element. Modeling uninavigable structures makes possible to save disk space, if the fraction of reality provides with information, that the database search within this relationship will be unidirectional.

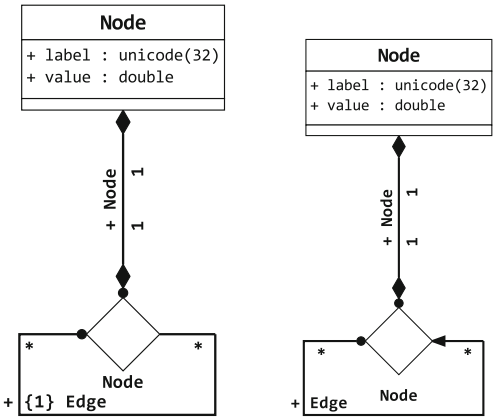


Fig. 9. AML intensional diagrams of models with autoreferential role with uniqueness (left) and uninavigability (right)

It is substantial, when Big Data applications are concerned, as each reference which is physically stored provides the storage overhead. Obviously, a consequence of the aforementioned is the fact, that searching the database against the direction of uninavigable relationship will be vastly less efficient, as the finding of related elements will require searching the whole collection or association. In the case of modeling autoreferential structures, which are directed graphs, if one can unambiguously designate this direction, application of uninavigability will be beneficial for the modeled structure.

4 Comparison of Modeling Features

During analysis of the database metamodels in terms of recursive relationships it was assumed that the main criterion of their evaluation will be their semantic capacity. By the term of semantic capacity, authors understand the richness of semantic features that can be used to enrich the model with additional information regarding relationship, i.e. multiplicity, uniqueness, ability to constrain multiplicity on the relationship side, lifetime dependency, navigability.

Comparison of selected modeling features has been presented in Table 1. Except for RDB, the multiplicity of recursive relationships seems not being problematic, but due to high importance of RDB in database world it still needs to be taken into account. Lack of direct *many-to-many* multiplicity is large drawback of this metamodel, which has been discussed before and shown in Fig. 2. The navigability is the very important concept in terms of trade-off between performance and data storage. Each binavigable relationship requires to store twice more information than uninavigable, thus defining navigability should be intentional. Only AODB is aware of this issue, what has been presented in Fig. 9. Binavigability in OM has been implemented as obligatory definition of traversal

Table 1. Comparison of analyzed modeling features in discussed metamodels

	Multiplicity	Lifetime dependency	Navigability	Uniqueness	Constraining multiplicity on the relationship side
RDB	*One-to-one*, *one-to-many*	No	Always uninavigable	Yes	No
XML	Any	No	Dependent on implementation	Yes	No
OM	Any	No	Always binavigable	No	No
AODB	Any	Yes, on both sides of the relationship	Uninavigable or binavigable	Yes	Yes

paths, that enable applications to use the logical connections between the objects participating in relationships. The RDB metamodel has the possibility to define uniqueness both in the relational algebra and in the specific implemented tools, such as SQL language. XML metamodel features several mechanisms to handle uniqueness, e.g. `xsd:IDREFS` datatype or `xs:unique` element. OM ODMG 3.0 does not support this feature. The metamodels RDB, XML, OM does not define any mechanisms in respect to lifetime dependency. In practice, they are implemented within the database management systems. In contrast to conventional database metamodels, AODB is also supplied with the possibility to constrain multiplicity on both sides of relationships, i.e. binding element and bound element side. In case of XML and RDB it is hard to consider data bounds as a "relationship" category, thus such constraints do not exist. In OM ODMG 3.0 the relationship is a separate category, but considered constraints are not provided.

5 Summary

In the paper the autoreferential relationship database design pattern has been considered. It has been presented in a number of metamodels with special consideration of novel database system named AODB. The pattern has been modeled with strong focus on structural constraints entailed by selection of specific database metamodel. It means, that all presented models contain physical (not conceptual) reference to the element with the same structure. The possibility of database structure modeling, and consequently data storage capability without loss of semantics, differ depending on the selected metamodel. It can be noticed in the example of such simple design pattern, as is autoreferential relationship. The greater is the expressiveness of metamodel, the bigger is the information capacity for physical implementation of considered design pattern.

Due to separation of data and relationships in AODB, mapping this construction to AODB does not create the direct recursive relationships, but regular association with two roles, which are fulfilled by the same collection. With the possibility of combining relationships with each other, the special case is to combine the association with itself using single role. Then, the same association is both owner and destination of this role. Therefore, autoreferences in AODB are

not typical, they are cascade relationships of the same type. It is very significant difference which has to be taken into account at the foundation of association modeling, where the entities are primarily considered in terms of its relationships and afterwards what data they can store.

This paper elaborates upon very simple and elementary design pattern, namely autoreferential relationship. One can say, that this structure is rudimentary, but the above considerations show, that even such simple structure can be subject of many customizations over the design process. Furthermore, existing database metamodels cannot efficiently handle the following: physical navigability along with multiplicity, uniqueness and lifetime dependency. The great power of expression is very important in modeling complex knowledge-based systems such as [6] or implementing storage for systems operating on very large collections of data. Authors see great potential in AODB Metamodel in terms of freedom in modeling autoreferential structures, as well as the more complex ones. Moreover, it seems obvious that attempt to map association-oriented schemata to other metamodels risks of the semantics loss, which is inconvenient with any approach to semantic modeling [13].

Future work regarding autoreferential structures in AODB will cover the analysis of query patterns to this structures in Association-Oriented Query Language (AQL) in terms of their complexity, performance and expressiveness in case of recursive queries.

References

1. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. The MIT Press, Cambridge (2004)
2. Brandon, D.: Recursive database structures. J. Comput. Sci. Coll. **21**(2), 295–304 (2005)
3. Cattell, R.G., Barry, D.K., Berler, M., Eastman, J., Jordan, D., Russell, C., Schadow, O., Stanienda, T., Velez, F.: The Object Data Standard: ODMG 3.0, p. 280 (2000)
4. Choi, M.Y., Moon, C.J., Baik, D.K., Wie, Y.J., Park, J.H.: The RDFS mapping for recursive relationship of relational data model. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–6. IEEE, December 2010
5. Connolly, T.M., Begg, C.E.: Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson Education, Upper Saddle River (2005)
6. Do, N.V.: Ontology COKB for knowledge representation and reasoning in designing knowledge-based systems. In: 13th International Conference on Intelligent Software Methodologies, Tools and Techniques, SoMeT 2014, Langkawi, Malaysia, 22–24 September 2014, Revised Selected Papers, pp. 101–118. Springer International Publishing, Cham (2015)
7. Jodłowiec, M., Krótkiewicz, M.: Semantics discovering in relational databases by pattern-based mapping to association-oriented metamodel a biomedical case study. In: Advances in Intelligent and Soft Computing (2016)
8. Krishnamurthy, R., Chakaravarthy, V.T., Kaushik, R., Naughton, J.F.: Recursive XML Schemas, Recursive XML Queries, and Relational Storage: XML-to-SQL Query Translation

9. Krótkiewicz, M.: Association-oriented database model n-ary associations. Int. J. Softw. Eng. Knowl. Eng. (2017)
10. Krótkiewicz, M., Wojtkiewicz, K.: Functional and structural integration without competence overstepping in structured semantic knowledge base system. J. Logic Lang. Inform. **23**(3), 331–345 (2014)
11. Krótkiewicz, M., Wojtkiewicz, K., Jodłowiec, M., Pokuta, W.: Semantic Knowledge Base: Quantifiers and Multiplicity in Extended Semantic Networks Module, pp. 173–187. Springer, Cham (2016)
12. Polyzotis, N., Garofalakis, M.: Statistical synopses for graph-structured XML databases. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, SIGMOD 2002, p. 358. ACM Press, New York, June 2002
13. Zaidi, H., Pollet, Y., Boufarès, F., Kraiem, N.: Semantic of data dependencies to improve the data quality. In: Proceedings of 5th International Conference on Model and Data Engineering, MEDI 2015, Rhodes, Greece, 26–28 September 2015, pp. 53–61. Springer International Publishing, Cham (2015)

Identification of Objects Based on Generalized Amplitude-Phase Images Statistical Models

Viktor Vlasenko[1(✉)], Sławomir Stemplewski[2], and Piotr Koczur[1]

[1] Faculty of Nature and Technical Sciences, Opole University, Oleska 48, 45-052 Opole, Poland
vlasenko@uni.opole.pl
[2] Institute of Mathematics and Computer Science, Opole University, Oleska 48, 45-052 Opole, Poland

Abstract. The article presents the dynamical objects identification technology based on statistical models of amplitude-phase images (APIm) – multidimensional data arrays (semantic models) and statistical correlation analysis methods using the generalized discrete Hilbert transforms (DHT) – 2D Hilbert (Foucault) isotropic (HTI), anisotropic (HTA) and total transforms – AP-analysis (APA) to calculate the APIm. The identified objects are modeled with 3D airplanes templates rotated in space around the center of Cartesian coordinate system. The DHT domain system of coordinates displaying the plane projections (2D flat images) remains to be space-invariant. That causes the anisotropic properties of APIm and makes possible the tested objects effective matching to rotated templates and identification of shapes at DHT domains. As additional method for objects matching accuracy increasing the difference (residual) relative shifted phase (DRSP-) images templates are proposed. The hierarchical system for identification is based on correlation analysis and decision making on semantic models – sets of AP-histograms (adjacency arrays), DRSP-images, APIm with specified angles shifts.

Keywords: Generalized Hilbert transforms · Amplitude-phase images · Dynamic object identification

1 Introduction

The complex scenes analysis [1, 2] usually includes the methods to the moving objects images modeling and recognition. Objects move on scenes' background consists of other objects and textures and have the complex trajectories. Due this the shapes of dynamical objects displayed with imagery systems as flat 2-D time-spatial images are changed drastically. Application the theory of generalized analytical signals [3] to such methodic elaboration is recently very prospective and more often used approach to imaging systems analysis and modeling. This approach is based on discrete Hilbert transform and hyper-complex signals and generalized Fourier spectra theory [3, 4] for multidimensional images representation [4–6]. The fundaments of information technologies (IT) elaborated within this approach are methods of amplitude-phase analysis (APA) multidimensional data arrays – AP-images of DHT spectra, Foucault (DH(Fc)T) and others hybrid

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_6

transforms [6–8]. Hybridization of 2-D images APA methods based on generalized (hyper-complex – quaternion - QHT, octonions - OHT) Hilbert transform increases the discrimination abilities of shape-distorted objects' images identification [6, 7]. However introduction to IT additional more complicated transformations increases the apparatuses and computing time demands ("identification costs"). In case of very large size library of templates to be compare with tested objects the identification IT effectiveness is decreasing. This demands the effective minimization of identification costs. The main goal of article presented is the describing of methods and structures of IT based on statistical models of multidimensional APA-data images may be used for dynamical objects localization and identification. The article is further developing of previous authors' results in practical IT tasks modeling [8–11]. The DHT-transforms, AP-images, 2-D histograms of amplitude and phase are synthesized and specified at controlled movement (flat shift translation and rotation) conditions as matrix models of flying objects. Correlative analysis of these matrixes (statistic models space) leads to clustering the groups of AP-objects with close shape and structure properties accumulation identifying objects as instances - members of particular class. After detecting "potentially suspected objects" on 2-D histogram templates level and gathering its close to tested object the changing of identification level (to AP-image) is occurring. The dimension of characteristic features space (number of templates to compare) after level changing to catalogue of AP-images (structural models space) is reduced significantly. The hierarchical structure of IIT and growth of object - template matching accuracy due the using of AP- (DRSP-) images as templates permits to obtain the reducing costs of identification.

2 Structure and Functions of AP-Images Identification IT (IIT)

Conceptual chart of APA-methodic based IIT to dynamical objects identification is presented at Fig. 1. IIT structure consists of such modules: synthesis of 3-D objects – spatial models as "source" templates to be identified and design, using its as input initial points, local data base (LDB) as a set of semantics {Class – Object – Image}; synthesis of 2-D images – spatial-time objects in process of moving, and design of LDB based on semantics {Object – Image – DataArray_AP}; synthesis of statistical models as matrix-histograms (adjacency arrays) of AP-image sets {*hist2*(*A, P1*); *hist2* (*A, P2*); *hist2*(*P1, P2*)}; design of template library of models synthesized. Second part of IIT is the module to APA – analysis and identification of tested objects images. Analysis is provided on base of AP-image description correlative models (secondary semantics derivative on image, data array and statistical models levels).

Structure of taxonomy of semantics of templates objects is presented on Fig. 2. As root of semantic tree (layer O_R) there are models of objects – sets of graphics files of rendered image of 3-D objects at initial (physical) space (O_PS) generated and controlled by graphical animation programming environment rotated at 3-dimensions space. Next layers – levels of description are organized on basis of sets of files 2-D gray (O_GR), binary (black - white) (O_BW), amplitude-phase (O_AP) images. The leafs of taxonomy tree are sets of 2-D matrix-histograms {*hist2*(*A, P1*); *hist2*(*A, P2*); *hist2*(*P1, P2*)}. Semantics are presented as 3-D arrays of graphical files corresponding to levels of

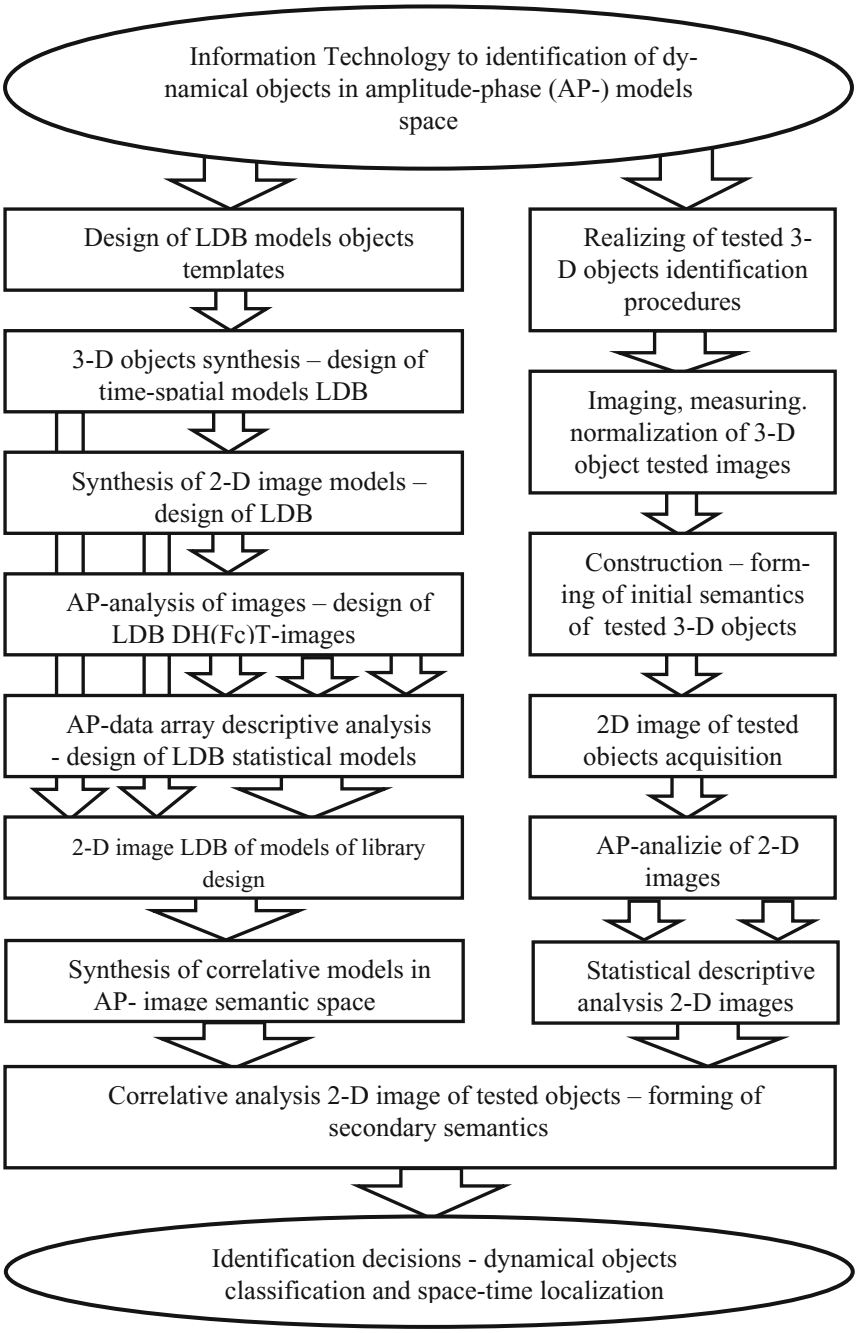


Fig. 1. Conceptual chart of IT to dynamical objects identification based on AP-images

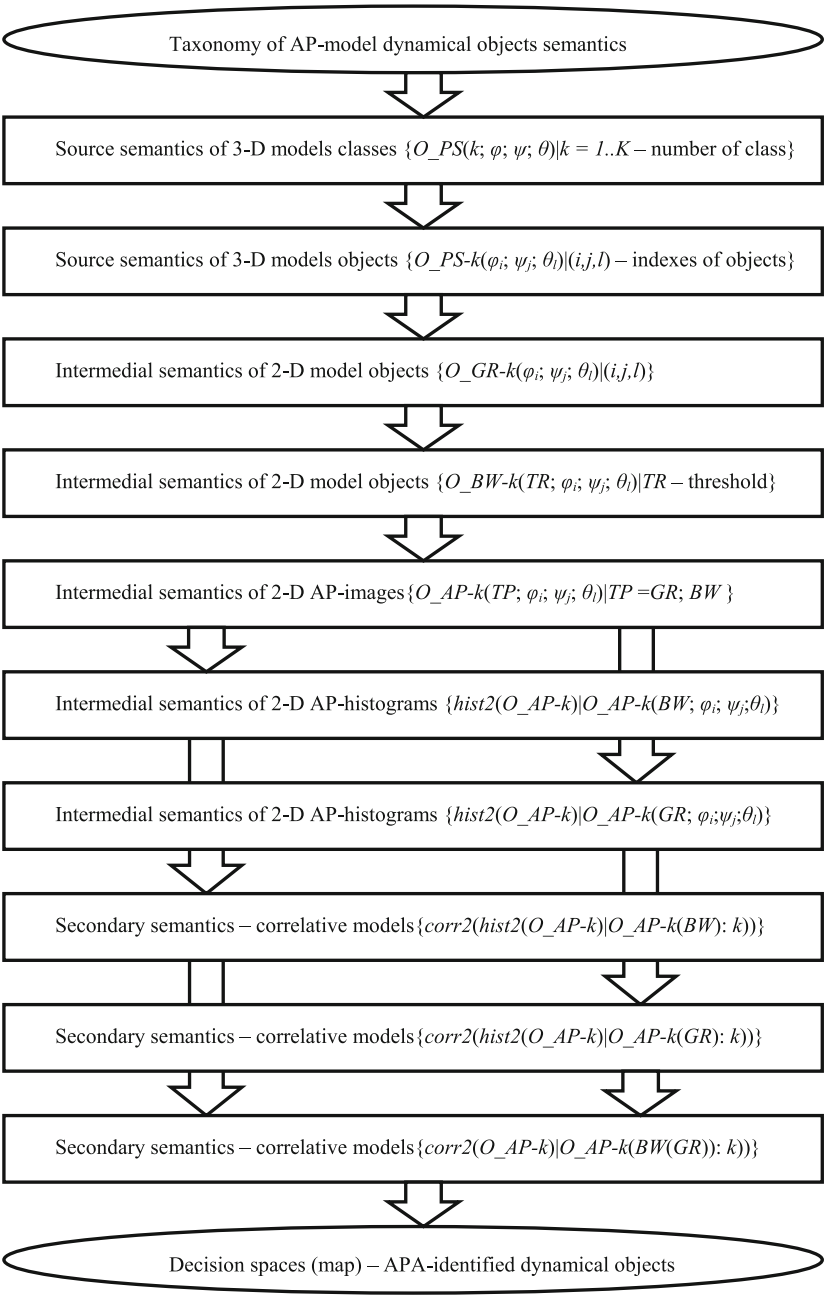


Fig. 2. Structure of semantic models taxonomy of objects identified at AP-analysis

objects' description. During the identification - decision making (classification) process (DMP) over the sets of tested objects is moving at back-propagation direction "histograms {*AP-hist*} – AP-images {*O_AP*} – 2-D images {*O_BW*} – 2-D images {*O_GR*}" via estimation and choice the maximum correlated objects at each level (step of DMP) as potentially members of the same class. As example of the identification

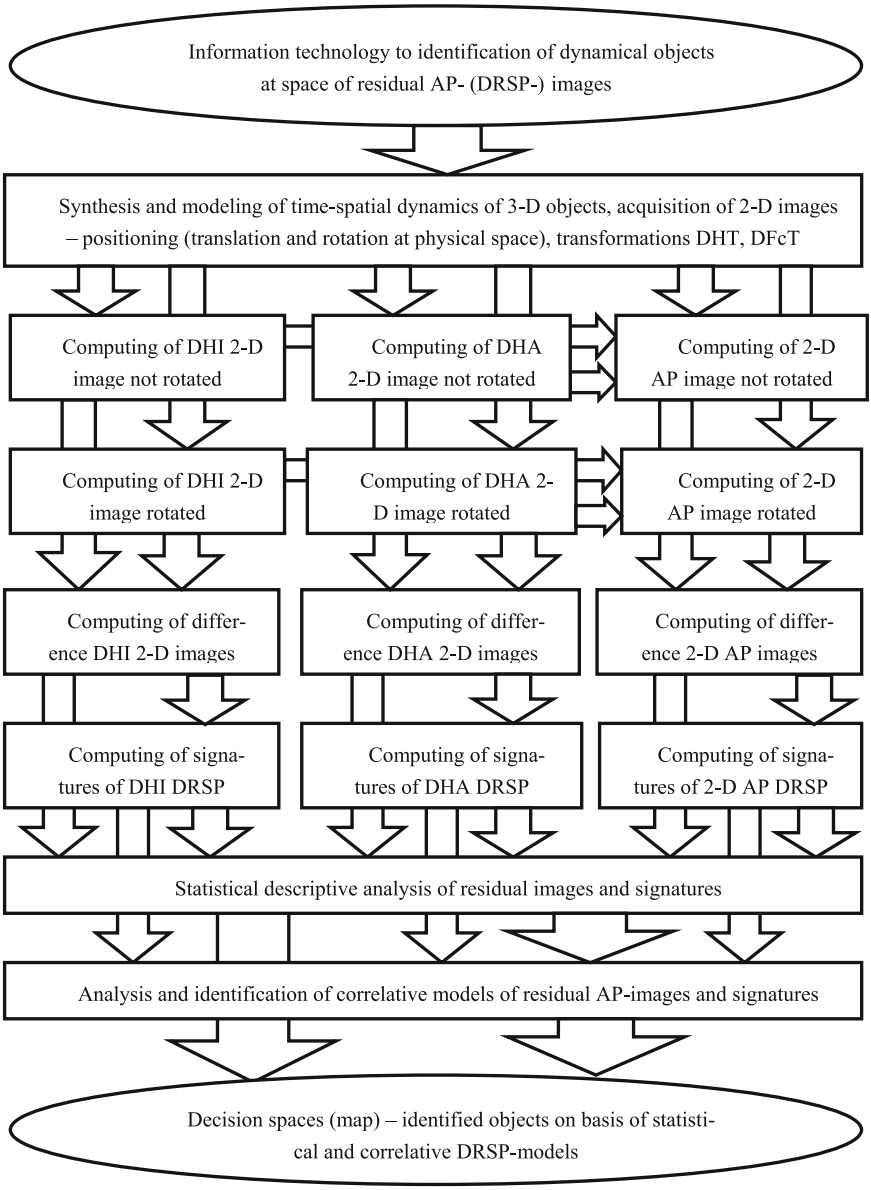


Fig. 3. Conceptual chart of information technology to dynamical objects identification based on method of AP-analysis and modeling of residual AP-images

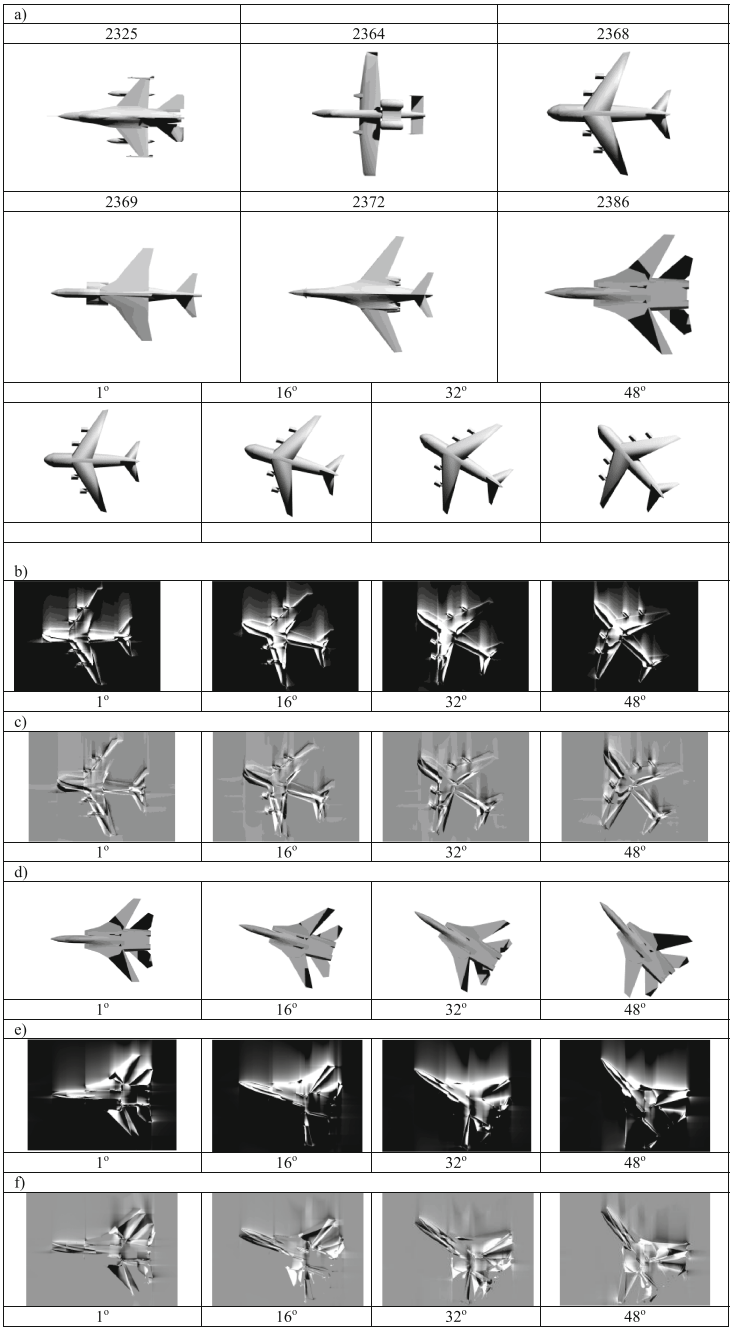


Fig. 4. Examples of tested objects images: initial – (a), (d); AP- – (b), (e); DRPS- – (c), (f)

methods described above Fig. 3 presents the conceptual chart of IIT based on AP-analysis i modeling at the space of residual phase (DRPS) images. Diagram contains modules of synthesis and rotations 3-D-images (on orthogonal planes $(x0y) - \varphi$, $(x0z) - \theta$, $(y0z) - \psi$), acquisition of 2-D images – projections, computing of AP-images sets, residual phase images $\Delta\Phi(x,y \mid \varphi, \psi, \theta) = \Phi(x, y \mid \varphi, \psi, \theta) - \Phi(x, y \mid \varphi + \Delta\varphi, \psi + \Delta\psi, \theta + \Delta\theta)$, statistical descriptive and correlative analysis.

3 Examples of Models of Dynamical Objects AP-Images

Modeling of AP-images as illustration of designed IIT functionality has been provided on basis of designed LDB of 3-D airplane models (format GR with specified objects parameters of illumination, ranges of angles rotation φ, ψ, θ and steps of relative angle shift at DRPS-images $\Delta\varphi, \Delta\psi, \Delta\theta$). Figure 4a, d presents the examples of initial flat rotated airplanes (O_PS, $\varphi = 0^o; 16^o; 32^o; 48^o$).

Figure 4b, e presents phase images of these objects, Fig. 4c, f – residual phase DRPS-images corresponding to relative shifts of AP-images $\Delta\varphi = 5^o$. Figure 5 presents

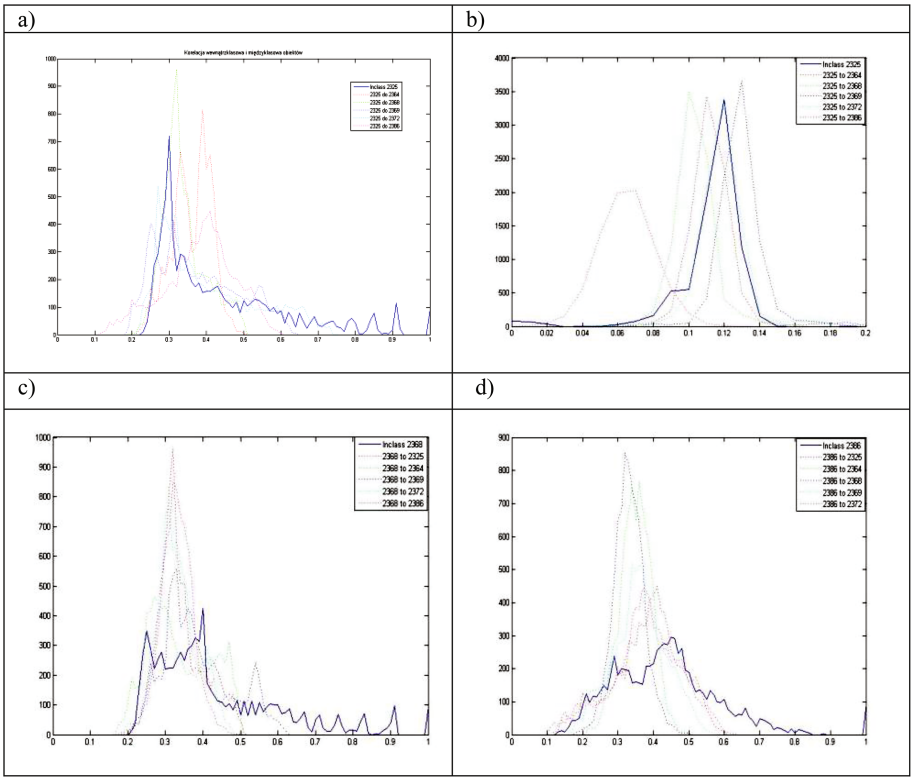


Fig. 5. Examples of CoC histograms: AP-images: 2325 – (a), 2368 – (c), 2386 – (d); DRPS-images: 2325, $\Delta\varphi = 8^o$ – (b).

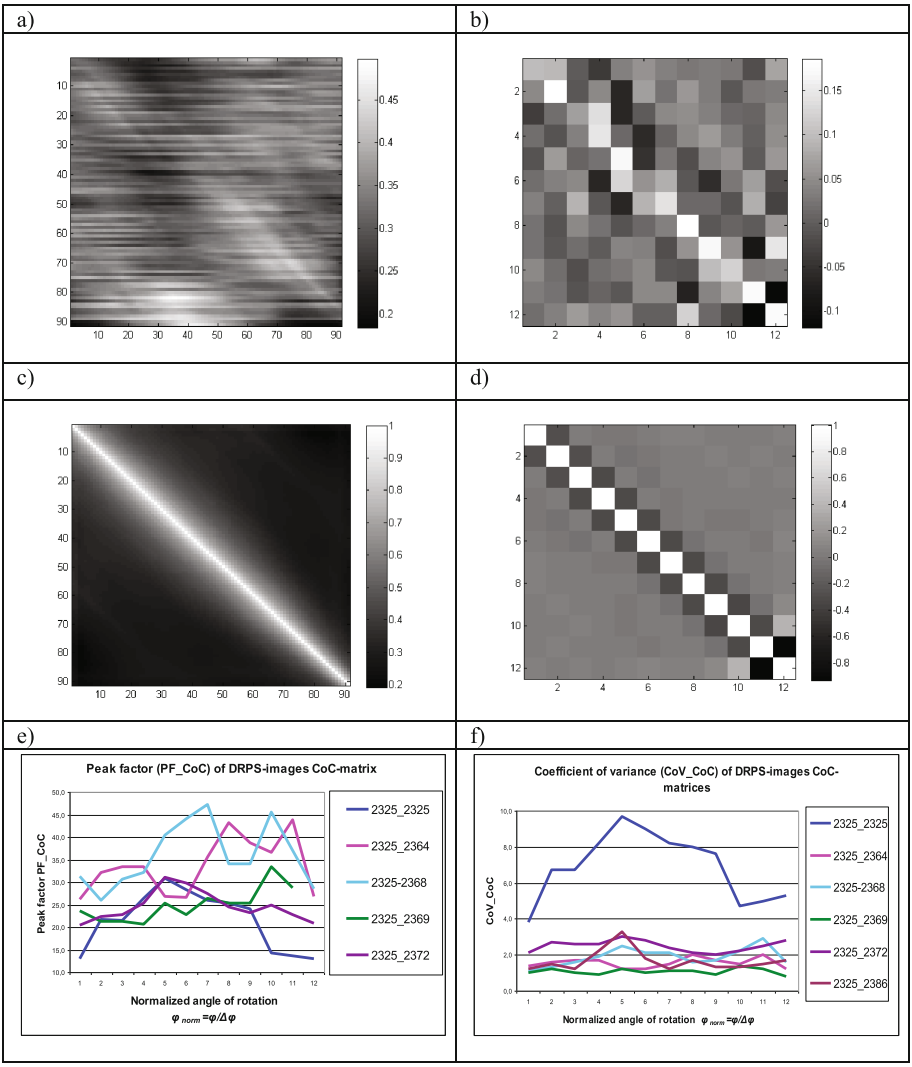


Fig. 6. Cross- and auto-correlation matrices of rotated DRPS-images: cross-correlation ($\varphi = (0..90)^o$, $\Delta\varphi = 1^o$) – (a); cross-correlation ($\varphi = (0..90)^o$, $\Delta\varphi = 8^o$) – (b); auto-correlation ($\varphi = (0..90)^o$, $\Delta\varphi = 1^o$) – (c); auto-correlation ($\varphi = (0..90)^o$, $\Delta\varphi = 8^o$) – (d); peak factor of correlation matrices (e), coefficient of variance of correlation (f) – as functions of normalized angle shift $\varphi_{norm} = \varphi/8^o$ value

the histograms of coefficients of auto- and cross-correlation phase images (AP-) – (a) and residual phase angle relative shifted images (DRPS-) – (b).

Figure 6 presents matrices of DRPS-images auto- and cross-correlation: cross-correlations of different objects images – (a), (b), auto-correlations of the same object images – (c), (d). Analysis of experimental data – (e), (f) shows the advantages

of DRPS-phase image correlation – growth of correlation pick-factor leads to increasing of discrimination abilities of identification methods and effectiveness of IT.

4 Summary

Semantic analysis and modeling of dynamical objects with various shape and localization based on generalized Hilbert–Foucault transforms (DHO) and calculating of convergence and adjacency measures improve the discrimination abilities of identification IT and systems its realizing. Evaluations of correlation models of APIm pointed the significant arising of matrix peak-factor corresponding to class and localization. This fact proves the perspectives of both side reducing – time and cost of shape identification and decreasing of tested objects localization (matching) errors. These facts point optimal solution – hierarchical rotated-invariant fast identification and minimization of number of angle-correlation channels at parallel (or hierarchical) classificators.

References

1. Pratt, W.K.: Digital Image Processing: PIKS Inside, 4th edn. Wiley, New York (2010)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
3. Hahn, S.L., Snopek, K.M.: Complex and Hypercomplex Analytic Signals: Theory and Applications. Artech House, Boston (2017)
4. Hahn, S.L.: Hilbert Transforms in Signal Processing. Artech House, Norwood (1996)
5. Lorenco-Ginori, J.V.: An approach to 2D Hilbert transform for image processing applications. In: Kamel, M., Campilho, A. (eds.) ICIAR 2007, pp. 157–165 (2007). Montreal
6. Wietzke, L., Flejschmann, O., Sommer, G.: 2D image analysis by generalized Hilbert transforms in conformal space, In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II, pp. 638–649 (2008). Marseille
7. Wietzke, L., Flejschmann, O., Sedlazeck, A., Sommer, G.: Local structure analysis by isotropic Hilbert transforms. In: Goesele et al. (eds.) DAGM 2010, pp. 131–140 (2010). Darmstadt
8. Sudoł, A., Stemplewski, S., Vlasenko, V.: Methods of digital Hilbert optics in modelling of dynamic scene analysis process: amplitude-phase approach to the processing and identification objects' pictures. In: Information Systems Architecture and Technology, pp. 129–138. Politechnika Wrocławska, Wrocław (2014)
9. Vlasenko, V., Sudoł, A.: DHO-methodology for complex shape objects and textures at dynamic scenes identification: structure design, modeling and verification. Syst. Sci. **35**(3), 15–29 (2009)
10. Vlasenko, V., Sudoł, A.: Identyfikacja Scen Dynamicznych: Zastosowania Cyfrowej Optyki Hilberta w Modelowaniu Procesów Obróbki Obrazów i Sygnałów w Systemach Optoelektronicznych. Przegląd Telekomunikacyjny, pp. 1892–1902 (2009). Warszawa
11. Vlasenko, V., Vlasenko, N., Semenov, D., Sudoł, A.: Modeling and verifications of information technologies for MDHO-identification of objects and textures at dynamic scenes. In: Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, pp. 49–62 (2008). Wroclaw

Abnormal Textures Identification Based on Digital Hilbert Optics Methods: Fundamental Transforms and Models

Viktor Vlasenko[1(✉)], Sławomir Stemplewski[2], and Piotr Koczur[1]

[1] Faculty of Nature and Technical Sciences, Opole University, ul. Oleska 48, 45-052 Opole, Poland
vlasenko@uni.opole.pl
[2] Institute of Mathematics and Computer Science, Opole University, ul. Oleska 48, 45-052 Opole, Poland

Abstract. The article presents the abnormal textures identification technology based on structural and statistical models of amplitude-phase images (APIm) – multidimensional data arrays (semantic models) and statistical correlation analysis methods using the generalized discrete Hilbert transforms (DHT) – 2D Hilbert (Foucault) isotropic (HTI), anisotropic (HTA) and total transforms – AP-analysis (APA) to calculate the APIm. The identified fragments of textures are obtained as examples of experimental observation of real mammograms contains areas of pathological tissues. The DHT based information technology as conceptual chart description is discussed and illustrated with DHO domain images. As additional method for anomaly of tissue detecting the multiply cascade DHT is proposed and elaborated at base transforms domains. The enhancement of abnormal texture areas at mammograms processed could increase the abilities of identification methodic and diagnostic systems.

Keywords: Generalized Hilbert transforms · Amplitude-phase images · Abnormal textures identification

1 Introduction

The systems and technologies of digital image processing and recognition of images and textures in complex scenes [1–3] recently are more often based on digital Hilbert optics (DHO) methods used to analyze the amplitude and phase distributions of multidimensional data fields, identify complex shape objects, detect and localize the abnormalities of the fields backgrounds and to create the scenes structural descriptions on this basis. The development of generalized multi-dimensional (hyper-complex) analytical signals - QHT and OHT (Hilbert's quaternion and octonions processing) theory [4, 5] opens the prospect of developing and large-scale applying of DHO information technologies (IT) methods in automated imaging and treatment including identification of images in dynamic scene analysis, image diagnosis, microscopy, remote detection and others [7–12]. As one of the important areas of application should be an analysis of biological tissue in describing X-rays images, especially – mammographic (MMG). The basic activities in these issues are spatial-temporal analysis,

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_7

detection and identification of local anomalies in tissue textures, evaluation of dynamics of their development and degree of pathology risk. The use of tomographic synthesis for this purpose enhances the quality and effectiveness of diagnosis but is associated with an increasing in the level of radiation and costs of diagnosis [7–10]. Thus, current actually is the issue to deepening the analysis of traditional X-rays images by using additional digital filtering procedures – detection, modeling and identification of anomalous image fragments, which contains texture fields with eventual additions of different shapes objects (fragments of other textures). As texture models descriptions of elements of textures (TE), stochastic fields, fractals, amplitude-frequency spectrums (transformation), amplitude-phase models are used [1–3].

The purpose of the article is presentation of the possibility of using DHO methods in issues of diagnostic signs of mammography images visualization and the description of the structure and fundamental function of information technology which implements these issues. Amplitude-phase analysis (APA) of images are based on the evaluation of structural decomposition (space-time) of generalized amplitude fields and optical density phases of adaptive normalized images processed with Hilbert (Foucault) multidimensional transformations and their extensions [13–15]. The identification of fragments of anomalous textures is carried out in a multilayered decision space structure at 2-D levels of Hilbert's (Foucault's) transformed images, AP-images, 2-D histograms of initial images and transformations.

2 Structure and Functions of AP-Images Identification Technology

IT conceptual chart is presented on Fig. 1. Methodic consists of module 1 – set of fragmentation and segmentation procedures under diagnostic strategy control. The initial frames – MMG are fragmented with various sizes and orders of scanning. Also adaptive normalization is used to preserve the high level contrast and film (illumination) sensitivity invariance of optical density over whole frame area. In such kind normalized images (integral – whole non-fragmented frame or map of fragments) are transformed with DHT (DFcT) fast algorithms and presented as files of DHT transforms (isotropic DHI, anisotropic HTA) and amplitude-phase (AP-) images $A(m, n)$ and $\Phi(m, n)$ (2-D AP-images) (modulo 2). Transforms are analyzed visually by experts (physician – radiologist etc.). In parallel the local data base (LDB) is designed. LDB contains the images of normal tissues and abnormal pathological tissues. The source (initial) images are segmented (modulo 3) under correlation analysis and decision making processes control. As result of this analysis the map of decision is designed which overlaps the initial image, boundaries of abnormal segments and images fragments with pathologies.

Texture image semantic models taxonomy is presented on Fig. 2 as conceptual chart. As initial (source) the structures contain whole and fragments MMG image are used. The sizes of image could be adapter accordingly to semantics of images. Also the formats of image files could be changed in accordance to data export-import imaging hardware applications standards demands. Medium semantics are formed as result of data processing – computing of HTI, HTA, DFcT, QHT, OHT, AP-analysis etc.

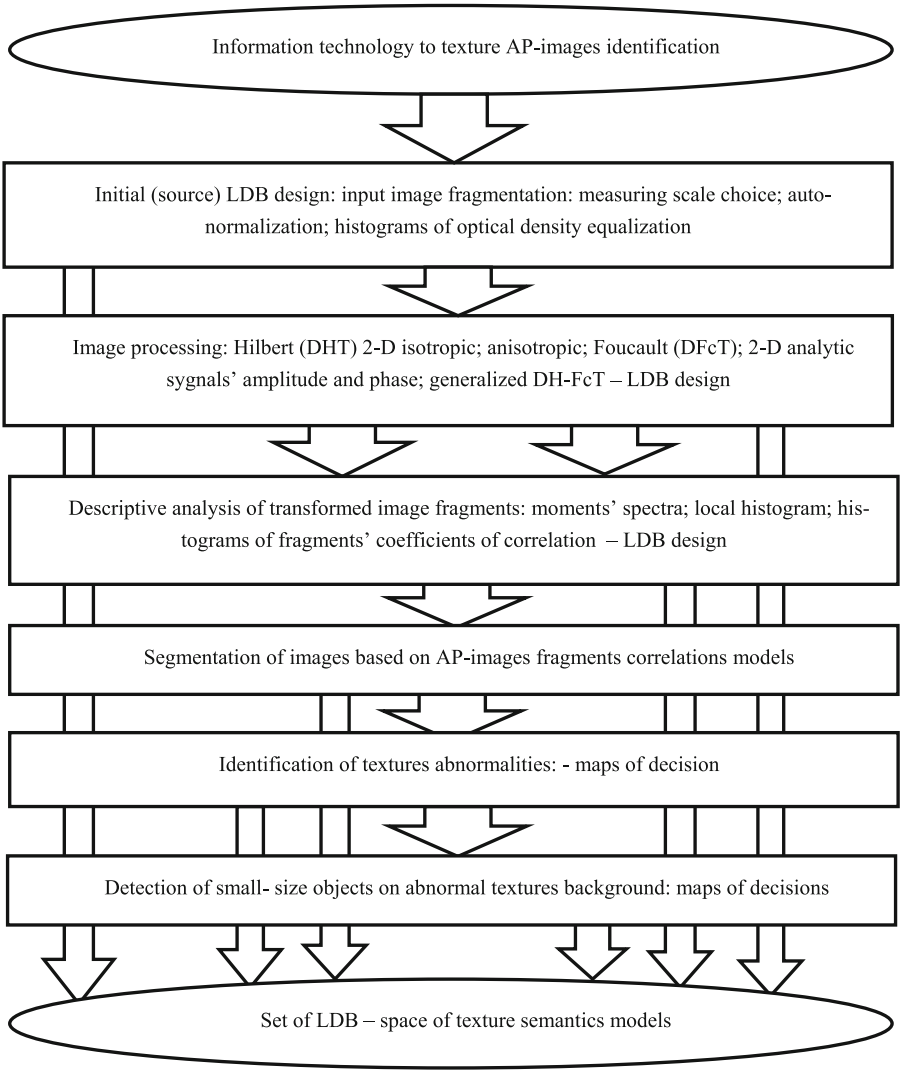


Fig. 1. Conceptual chart of methodic AP-analysis, modeling and identification of textures

Semantics are combined at LDB containing the frames – structures of graphical and textual descriptive files. The secondary (finish) semantics combine the results of analysis and correlative modeling with estimations of correlation coefficients of transform domain image fragments. Also these semantics contain maps of decisions and locations (labeling of fragments, objects, artifacts, abnormalities etc.), sets of statistical parameters etc.

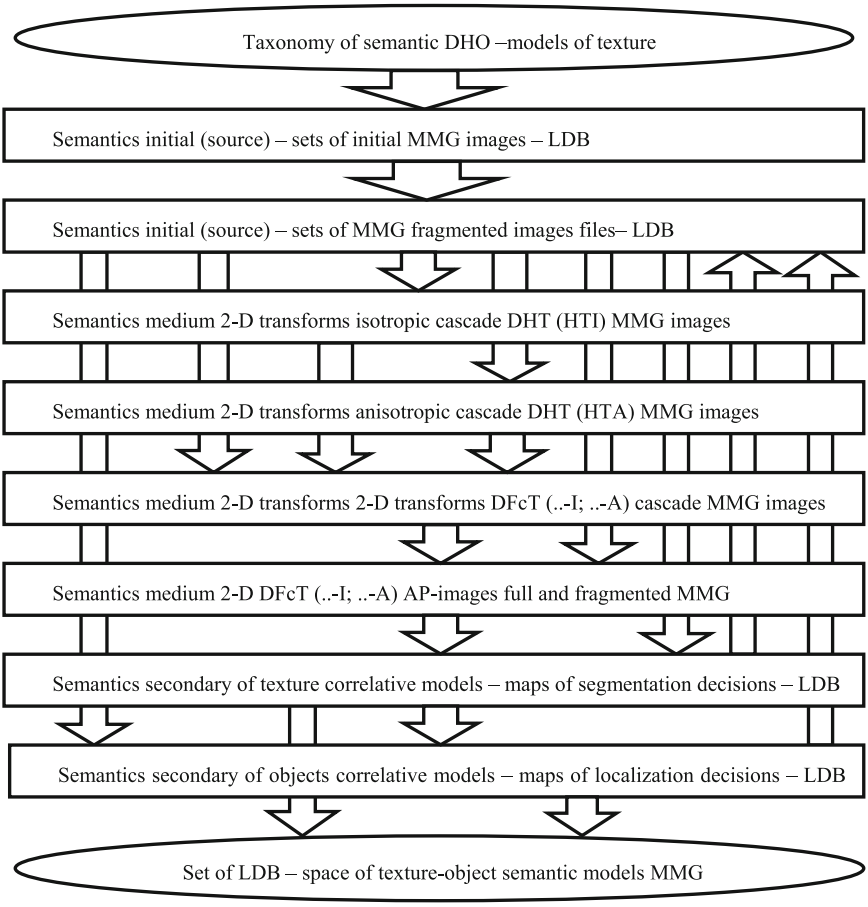


Fig. 2. Texture image semantic models taxonomy – conceptual chart

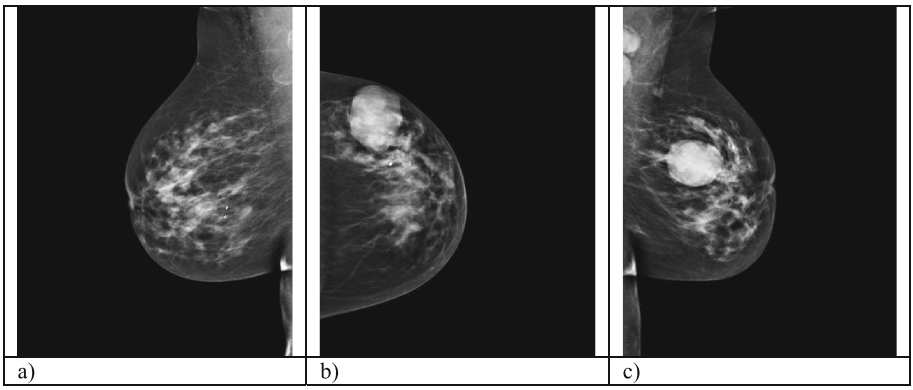


Fig. 3. Examples of MMG images with pathologies used for experiments with DHO IIT: (a) – 2977, (b) – 2943, (c) – 3002

3 Examples of DHO Identification Technology Applications at MMG Images Processing

Examples of source images (initial semantics) are presented at Fig. 3(a)–(c) as MMG images with pathologies. Figure 4 contains the results of processing based on DHO technology – transform HTI – (a), HTA – (b), FcTA – (c), FcTI – (d). Presented

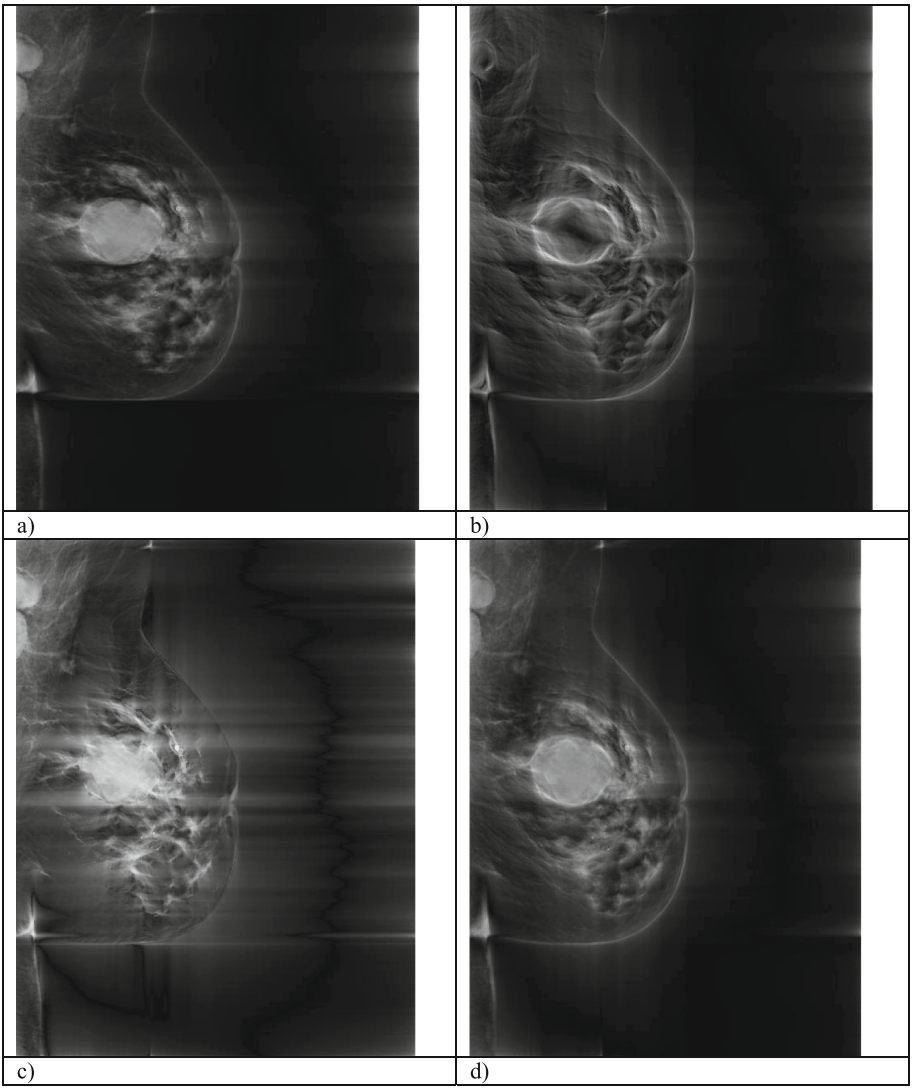


Fig. 4. Examples DHT (DFcT-) transforms MMG-image (3002: (a) – *hta_rgb*; (b) – *hti_rgb*; (c) – *fukoa_rgb*; (d) – *fukoi_rgb*)

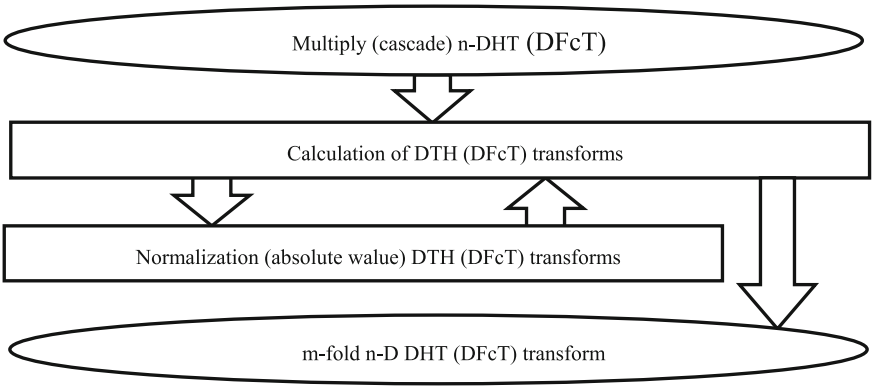


Fig. 5. Conceptual chart of double cascade 2-D DHT (DFcT)

transforms' images are used as data arrays to amplitude-phase image synthesize With technics of QHT. Application of iteration procedures – cascade 2-D Hilbert (Foucault) transforms leads to growth of local contrasts and gains characteristic features of abnormalities. This method increase the discrimination abilities of identification technology. Conceptual chart of cascade transform technology is presented on Fig. 5. Examples of textures images processed with cascade DHT method is presented on Fig. 6 (double 2-D transforms *hti_hti, hta_hta, fukoa_fukoa, fukoi_fukoi*).

4 Summary

The use of the amplitude-phase analysis method of mammography images based on 2-D isotropic and anisotropic Hilbert's (Foucault's) processing allows to increase the capabilities of discriminatory diagnostic procedures. This is related to the elevation of local contrasts of images as a result of processing (DHO filtration methods), reinforcement of characteristics features of amplitude and phase images - edges, vertices in shapes and line crossings, extreme points of sharp optical density changes, etc., distortions of phase structures - spatial-time changes, structures of texture field elements. These factors appear as markers of anomaly and supports the work of diagnostics. As a part of the experimental research, it have been identified the possibility of using the Hilbert's (Foucault's) double cascade processing method to treatment of image. That helps to enhance the AP-images and improve localization of pathologies.

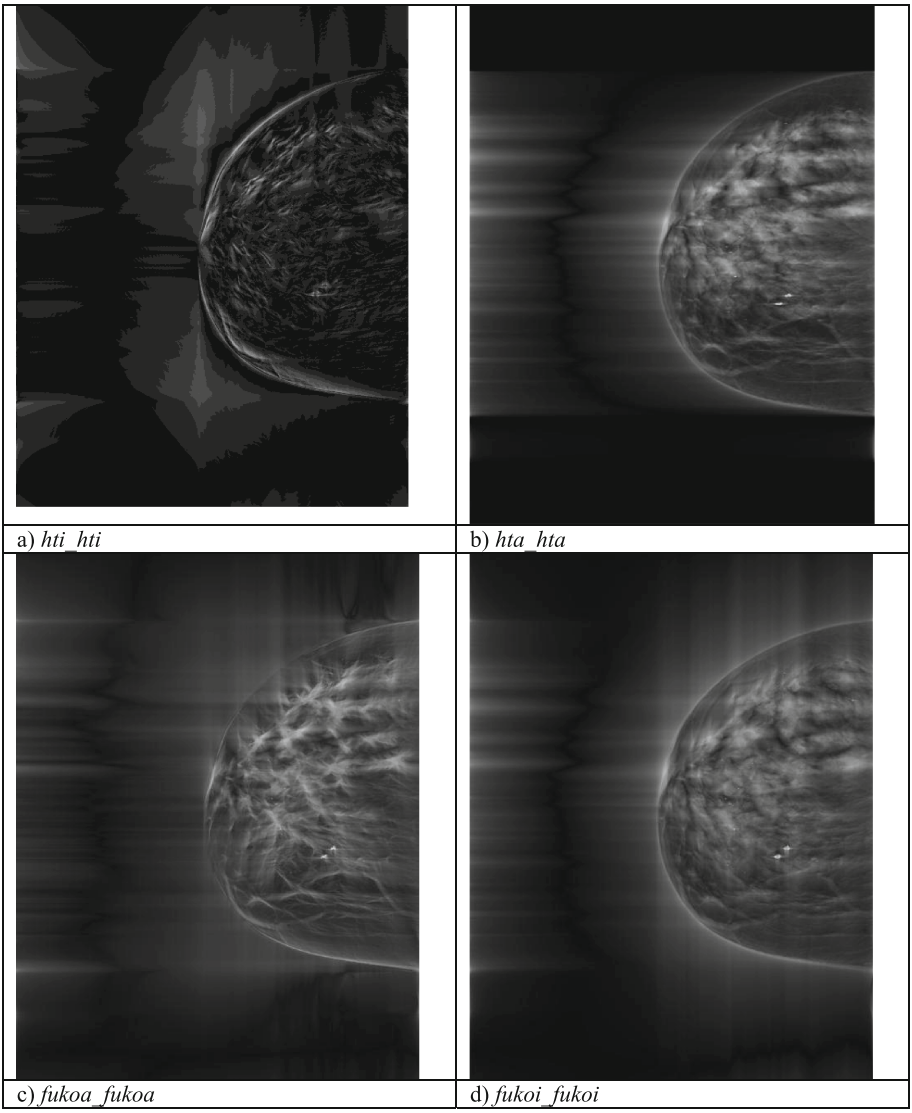


Fig. 6. Examples of 2-fold DHT (DFcT) transforms: (a) *hti_hti;* (b) *hta_hta* (c) *fukoa_fukoa;* (d) *fukoi_fukoi*

References

1. Pratt, W.K.: Digital Image Processing. PIKS Inside, 4th edn. Wiley, New York (2010)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)

3. Tuceryan, M., Jain, A.K.: Texture analysis. In: Chen, C.H., Pau, L.F., Wang, P.S.P. (eds.) The Handbook of Pattern Recognition and Computer Vision, 2nd edn, pp. 207–248. World Scientific Publishing Co., Singapore (1998)
4. Hahn, S.L., Snopek, K.M.: Complex and Hypercomplex Analytic Signals Theory and Applications. Artech House, Boston (2017)
5. Hahn, S.L.: Hilbert Transforms in Signal Processing. Artech House, Norwood (1996)
6. Wietzke, L., Flejschmann, O., Sedlazeck, A., Sommer, G.: Local structure analysis by isotropic Hilbert transforms. In: Goesele et al. (eds.) DAGM, pp. 131–140, Darmstadt (2010)
7. Guanlei, X., Xiaotong, W., Xiaogang, X.: Improved bi-dimensional EMD and Hilbert spectrum for the analysis of textures. Pattern Recogn. **42**, 718–734 (2009)
8. Ye, Y., Yu, H., Wei, Y., Wang, G.: A general local reconstruction approach based on truncated Hilbert transform. Int. J. Biomed. Imaging **2007**, 1–8 (2008). doi:10.1155/2007/63634
9. Li, L., Kang, K., Chen, Z., Zhang, L., Xing, X.: A general region-of-interest image reconstruction approach with truncated Hilbert transform. J. X-Ray Sci. Technol. **17**, 135–152 (2009)
10. Ji, C., Cui, Y., Cao, W., Bao, S.: Fast finite Hilbert transform via DE quadrature scheme. J. X-Ray Sci. Technol. **18**, 27–38 (2010)
11. Kumar, U.P., Somasundaram, U., Kothiyal, M.P., Mochan, N.K.: Single frame digital fringe projection profilometry for 3-D surface shape measurement. Optik **124**, 166–169 (2013)
12. Arnison, M.R., Cogwell, C.J., Smith, N.J., Pekete, P.W., Larkin, K.G.: Using the Hilbert transform for 3D visualization of differential interference contrast microscope images. J. Microsc. **199**, 79–84 (2000)
13. Sudoł, A., Stemplewski, S., Vlasenko, V.: Methods of digital Hilbert optics in modelling of dynamic scene analysis process: amplitude-phase approach to the processing and identification objects' pictures. In: Information Systems Architecture and Technology, pp. 129–138. Wroclaw University of Technology, Wroclaw (2014)
14. Vlasenko, V., Sudoł, A.: DHO-methodology for complex shape objects and textures at dynamic scenes identification: structure design, modeling and verification. Syst. Sci. **35**(3), 15–29 (2009)
15. Vlasenko, V., Vlasenko, N., Semenov, D., Sudoł, A.: Information technologies for dynamic objects and textures identification based on methods of digital Hilbert optics. In: Information Systems Architecture and Technology, pp. 153–160. Decision Making Models, Wroclaw University of Technology, Wroclaw (2007)

An Intelligent Multi-Agent System Framework for Fault Diagnosis of Squirrel-Cage Induction Motor Broken Bars

Maria Drakaki[1(✉)], Yannis L. Karnavas[2], Ioannis D. Chasiotis[2], and Panagiotis Tzionas[1]

[1] Department of Automation Engineering, Alexander Technological Educational Institute of Thessaloniki, P.O. Box 141, 574 00 Thessaloniki, Hellas, Greece
drakaki@autom.teithe.gr

[2] Electrical Machines Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Hellas, Greece

Abstract. Decision making in fault diagnosis is a critical factor in industrial production maintenance. Artificial intelligence (AI) techniques are widely used to accurately identify faults in induction motors. Multi-agent systems (MAS) as a distributed AI method are efficient in representing intelligent manufacturing systems to achieve decision robustness. In this paper, an intelligent MAS is developed for the decision making in fault diagnosis of three-phase squirrel cage induction motor rotor bars. Agents in the proposed MAS represent induction motor in different health conditions, i.e. healthy motor and motor with 1, 2 and 3 broken bars and also a supervisor agent. Each agent is embedded with an artificial neural network and trained with measurement data taken from a motor in the corresponding health condition. Measurement data are obtained with the classical motor current signature analysis (MSCA) method. Each agent makes local decision making and communicates its output to the supervisor agent that makes the final fault diagnosis based on a threshold value.

Keywords: Multi-agent system · Induction motor · Fault diagnosis · Broken bars

1 Introduction

Induction motors are of paramount importance in the industry sector as they consume over 50% of the whole electric energy demand in most countries. Therefore maintenance has become a critical industrial process and capacity. Machine failure results in the production system being shut down, disruption or interruption in operations and both services and profit losses. Induction motor faults have been mainly categorized as electrical faults, mechanical faults and outer drive system faults [1, 2], whereas rotor counterparts account for approximately 10% of the total induction motor failures [3].

Early fault detection is critical for optimal operation, therefore both cost and performance issues have established the condition based preventive maintenance. Accordingly, fault decision making follows fault detection, whereas maintenance scheduling is done based on the decision output. Early fault detection methods in

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_8

squirrel cage induction motor (SCIM) were first based on vibration or thermal monitoring e.g. [4] but their response were strongly conditioned by dynamic characteristics and the mounting structure. Also, there were difficulties associated with the use and positioning of the sensors. Later, the researchers focused on electrical quantities monitoring, standing out the technique known as motor current signature analysis (MCSA) (e.g. [5]) which is currently considered as a standard in preventive maintenance and is the reference method for the diagnosis of medium-to-large machines in industrial applications basically due to the advantages of just a clamp meter used as sensor [6]. At the same time, many other techniques were developed and proposed in the literature as alternative ways to diagnose the motor condition, such as wavelet transformation and signal analysis, reconstruction of phase-space, Hilbert transform, start-up transient analysis, finite element method analysis and other. Model-based fault diagnosis includes methods lying mainly in the artificial intelligence (AI) area. Neural networks [7], fuzzy logic [8], genetic algorithms as well as their combinations are only some recent examples. The MCSA is based on the spectral analysis of measured motor supply currents. Power spectra profiles are processed to derive motor faults, as harmonics are induced resulting from faulty motor stator currents. Although MCSA as a stand-alone technique is considered nowadays a reliable fault detection technique, background noise in recorded data may introduce difficulties in the spectra interpretation. Hence, a high level of expertise is required then to perform a reliable analysis. Additionally, the accuracy of a spectrum is limited due to "energy" leakage effect along with the fixed resolution used [9], and a nonlinear behavior of the machine with or without faults can make this task more difficult [10]. Thus, the objective of an effective condition monitoring system is to determine a procedure for extracting different features from the current signal in order to discriminate among various machine conditions, classifying faulty modes from normal modes. However, each problem solving procedure has its own limitations, e.g. AI based techniques such as neural network (NN) or neural-fuzzy techniques require either large data volumes or increased computation times to obtain an optimal solution. Thus, there is no global "one size fits all" solution for different fault types or different measurement methods.

Driven by global competition and manufacturing sustainability, fault tolerance is a major requirement in distributed intelligent manufacturing systems (IMS), which exhibit among others autonomous decision making, emergent behavior, adaptation, cost effectiveness and high quality. In the distributed artificial intelligence (AI) field, an agent is an autonomous software system, a problem solver, with local decision making, capable of interacting with other agents using communication and cooperation in order to solve a complex problem when it is beyond its own capability [11]. Multi-agent systems (MAS) were initially deployed to address the features of IMS with a focus on providing decision robustness with respect to disruptions and complex dynamic environments (e.g. [12, 13]). A MAS is defined as a set of agents that interact to achieve their individual goals, when they have not enough knowledge and/or skills to achieve individually their objectives [13].

In this paper, a MAS is presented to make the fault diagnosis on SCIM rotor bar health. The MAS uses neural networks to assess the rotor bar health status. Neural networks are widely used in intelligent systems for motor fault diagnosis. They do not involve derivation of complex mathematical models. They are good classifiers,

especially when the training data set concerns experimental measurement data. Experimental measurement data have been obtained from a healthy motor and from the same motor with 1, 2 and 3 broken rotor bars. The MAS consists of a supervisor agent and eight motor agents. The motor agents represent a healthy motor and motors with 1, 2 and 3 broken rotor bars trained with either the FANN or the k-NN classifier method. The supervisor agent receives test data and sends it to the motor agents trained with the FANN method. Each agent makes a fault diagnosis and informs the supervisor agent with the result. The supervisor makes the final fault diagnosis based on a predefined threshold, whereas in case of conflict it will request repetition of fault diagnosis from motor agents trained with the k-NN classifier method. The MAS is implemented with JADE (Java Agent Development Framework) [14].

2 Rotor Bar Faults

It is known that, high rating machines are manufactured with copper rotor bars whereas lower rating ones are manufactured by die casting process. Thus, failures may result in rotors due to several reasons. The most important are (a) dangerous stresses on the bars because of large centrifugal forces caused by heavy end ring, (b) resulting thermal stresses imposed during start-up while the rotor bar cannot move longitudinally in the occupied slot and (c) possible non uniform metallurgical stresses into the cage assembly during the brazing process in manufacturing. In any case, when one or more rotor bars are damaged an unbalance rotor situation occurs. Even in case of a small crack, a bar break may take place due the overheating of the surrounding area of the crack. Consequently, the surrounding bars will carry higher currents and therefore they are subjected to even larger thermal and mechanical stresses and they may also start to break. In this rotor asymmetry condition, a resultant backward rotating field at slip frequency appears with respect to the forward rotating rotor. This backward-rotating field induces a voltage in the stator at the corresponding frequency and its related current modifies the stator-current spectrum. Thus, the side-band frequencies which can be considered as expected for fault indications are,

$$f_{sh} = f_1(1 \pm 2s) \tag{1}$$

where f_1 is the stator currents fundamental frequency and s is the slip (difference from synchronous speed divided by the latter) at which the motor operates under a certain load.

However, another common relationship is usually used [5] for side-band frequencies over the double fundamental frequency ($2f_1$) which is,

$$f_{sh} = f_1[k(\frac{1-s}{p}) \pm s] \tag{2}$$

where p is the number of poles and k = 1, 2, ..m. It has been shown in [6] that for a k/p ratio values of 3, 5, 7 the current harmonic amplitudes are quite important. In any

case, the method employed should be able to predict with the highest possible accuracy the relevant harmonics characteristics corresponding to a specific fault.

3 MAS

Modularity and abstraction, present in MAS, are necessary tools to solve complex, real-life problems. Interaction is achieved with communication. Coordination, co-operation and negotiation are important agent communication characteristics. Coordination between agents in order to achieve system goals is necessary when agents have different capabilities or when different agents work on the same goal in order to complete their tasks more efficiently. Negotiation among cooperative agents is an effective coordination technique for complex problem solving [14].

A MAS is hosted on a dedicated software platform and organized according to predefined principles such as agent architecture, communication language, message protocols. The most popular agent architectures are the reactive and belief desire intention (BDI or deliberative) architectures. BDI agents exhibit human-like behavior. Reactive agents act with a stimulus-response mechanism. Reactive agents acquire intelligence as a result of their emergent behavior [14]. They do not have a model of the environment, the global system behavior is a result of their interactions, thus, simplicity, robustness and fault tolerance are their key properties. A main disadvantage of the reactive agent architecture is that the prediction of global system behavior is not an easy task, i.e. it is a trial-and-error exercise based on experimentation since individual agents acquire local information only.

Manufacturing is a main application area of reactive agents. Literature reviews on MAS in manufacturing have been given [11, 13]. The most important properties of an agent for manufacturing systems are autonomy, intelligence, adaptation, and co-operation [13], whereas intelligence can be increased through learning. Agents can represent physical objects such as machines, parts or human operators, as well as functions such as control, execution, maintenance and planning.

Among agent concepts deployed in industrial applications, knowledge sharing and distributed learning are applied to industrial distributed diagnostics, whereas complex problems and dynamic real-world environments have been efficiently modeled with multi-agent simulation. Distributed problem using MAS solving seems appropriate for complex problems, as such problems ideally need more computational power. Decisions in MAS depend on individual agent local decision making as well as on negotiation and coordination mechanisms between agents. Fault diagnosis methods in induction motors have a limited classification capability depending on the data sources or fault type.

When data sets are too large to be addressed by a centralized solution, e.g. when multiple faults are present in SCIMs, distributed diagnostics with MAS can be effective. In [15] an intelligent MAS was developed to identify different types of faults in the stator, rotor and bearings of a three-phase induction motor. Each agent was responsible for different type of fault and trained with different classifier method. Real-time industrial agent-based distributed diagnostic solutions for manufacturing have been presented (e.g. [16, 17]).

4 JADE

JADE is a widely used middleware for the development of FIPA compliant (Foundation for Intelligent Physical Agents) agent frameworks. It provides the infrastructure on top of which agent logic can be developed [14]. The main purpose of FIPA is to provide interoperability between different agent platforms, thus JADE is compatible with different agent platforms. JADE has been applied in manufacturing (e.g. [15, 18, 19]). JADE includes the Java classes required to develop agents and the containers, i.e. the Java processes that support the run-time and the services which support agent existence and execution. The JADE agent platform is the set of all active containers. The main container provides the GADT (global agent descriptor table), which is the registry of all agents present in the platform. It hosts two special agents, i.e. the Agent Managements Service (AMS) and Directory Facilitator (DF). The AMS agent provides access to the platform's white pages and manages agents' life cycle. Agents register with AMS to get a valid agent identifier (AID). The DF agent provides a yellow page service. Agents register their services to DF and search for other agents' services in DF. Agent communication is compliant with FIPA and based on asynchronous message passing. The message structure format is compliant with the widely used FIPA-ACL (asynchronous communication language) with predefined interaction protocols (IPs).

5 Description of the Proposed MAS Framework

The proposed MAS consists of a supervisor agent and eight motor agents, namely four motor agents representing a healthy motor and motors with 1, 2 and 3 broken rotor bars, trained with either a FANN or a k-NN classifier method. The MAS global goal is to make a decision on the health status of a SCIM rotor bar. After the initialization of the MAS on the JADE reactive agent platform, each motor agent is initially trained by its corresponding classifier method with measurement data. Then, when the supervisor agent receives test data it forwards it to the four motor agents trained by the FANN. Agents interact using the FIPA request IP. The supervisor agent initiates a conversation with each motor agent by sending the test data and requesting a decision on the type of fault. Each agent replies with an agree communicative act. When an agent fulfills the request it communicates an inform-result communicative act, to inform the supervisor agent that the request has been fulfilled and notify the results. The supervisor makes the final fault diagnosis based on a predefined threshold value. In case of conflict, e.g. when two or more agents assess positively, the supervisor agent sends the input data to the four motor agents trained with the k-NN method. If the conflict is still not resolved the supervisor agent issues the results along with a message indicating conflict. The case study architecture is shown in Fig. 1 while Table 1 shows the agents of the presented MAS. Moreover, Fig. 2 depicts analytically the agent interaction diagram.

Experimental measurement data have been obtained from a healthy motor and from the same motor with 1, 2 and 3 broken rotor bars. Figure 3(a) shows a generic view of the squirrel type rotor used here (in healthy condition, mounted on its rotating shaft) which has 17 copper bars. Vertical drilling of approximately 1 cm was performed consecutively in order to "simulate" an actual rotor bar fault, starting with the healthy

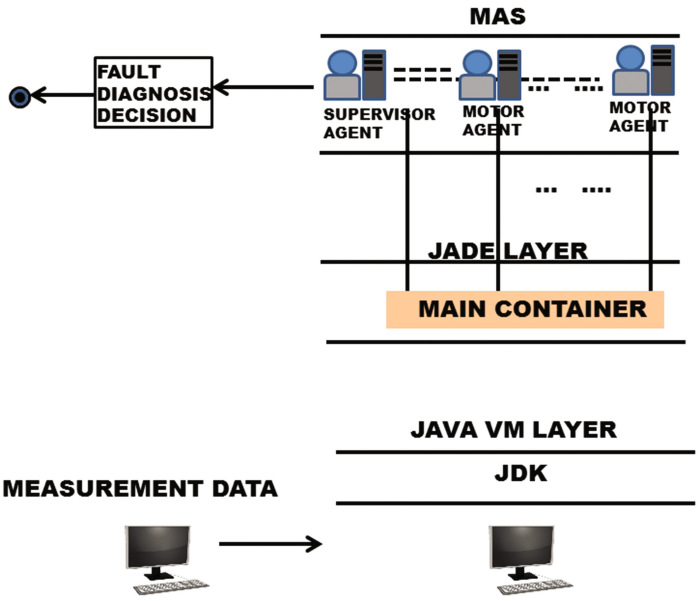


Fig. 1. The case study architecture.

Table 1. MAS agents and their respective goals

Agent		Goal
Supervisor agent (SA)		To make the final fault diagnosis decision of the global system
Healthy motor agent trained with FANN (HMA-FANN)	To decide whether the motor	is healthy using a FANN
Motor agent with 1 broken rotor bar trained with FANN (MA1BB-FANN)		has 1 broken rotor bar using a FANN
Motor agent with 2 broken rotor bars trained with FANN (MA2BB-FANN)		has 2 broken rotor bars using a FANN
Motor agent with 3 broken rotor bars trained with FANN (MA3BB-FANN)		has 3 broken rotor bars using a FANN
Healthy motor agent trained with k-NN (HMA-kNN)		is healthy using a k-NN
Motor agent with 1 broken rotor bar trained with k-NN (MA1BB-kNN)		has 1 broken rotor bar using a k-NN
Motor agent with 2 broken rotor bars trained with k-NN (MA2BB-kNN)		has 2 broken rotor bars using a k-NN
Motor agent with 3 broken rotor bars trained with k-NN (MA3BB-kNN)		has 3 broken rotor bars using a k-NN

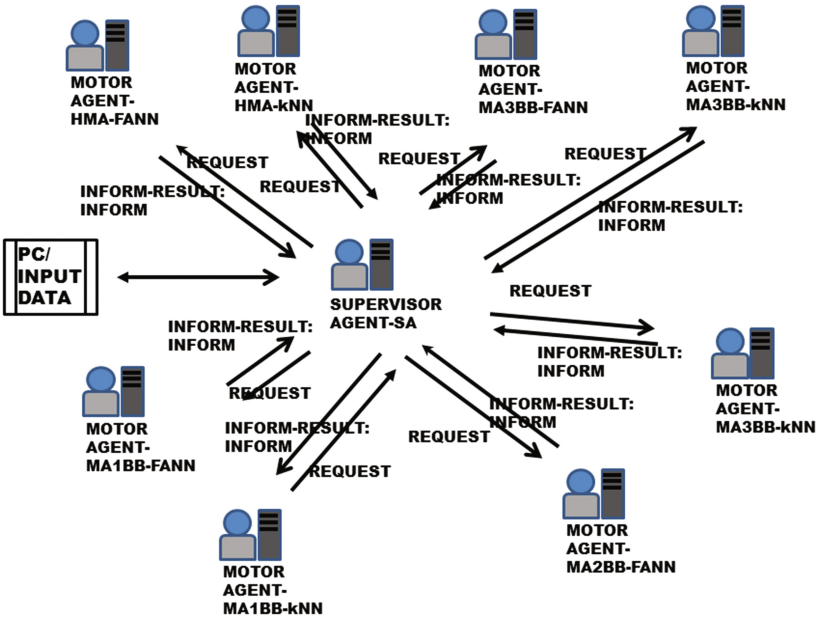


Fig. 2. Agent interaction diagram in the presented MAS.

rotor and increasing the holes/faults onto 3 adjacent bars. Figure 3(b) depicts a close view of the severely faulty rotor. Finally, in Fig. 3(c) the overall bench experimental setup is shown. Current waveforms were acquired and recorded using a digital oscilloscope through a current transducer for 4 different operating (loading) conditions and for each one of the 4 health states, thus 16 sets of measurements were obtained in total. The characteristics of the operating conditions examined are,

- Case 1: n_m = 1420 rpm (s = 0.0533), V_{ph} = 220 V
- Case 2: n_m = 1380 rpm (s = 0.0800), V_{ph} = 220 V
- Case 3: n_m = 1290 rpm (s = 0.1400), V_{ph} = 138 V
- Case 4: n_m = 1235 rpm (s = 0.1766), V_{ph} = 138 V

where n_m is the rotor speed and V_{ph} the SCIM phase voltage. These recordings were post-processed using FFT to obtain (among others) the corresponding power spectra.

As input nodes, 4 quantities have been selected, namely, (a) the slip of the motor (which inherits information about the loading condition), (b) the k/p ratio (which inherits information about the motor poles number), (c) the "expected" side-band frequency using Eq. (2) for k/p varying up to 9 with a 0.25 step (this covers a nearly 450 Hz harmonics spectrum) and (d) the corresponding current harmonic amplitude derived from measurements through FFT. In this way, for each health state and also for each of the operating condition cases, 16 input/output data sub-sets vectors were prepared, giving a final training/test data vector of 1056 rows by 4 columns.

Each agent is created by defining a class that extends the jade agent class. The agents are created by defining classes that extend the jade agent class. The motor agents

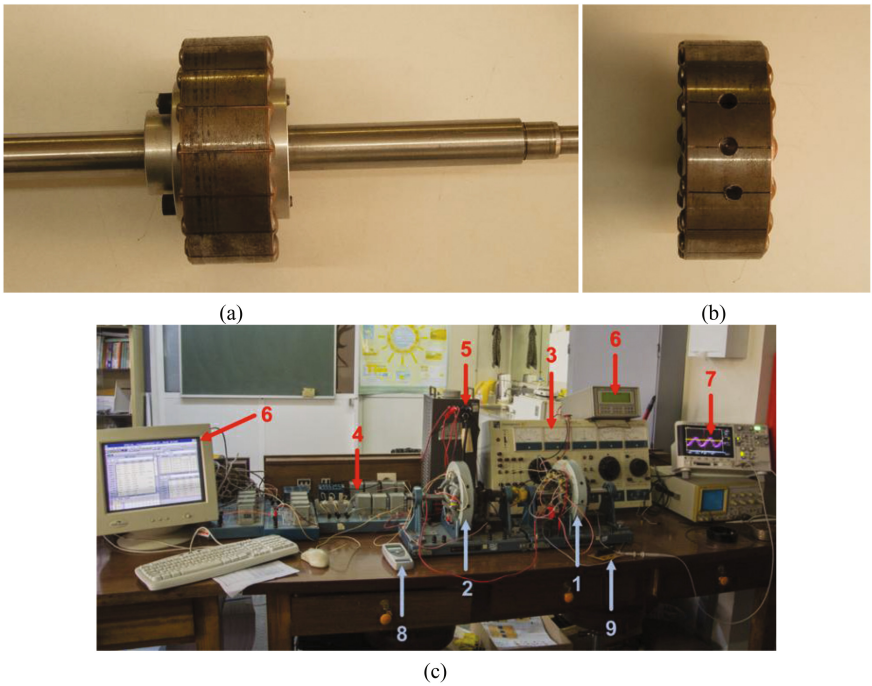


Fig. 3. (a) Actual SCIM rotor (on its shaft) used in experiments in healthy condition, (b) same rotor with 3 bars drilled consecutively to simulate broken bars, (c) Overall experimental setup: (1) three phase 4-pole induction motor under test, (2) dc-generator, (3) dc-generator excitation power supply, (4) resistive load, (5) three-phase power supply (variac), (6) power analyzer with PC connection, (7) digital oscilloscope, (8) tachometer, (9) current transducer.

incorporate a neural network, as a neural network class. The tasks that an agent has to fulfill are realized by adding behaviors to each agent class. The k-NN classifier training method uses four nearest neighbors and the Euclidean distance measure. The FANN training method is based on the gradient descent technique. The criterion for fault diagnosis decision making is the mean square error (MSE). The MSE is calculated from the sum of the squared errors for all input patterns used in the training set of the net. The activation function is the hyperbolic tangent. The weights are calculated and adjusted in each epoch until the MSE is minimized, where the training stops. The number of input processing elements is equal to four. The number of output layer processing elements is equal to one, as each motor agent corresponds to a specific fault type. One hidden layer is chosen and the number of hidden layer processing elements is equal to fifteen, based on trial and error. Training set parameters in the hidden and output layer are determined based on average minimum MSE. Step size is equal to 0.1 and momentum is equal to 0.7. The number of epochs is equal to 10000, the training data set is 70% and the test data set is 30% of the input data.

The results will be compared with fault diagnosis results extracted using the adaptive neuro-fuzzy inference system (ANFIS) method from the same motor [20], in

order to assess the effectiveness of the MAS as a fault diagnosis technique, as well as with respect to the conventional AI based ANFIS method.

6 Conclusions

Early fault detection in induction motors is critical for their operation, whereas conventional AI techniques for fault diagnosis have limitations with respect to an optimal solution. In this paper, an intelligent MAS is presented for the decision making in fault diagnosis of three-phase SCIM rotor bars. Agents in the proposed MAS represent induction motor in different health conditions, trained with either a FANN or a k-NN classifier method. Measurement data are obtained with the classical motor current signature analysis. Each agent makes its own decision making and sends its output to the supervisor agent that makes the final fault diagnosis based on a predefined threshold value. The MAS will be implemented with the JADE reactive agent platform. The effectiveness of the proposed MAS for SCIM fault diagnosis with respect to a conventional AI method will be tested by comparing fault diagnosis results extracted with this method with results extracted using the ANFIS method from the same motors. An additional advantage of using a MAS is that it can provide online fault diagnosis, since it is configured for real-time control.

References

1. Bonnett, A.H., Soukup, G.C.: Rotor failures in squirrel cage induction motors. IEEE Trans. Ind. Appl. **IAS-22**(6), 1165–1173 (1986)
2. Thomson, W.T., Fenger, M.: Current signature analysis to detect induction motor faults. IEEE Industry Applications Magazine June/July, pp. 26–34 (2001)
3. Marcello, C., Fossatti, J.P., Terra, J.I.: Fault diagnosis on induction motors based on FFT. In: Salih, S. (ed.) Fourier Transform Signal Processing. InTech, USA (2012). ISBN 978-953-51-0453-7
4. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electrical motors: a review. IEEE Trans. Energy Convers. **20**(4), 719–729 (2005)
5. Benbouzid, M.E.H., Kliman, G.B.: What stator current processing based technique to use for induction motor rotor faults diagnosis? IEEE Trans. Energy Convers. **18**(2), 238–244 (2003)
6. Bellini, A., Yazidi, A., Filippetti, F., Rossi, C., Capolino, G.A.: High frequency resolution techniques for rotor fault detection of induction machines. IEEE Trans. Ind. Electron. **55**(12), 4200–4209 (2008)
7. Matic, D., Kulic, F., Climente-Alarcon, V., Puche-Panadero, R.: Artificial neural networks broken rotor bars induction motor fault detection. In: Proceedings of 10th Symposium on Neural Network Applications in Electrical Engineering (NEUREL), 23–25 September, Belgrade, Serbia, pp. 49–53 (2010)
8. Laala, W., Zouzou, S.E., Guedidi, S.: Induction motor broken rotor bars detection using fuzzy logic: experimental research. Int. J. Syst. Assur. Eng. Manage. **5**(3), 329–336 (2014)
9. Payne, B., Ball, A., Gu, F.: Detection and diagnosis of induction motor faults using statistical measures. Int. J. COMADEM **5**(2), 5–19 (2002)
10. Haji, M., Toliyat, H.A.: Pattern recognition: a technique for induction machines rotor broken bar detection. IEEE Trans. Energy Convers. **16**(4), 312–317 (2001)

11. Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H.: Applications of agent-based systems in intelligent manufacturing: an updated review. Adv. Eng. Inform. **20**(4), 415–431 (2006)
12. Babiceanu, R., Chen, F.: Development and applications of holonic manufacturing systems: a survey. J. Intell. Manuf. **17**, 111–131 (2006)
13. Leitao, P.: Agent-based distributed manufacturing control: a state-of-the-art survey. Eng. Appl. Artif. Intell. **22**, 979–991 (2009)
14. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE, vol. 7. Wiley, New York (2007)
15. Palácios, R.H.C., da Silva, I.N., Goedtel, A., Godoy, W.F.: A novel multi-agent approach to identify faults in line connected three-phase induction motors. Appl. Soft Comput. **45**, 1–10 (2016)
16. Marik, V., McFarlane, D.: Industrial adoption of agent-based technologies. Intell. Syst. IEEE **20**(1), 27–35 (2005)
17. Albert, M., Laengle, T., Woern, H., Capobianco, M., Brighenti, A.: Multi-agent systems for industrial diagnostics. In: Proceedings of 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp. 483–488, Washington DC, USA (2003)
18. Lopez-Ortega, O., Lopez-Morales, V., Villar-Medina, I.: Intelligent and collaborative multi-agent system to generate and schedule production orders. J. Intell. Manuf. **19**, 677–687 (2008)
19. Komma, V.R., Jain, P.K., Mehta, N.: An approach for agent modeling in manufacturing on JADE TM reactive architecture. Int. J. Adv. Manuf. Technol. **52**(9), 1079–1090 (2011)
20. Karnavas, Y.L., Chasiotis, I.D., Vrangas, A.: Fault diagnosis of squirrel-cage induction motor broken bars based on a model identification method with subtractive clustering. In: 11th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED), 29 August–1 September, Tinos, Greece (2017)

Performance Comparison of Neural Network Training Algorithms for Load Forecasting in Smart Grids

Robert Lis[1(✉)], Artem Vanin[2], and Anastasiia Kotelnikova[2]

[1] Wroclaw University of Science and Technology, 50370 Wroclaw, Poland
Robert.Lis@pwr.edu.pl
[2] Moscow Power Engineering Institute, National Research University, Moscow 111250, Russia

Abstract. Voltage control of distribution systems need load forecast. For improvement of efficiency and sustainability of the automated control of Smart Power Grids (SPG) the control system of the process should contain a subsystem of forecasting time series - load forecasting, as a parameter directly associated with voltage. The highest requirements applicate for the accuracy of short-term (day-week-month) and operational (within the current day) forecasts, as they determine the management of the current mode of operation of the SPG. The paper describes forecasting methods and concludes that using of artificial neural networks for this problem is preferable. It shows that for the complex real networks particle swarm method is faster and more accurate than traditional back propagation method. Finally the used model of short-term load forecasting (STLF) is described and tuning of all presented models is done. The paper concentrates of training methods of neural networks and uses "back propagation" algorithm and "particle swarm" algorithm for this purpose.

Keywords: Voltage control · Load forecasting · Artificial neural network

1 Introduction

Nowadays, power quality is a very important parameter, both from a technical and an economic point of view. One of the monitored parameters is the level of voltage among consumers. With the development of network, users with different curves of load connected to the same main substation, but they have different requirements to the voltage regulation. Today in Russian distribution grids there are a variety of problems connected with power quality. The most common problem is an unacceptable level of voltage among consumers. Invalid voltage deviation leads to increased deterioration and failures of equipment, breakdowns in technological processes, incorrect operation of the control and automation systems. The common reasons of deviations are high network load, inconsistency of load curves, incorrect or insufficient regulation of voltage.

Now voltage control in distribution networks is carried out mainly using the tap changing transformers at power supply centers of 110–220/6–20 kV. Regulation is

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_9

performed in most cases in the stabilization voltage mode rarely counter regulation is applied.

Control actions for the both methods occur after the actual changing of mode settings that do not provide an optimal effect. In modern works is proposed to operate branches of tap changers of transformers using intelligent algorithms. In the Fig. 1 is shown example of working of such algorithm [1]. For the SPG's test system [5], in the Fig. 1 are presented consumers with different load graphs and with different requirements to the power quality. The points of connection with the consumer's service are numbered from 3 to 36. Bus number 2 marks the distribution substation. As it can be clearly seen, all the consumers' requirements for power quality is full-filled, but there is a problem with huge amount of tap-changings, which is lined by blue stepped line (up to 10 in one day). The resource of these devices is limited and it leads to the need to solve the optimization problem of distribution of control operations in time to ensure the best mode of all consumers. To solve this problem and provide an active-adaptive regulation is necessary to build STLF (from several minutes to a day) based on archive measurement and ambient parameters (temperature, light, day of the week).

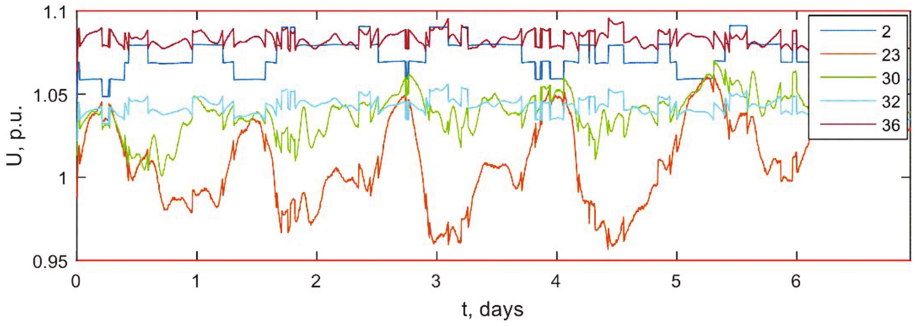


Fig. 1. The results of the work of active-adaptive control algorithm [5]

Thus, for the successful work of the active-adaptive algorithms, based on results of the known methods of load forecasting, it is necessary to choose the most effective method of forecasting and apply it to the available data [2].

The aim of this paper is to build the neural network load-forecasting model as the part of the active-adaptive algorithm of regulating of voltage for specific Moscow network based on real load data. In recent years, the new training approaches for neural networks achieved and it is needed to compare its performance and to choose the best solution for particular network.

2 Electric Load Forecasting Methods

In general, forecasting techniques are divided into intuitive and formal methods. The group of formalized methods is divided into statistical methods and artificial intelligence techniques, so-called heuristics. All the methods listed can be applied for the

load forecasting, but the selection of specific technique should be done based on particular problem and understanding for which purpose the chosen technique is used, which advantages or disadvantages it have and could it provide adequate results in this case or not [3, 8]. In this paper for forecasting the neural networks approach is implemented, because the data set is not big enough for using statistical methods, the fuzzy logic and SVM techniques gives big error in forecasting and requires an expert for setting the rules and building the model. Further some advantages and disadvantages for methods listed is given. In addition, recommendations and possible cases in which following techniques could be implemented is done.

2.1 Statistical Methods

Usually statistical methods may accurately predict the daily schedule load on ordinary days, but they lack the ability to analyze the load on holidays or other days, due to the absence of flexibility of their structure [9]. Statistical methods include multiple linear and non-linear regression, stochastic time series, the general exponential smoothing, the methods of state space, and others.

The apparent advantage of statistical methods is their "transparency", i.e. when solving problems of forecasting there is a known equation, based on which the problem is solved one way or another.

The main disadvantage of statistical methods are their computational complexity, long duration of computing and the need for large amounts of archived data. Statistical methods are difficult to apply to a model in which there are non-linear dependence, rapid load changes or missing data.

2.2 Heuristic Methods

Heuristic methods include artificial neural networks, fuzzy logic systems, Support Vector Machine (SVM) methods. An important advantage of these methods is their adaptability, i.e. the ability dynamically adapt to changing conditions. This property is one of the key for applications of the method as part of an intelligent control system for distribution networks.

However, fuzzy logic techniques itself are used very rarely, because the knowledge of operators and load monitoring system for long period of time have to be used in order to achieve good performance; fuzzy logic can be used for improving the performance of the artificial neural networks as well as self-organizing maps and genetic algorithms.

Heuristics are well suited for the prediction for the following reasons: first, they are capable numerically approximate any continuous function with a given accuracy. Secondly, to predict there is no need to build an accurate model of the system [4].

In addition, this methods are able to provide the required prediction accuracy in low-quality initial data, the presence in the archive gaps and abnormal deviations.

The disadvantages include the complexity of their initial design and the fact that most of them is "black boxes".

2.3 Artificial Intelligent (AI) Technique

A detailed comparison of classical and heuristic methods is given in [5]. Particular attention is paid to neural networks, since they are well suited for forecasting loads. They are able to generate a forecast for the load schedule of any complexity based on previous experience. In addition, there is no need to build a mathematical model of the network under consideration. For predictions with reasonable accuracy, it is sufficient to have retrospective data of the measured load values [7, 10].

To solve the load prediction problem, the following types of neural networks are mainly used: multi-layer perceptron, radial-basis functions and linear networks. For each practical task, the quality of prediction in these models is estimated by the MAPE (Mean Absolute Percentage Error). In most cases, the best results are obtained by a multilayer perceptron.

The quality of neural network prediction is influenced by various factors, such as adjusting the model parameters and training algorithms.

The training of a multilayer perceptron is most often carried out by the method of Back Propagation of the Error (BPE). Its advantages include ease of implementation and speed, disadvantages - the ability to find a locally optimal solution instead of global, the sensitivity to the order of training examples.

In addition, evolutionary algorithms, such as the Genetic Algorithm (GA) and the Particle Swarm Optimization method (PSO), have recently been used to train the network. Particle swarm algorithm quickly converge to the best solution, the method is easy to implement and is very effective. However, with incorrect selection of optimization parameters, the training time of the network increases [6].

3 The Used Model of Neural Network

Today a multilayer perceptron is one of the most widely used neural network models because of its ability to reflect complex nonlinear relationships between input and output parameters. The network consists of several layers of neurons and weight coefficients, reflecting the connections between them. The transmission of information is carried out based on direct dissemination. The model of the neural network implemented in this paper is shown in Fig. 2.

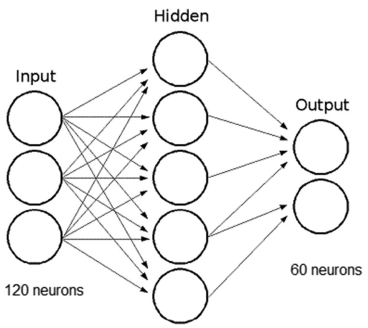


Fig. 2. Structure of used model

Let us denote the number of layers and neurons in the layer. Input layer: N_I neurons; N_H neurons in each hidden layer; N_O output neurons; x – vector of input network signal, y – vector of output signals.

Operation of multilayer perceptron is described by the formulas 1, 2, and 3:

$$NET_{jl} = \sum\nolimits_i w_{ijl} x_{ijl} \tag{1}$$

$$OUT_{jl} = F(NET_{jl} - \theta_{jl}) \tag{2}$$

$$x_{ij(l+1)} = OUT_{il}, \tag{3}$$

where i – number of the input, j – number of neuron in the layer, l – layer number.

Then x_{ijl} – i input signal of j neuron in the layer l; w_{ijl} – weight coefficient i input of neuron number j in layer l; NET_{jl} – the signal of j neuron in the layer l; OUT_{jl} – output signal of the neuron; θ_{jl} – threshold neuron j in layer l.

Each layer calculates the non-linear transformation of a linear combination of the previous layer signals. Hence it is clear that the linear activation function can be applied only for those network models, which do not require the series connection of the layers of neurons to each other. For multilayer network activation function must be non-linear, because the equivalent of a single-layer network can be built, and layering is unnecessary. If to apply a linear function of activation, each layer will result in an output linear combination of the inputs. The next layer will give a linear combination of the previous output and is equivalent to a linear combination with other factors, and can be implemented as a single layer of neurons.

To predict the electrical load one hour ahead, a model with the following parameters was used:

- the number of layers – 3 (input, hidden and output), because this configuration is the most common in solving forecasting problems;
- the number of neurons in the input layer – 120 (120 values of the actual power chart in the previous 2 h, each point represents one minute measurement);
- the number of neurons in the hidden layer will be defined in Sect. 4;
- the number of output neurons – 60 (forecasting done for 1 h ahead, each point represents one minute measurement);
- the function of activation of the hidden layer is tangential and given in formula (4), since it has the property of amplifying weak signals better than strong ones, and prevents saturation from strong signals, hence this function solves the dilemma of noise saturation of the Grossberg;

$$f(n) = \frac{2}{1 + e^{-2n}} - 1 \tag{4}$$

- training algorithm (a) back propagation of the error (b) swarm of particles.

The main idea of back propagation of error is simple. The network is trained on a certain number of training pairs. The output vector is presented, the network output is

calculated and compared with respective target vector, difference (error) is fed via a feedback, and weights are changing in accordance with the algorithm seeking the minimum of the error. The vectors of the training set are presented sequentially, errors are calculated and weights for each vector adjust as long as a teaching error across the array reaches an acceptably low level.

According to the method of least squares, minimized error function of neural network is:

$$E(w) = \frac{1}{2}\sum\nolimits_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2, \tag{5}$$

where $y_{j,p}^{(N)}$ – the actual output state of the neuron j of output layer N when applying its inputs p-image; $d_{j,p}$ – the desired output state of the neuron.

In this paper considers not only training by back propagation, but also particle swarm algorithm.

Searching of an optimal solution is not from the only one of the predetermined points, but from multiple randomly found points. Each particle "remembers" the coordinates of the best solution and found it, and find the best of all the particles found solutions. The algorithm searches for optimal solution iteratively moving the particles in the solution space. Each particle is attracted to the best solution found adjacent particles, and to the best solution found herself.

The main settings of the back propagation error algorithm are learning rate, the number of iterations, momentum.

The main parameters of particle swarm algorithm is the number of particles in the swarm, the number of iterations, the self-acceleration coefficients and the value of velocity changes.

These parameters for both of algorithms was settled in the Sect. 4.

In this study, the mean absolute error in percentage (MAPE) is used to evaluate the training and test samples:

$$MAPE = \frac{1}{N}\sum\nolimits_{i=1}^{N} \frac{\left|P^{forecast} - P^{real}\right|}{P^{forecast}} \cdot 100\%. \tag{6}$$

4 Results

In this study, the results of load measurement in the distribution networks of the Moscow region in 2008–2010 with the averaging interval of 1 min were used. To model the mode, the MATLAB2017a environment was used.

As far as the aim of this paper to create the neural network, which will adequate forecast the load curve for particular system, the network was trained on archived load data, which were provided by PAO «MOESK». After that, it was tested on a sample from data that was not used in the training samples.

Based on the lowest MAPE the number of hidden neurons for training method back propagation of the error is set to 38 using simple increasing of the hidden neurons

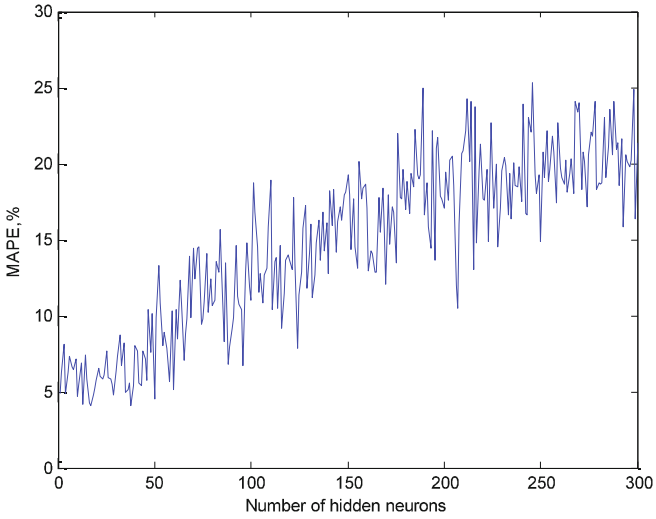


Fig. 3. Results of determination of number of hidden neurons for training method BPE

number. In each case network was trained and the resulting value of error was evaluated. MAPE in this case is equal 4.06%. The results of the experiment is shown on the Fig. 3.

Based on the lowest MAPE the number of hidden neurons for training method particle swarm is set to 27. Test was made in the same mode as for BPE algorithm. MAPE in this case is equal 1.298%. The results of the experiment is shown on the Fig. 4.

Figures 5 and 6 show the same time interval on January 13, 2010 from 01:00:00 to 02:00:00. The green line denotes the actual values of the load during this period; the blue line indicates the values predicted by the neural network.

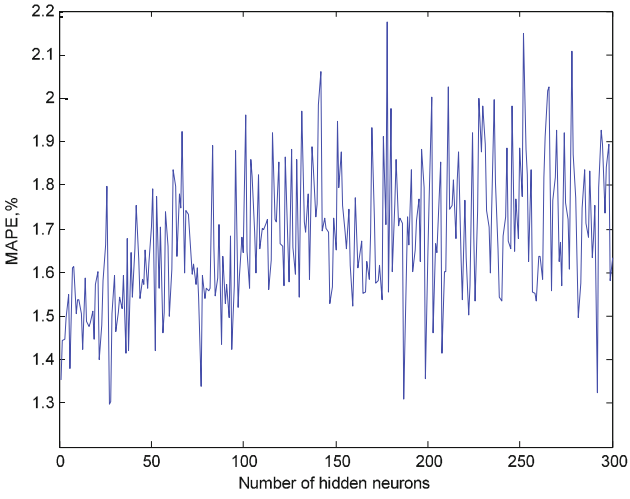


Fig. 4. Results of determination of number of hidden neurons for training method PSO

In Fig. 5 a graph of the predicted values of electrical load based on the created software algorithm, trained by BPE is presented. Algorithm in this case does not catch the overall trend gives 4.06% of MAPE.

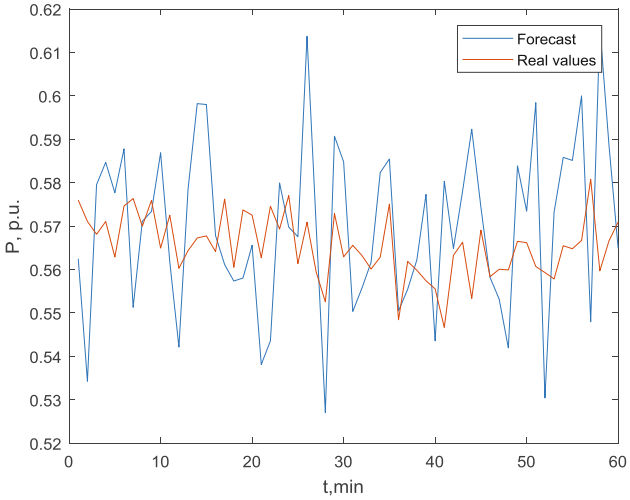


Fig. 5. The results of forecasting for the ANN trained by BPE

In Fig. 6 a graph of the predicted values of electrical load based on the created software algorithm, trained by PSO is presented. Algorithm in this case catches the trend and small deviations of the load and gives 1.298% of MAPE.

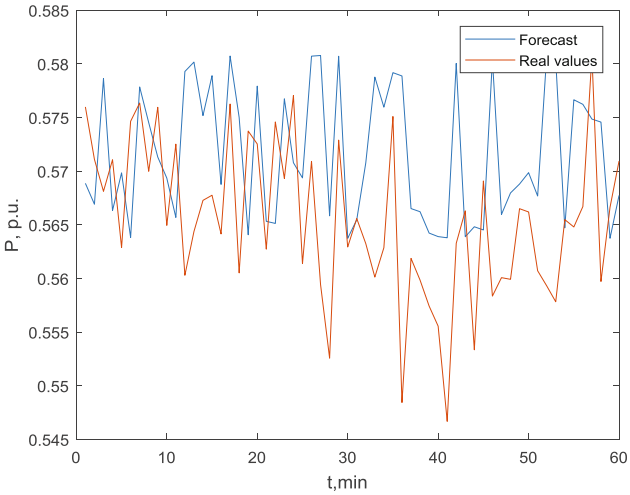


Fig. 6. The results of forecasting for the ANN trained by PSO

After the selection of the optimal number of hidden layer neurons it is necessary to select other parameters associated with the method of training the neural network. Backpropagation can be adjusted by changing the speed of learning, that is, setting up a step gradient descent. The date and number of epochs also may be changed. Momentum is used to accelerate the training process and in this paper is equal to 0,5. The other parameters were settled using the trial and error method. The specific points for each parameter were used. For the learning rate the range changed in [0,01; 0,5] interval. The smallest values leads to the «grid paralysis», the biggest values lead to the non-ability of the network to find the global minima. The number of epochs insignificantly influence on the training time and was changed in the range [50; 500]. The selection can be done using special algorithms such as self-organizing maps, genetic algorithms, but it is not the purpose of this paper and could be done for improving of the performance in next studies. The best results of setting the parameters is shown in Table 1.

Table 1. The results of STLF.

Name of training method	MAPE %	Note
ANN (BPE)	1.79–6.68	Tested on the data, which were not used for configuring (number of hidden neurons 38, learning rate 0.1, number of epochs 300)
ANN (PSO)	1.61–4.68	Tested on the data, which were not used for configuring (number of hidden neurons 27, population size 20, number of iterations 75)
ANN (BPE)	1.69	The best variant during configuring (number of hidden neurons 38, learning rate 0.1, number of epochs 300)
ANN (PSO)	1.44	The best variant during configuring (number of hidden neurons 27, population size 20, number of iterations 75)

Selection of the optimal parameters for changing the velocity of the particles is the subject of separate studies. In this paper are accepted parameters c1 = 1,49445, c2 = 1,3, found by the other authors and providing the best results for this method of training for the other problems such as pattern recognition. To this training algorithm, the population size is important, it directly affects the speed of finding the optimal solution, because the small size of the population make the search for the optimum longer. The population size were changed in the range of [5; 150]. The iterations number influence slightly on the training time and was settled in the range [50; 500]. The selection was done using trial and error method. The following Table 1 gives a summary of the best variant of setting parameters.

5 Conclusions

- Figures in the table show that, using a “particle swarm” training algorithm neural network is able to reduce the average prediction error by 0.18–2% and to reach a desired value.

- Forecast result in this case not only shows the general trend, but also reflects the changes in the local load curve. In addition, this method of training has a great speed, which is important for operating in real time.
- Back propagation algorithm for training the neural networks is simple to implement and configure, but for complex load profiles is inapplicable because it does not allow to achieve the required accuracy of prediction (1.79–6.68%).
- The algorithm "swarm of particles" for training the neural networks is slightly more difficult to implement and configure, but gives an acceptable error value (1.61–4.68%).
- Comparative analysis of training algorithms showed that for complex, non-linear load curves for neural network training is necessary to use evolutionary algorithms, as they give the best results of the absolute values of errors and training time.
- The prospects for further development include finding methods to reduce training time, and reducing the MAPE error, using of intelligent techniques for defining optimal parameters of the network (self-organizing, setting using genetic algorithms).

References

1. Vanin, A., Aleshin, S., Nasirov, R., Novikov, D., Tulsky, V.: Investigation of voltage control at consumers connection points based on smart approach. MDPI Inf. **7**(3), 42 (2016)
2. Rutkovskaja, D.: Neural Networks, Genetic Algorithms and Fuzzy Systems. Gorjachaja linija—Telekom, Moscow (2006)
3. Voronov, I.V.: Improving the efficiency of operation of the enterprises of power supply systems through the integrated use of smart grid and neural networks. Vestnik Kuzbasskogo gosudarstvennogo tehnicheskogo universiteta **2**(90), 63–66 (2012)
4. Lis, R.A.: Application of data mining techniques to identify critical voltage control areas in power system. In: Proceedings of the Third International Conference on Computational Collective Intelligence: Technologies and Applications, ICCCI 2011, vol. I, pp. 152–162. ISBN 978-3-642-23934-2
5. Badar, I.: Comparison of conventional and modern load forecasting techniques based on artificial intelligence and expert systems. Int. J. Comput. Sci. **8**(5), 504–513 (2011)
6. Roy, A.: Training artificial neural network using particle swarm optimization algorithm. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(3), 430–434 (2013)
7. Haykin, S.: Neural Networks: A Comprehensive Course, 2nd edn. Viliams, Moscow (2008)
8. Buhari, M.: Short-term load forecasting using artificial neural network. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. I, Hong Kong, 14–16 March 2012
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: 2012 Proceedings of IEEE International Conference on Neural Networks, Perth, Australia. IEEE Service Center, Piscataway (1995)
10. Khothanzad, A., Hwang, R.C., Abaye, A., Maratukulam, D.: An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities. IEEE Trans. Power Syst. **10**(3), 1716–1722 (1995)

Mathematical Modeling in Computer System Design

The Experiment with Quality Assessment Method Based on Strategy Design Pattern Example

Rafał Wojszczyk(✉)

Faculty of Electronics and Computer Science,
Koszalin University of Technology, Koszalin, Poland
rafal.wojszczyk@tu.koszalin.pl

Abstract. In software engineering, there are many methods and good practices which aim at ensuring quality of developed software. One of these practices is using design patterns. The aim of the article is to show the example of quality assessment method of the implementation of the strategy design pattern and benefits which are given by this method. The analysis of the results of method can help the leader of developer team in case of getting decision of the repairing defects in implementation of design patterns, that's lead to lower costs connected with the development and fixing the defects in software.

Keywords: Design patterns · Software quality · Software development · Strategy pattern · Experiment with students

1 Introduction

The commercial software production is very often related with the costs which are paid by the vendor. Particularly when he is producing software for business, which has to be constantly developed to ensure competitiveness towards other products on the market. And when the vendor has to adapt his product to the changing user rules or laws. Research and development in the software engineering give many solutions which help to reduce the costs of production [7], i.e. solutions helping at the beginning of the production process to lower the estimation costs of effort changes in the software [10]. During the whole production process it is popular to use agile factory methods in small teams [6], which also helps to lower the costs connected with managing the software development. There are also solutions used in the source code of created product, ready components, architecture patterns (e.g. MVC or MVP) and design patterns. Mentioned kinds of patterns need own contribution and proper preparation from people who are implementing them.

Work done by people, in this case the source code created by programmers, can include defects. In case of design patterns the defects noticeable by the program user can be omitted, whereas defects in design patterns will be noticeable first of all for the vendor. The vendor cannot be sure if despite the defects lower cost of development of the software will be assured. In other words, the vendor is searching for an answer to

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_10

his question: Whether the implementation quality of the given design pattern will assure him benefits that are expected from this pattern?

The aim of the article is to present the experiment with using method of quality assessment of implementation of design patterns, based on the example of the Strategy design pattern.

In the second chapter of the article it was briefly explained what is the implementation of design patterns and chosen methods related with searching of the occurrence and verification of the implementation of the patterns were discussed In the third chapter there were described used formal models and the process of using of the method to verifying the implementation of design patterns. The fourth chapter presents the strategy pattern, assumptions of the experiment, results and their interpretation. The fifth chapter is the summary of the work.

2 Design Patterns and Quality

2.1 Quality of Implementation of Design Patterns

The software vendor, through the implementation of the design patterns, aims at achieving chosen benefits and for majority of the design patterns such are [4]:

- reusable trusted solutions,
- way of avoiding chosen defects,
- lower cost of development and fixing the defects,
- bigger legibility of the code,
- substitute for documentation.

The benefits emerging from the implementation of the design patterns contribute to lower the costs of production e.g. new software developer, who knows the design patterns, faster absorb the newly introduced source code (which also contains patterns), than code without patterns. Faster absorbing the code means that new software developer will start to work with maximum efficiency earlier and the minimum efficiency time will be shorter.

The occurrence of the implementation of the design pattern in the source code does not mean that this code ensure mentioned benefits. To be sure, there is a need to check if the quality assessment of the implementation is evaluated by using proper metrics in accordance with the criterion. The criterion of evaluation is a perspective of the expected benefits. In other words quality assessment of the implementation of a particular design pattern means a number of actions which must be taken to show the analysed software code's conformity with rules and principles of implementation adopted for a particular pattern. To make the evaluation trustworthy it should consider many factors connected with the implementation of the patterns. Basic element in the source code which is evaluated is the structure creating the pattern, e.g. class with the private constructor in the Singleton pattern, field affording to share the Singleton instance etc. Alongside the code which creates the pattern there is a code which is passive in the pattern [3], usually it is processing of the data. The context of the implementation of the pattern is essential but in many times omitted. Elements of the

code, where the existence of the pattern is called is the context of the implementation. The metrics in the completed model of evaluation should describe every mentioned elements.

Design patterns are gladly used by software developers both in big and small productive teams. Small teams which work by agile methods often do not dispose of complete documentation of the project of manufactured product [6], therefore patterns are implemented on the basis of knowledge and experience of the members of the team. However, in many cases, decision about solving the project defect through the implementation of design pattern is made quickly during morning meetings. Mentioned difficulties are determining the problem, so the vendor does not know if what he expects from the implementation of the design patterns will be possible. Therefore from the vendor's perspective, a need for quality assessment of the implementation of design patterns appears, which will help to solve mentioned problem.

2.2 Detection for the Occurrence and Verification of Design Patterns

Classic models of quality assessment and metrics are known since many years in the software engineering area. Unfortunately, used for evaluation of the implementation of the design patterns are not sufficient or provide wrong results [5]. Other popular solutions that help in creating software, e.g. Model Driven Architecture, cannot be used by agile teams because of lack of complete project documentation.

Studies directly connected with design patterns very often concern searching of occurrence of patterns in source code only. Information that particular part of the source code contain some amount of patterns is not providing the correctness of implementation, to be precise - correctness of the documentation [8, 9]. Such solution will fail again in case of agile vendor teams. There are undertaken studies, which aim at showing correctness of structural implementation of patterns [1], which also brings too little information and does not include the evaluation.

3 The Method of Quality Assessment of Implementation of Design Patterns

3.1 Model Structure

The method based on model, which model has been described i.a. in: [12, 13], where the data models has been explained. In the model it is assumed that the source code is imitated (managed code type .NET or Java, to be precise) into formal model of the program. Templates of the design patterns (i.e. information about implementation variants) were divided into general characteristics (or features [11]) and provided with particular metrics, analogically to ISO9126. The norm is described by a tree structure which includes characteristics and in those, recursively, there are subcharacteristics. Characteristics-leaves have metrics - functions assigned to them, which define certain values on the basis of measurable software attributes. The definition of characteristics is informal - it expresses certain intent, while the metrics are formalised.

On the basis of the model it is possible to indicate the quality assessment of implementation of design patterns in the source code and pointing, which elements of the implementation should be corrected to achieve the highest quality in particular evaluation criterion.

3.2 Model of Computer Software

The most information on programming is included in the source code, which simultaneous shortcoming is a physical representation - these are properly catalogued text files which are more difficult to analyse than a structured formal representation. A formal model of the program provides the automated processing of data concerning the occurring implementation and, additionally, it enables one to remove useless code and unwanted information (e.g. comments written in a natural language, unit tests code).

Basic attributes of object oriented programming were imitated as set: modifiers *MOD = {static, partial, sealed, final, ...}*, access modifiers *AMOD = {public, private, internal, protected}*, kind of type *KOT = {class, interface, value, generic, array}*, kind of instruction *KOI = {declaration, initialize, called, call}*. The code of analysed computer program was mapped into set of types *T*:

$$T = \{t_i | i = 1, \ldots, t,\}, \tag{1}$$

where t – number of types, t_i denotes concrete type. Each type t_i:

$$t_i = (FIE_i, PRO_i, MET_i, mod_i, amod_i, name, kot_i), \tag{2}$$

where *FIE, PRO, MET* are the sets correspondly to fields, properties and methods included in t, $mod_i \in MOD$, $amod_i \in AMOD$, $name \in alphanumerical$, $kot_i \in KOT$. On account of further example, sets *FIE* and *PRO* do not require explanation. The set of methods MET_i:

$$MET_i = \{met_{i,j} | j = 1, \ldots, met\}, \tag{3}$$

where *met* – the number of method in t_i, $met_{i,j}$ denotes *j-th* method in t_i. Each method $m_{i,j}$:

$$m_{i,j} = (MP_{i,j}, rt_{i,j}, ctor_{i,j}, MB_{i,j}, mod_{i,j}, amod_{i,j}, name), \tag{4}$$

where $MP_{i,j}$ – the set of method parameters, $rt_{i,j}$ – returned type, $rt_{i,j} \in T$, $ctor_{i,j}$ denotes constructor (true if method is a constructor, false for otherwise), $MB_{i,j}$ – the set of instructions in method body, $mod_{i,j} \in MOD$, $amod_{i,j} \in AMOD$, $name \in alphanumerical$. Method body is a set of instructions $MB_{i,j}$:

$$MB_{i,j} = \{ins_{i,j,k} | k = 1, \ldots, ins\}, \tag{5}$$

where *ins* – the number of instructions in *k-th* method in type t_i, $ins_{i,j,k}$ – concrete instruction. Instruction $ins_{i,j,k}$:

$$ins_{i,j,k} = \left(koi_{i,j,k},\ it_{i,j,k}\right), \tag{6}$$

where $koi_{i,j,k} \in KOI$, $it_{i,j,k}$ instance of type, $it_{i,j,k} \in T$.

3.3 Use of the Method

Using the model defined in Sect. 3.1, the author's own method of quality assessment of implementation of design patterns is suggested. The main stages of the method of quality assessment of implementation of design patterns are shown in the Fig. 1. The first stage is used only once with the introduction of the method to teams Stage I, II and III should be made obligatorily, while stages IIa and IV (which is not marked in the picture) are optional and depend on the needs of the team.

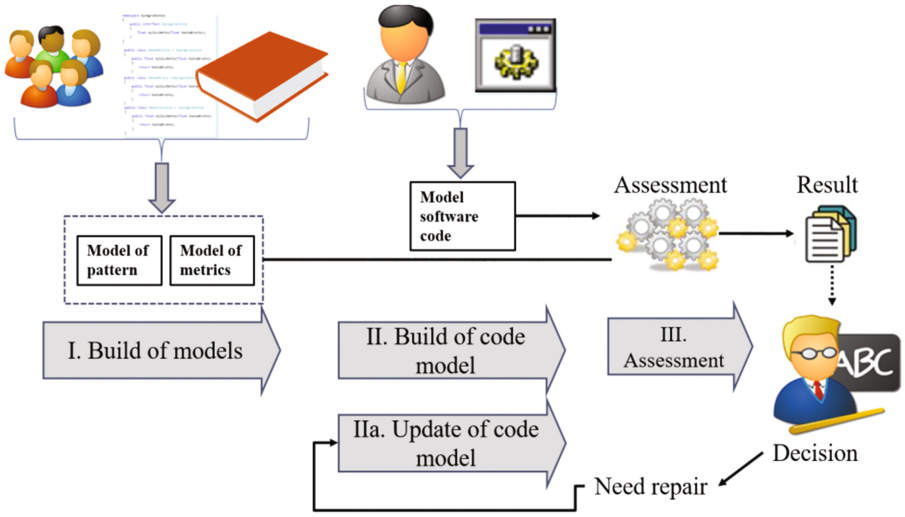


Fig. 1. Diagram of the method used to assessment the quality of implementation of design patterns.

Stage I – build of the pattern and metric models. This stage assume creating pattern and metric models according to previously presented structure methods. Responsibility for creating is on the leader of the team because in most cases he has the biggest experience and knowledge of design patterns.

Stage II – build of the program model. The second stage of suggested method appears cyclically, at least once during iteration of agile methodologies, every time before the III stage (automation required). Stage IIa, or actualisation of program model appears instead of stage II, when there is a need to renew the assessment.

Stage III – assessment. Occurrence of stage III in the software development cycle depends on the preference of the project team. Required time is one dedicated for inspections and code improvement. The stage of the assessment can be used more frequently rather than once during iteration, however it should occur after making the

potentially ready part of the source code. Person required and responsible for making the assessment is leader of the team, because he has the biggest experience. On the basis of his experience, the leader makes decisions weather the quality assessment of design patterns (made by team the leader is responsible for) is sufficient in the accepted quality assessment or it needs improvement. In case of required improvement of implementation of design patterns it is necessary to get suggestions from the metrics model, which will show the programmers parts that need changes and required type of changes. It is very important, that the assessment and improvement were made in the time of iteration, in which design patterns were implemented, i.e. when programmers remember previously made code.

Stage IV – development and modification of models from stage I. The last stage of the method occurs irregularly, i.e. when there is a need. Stage IV rely on making actions from the stage I fragmentarily, i.e. development or modification of pattern and metric models. The factors that impose the occurrence of this stage are i.a. development of the abilities of project team, which are increasing along with getting new experiences.

4 Process of Quality Assessment

4.1 Assumptions of the Experiment

Conducted experiment relied on simulating a part of iteration of development process, in which Strategy pattern software is developed. The main aim of this experiment was showing that time of software development by adding new operation to the pattern Strategy will be shorter in case of implementation of pattern according to directions resulting from the method of quality assessment of patterns, in regard to the time needed for development, when implementation of pattern contain defects. In the experiment it was not able to map the environment typical as in the project company, so it was necessary to make many simplifications.

Simulation consists of two processes, which are shown in the Fig. 2. In the first process, students took part only in iteration B. Iteration A consisted of making software along with few defects in implementation of strategy pattern. To begin working on code not until iteration B there was simulated passing time, i.e. code in the iteration B was a so-called legacy code. During the iteration B, students had to add a new operation to the strategy and fix defects that were noticed before. Same task was submitted in iteration B, however in case of the second simulation process students also took part in A3. Iteration A1 in both processes is the same, differences occur in time of code inspections, were additionally assessment of implementation of design patterns was made, then improvement of implementation according with worked out method – which was also made by students. After that, they begin to work on iteration B, however in this case implementation of the strategy pattern did not consist of defects as in iteration B.

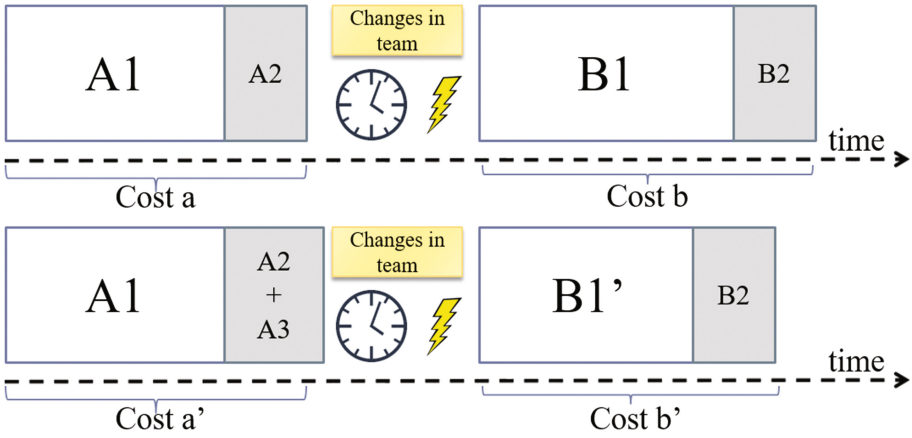


Fig. 2. Simulation scheme

4.2 Strategy Design Pattern

Benefits of Implementation

Strategy design pattern is one of the popular patterns [2]. The aim of this pattern is [4]: defining algorithm family, encapsulation each one of them and enabling their replacement. This pattern is useful in problems when there is a need to achieve one operation in many different ways. Popular example in literature is encapsulation of different sorting algorithms and their replacement. For the vendor the strategy pattern means following benefits:

- Lower development cost (or bigger accuracy of estimation of the cost) by adding another operations realized with the help of the strategy pattern,
- Repeatability of operations, i.e. operations realized with the help of the strategy pattern will generate the same result regardless of the modulus program in which they will be run in,
- Lower cost of fixing the defects, i.e. the defect should be fixed in one operation, no matter of the cases of use of this operation.

Characteristics

The template of the Strategy design pattern was imitated in proposed model as a group of characteristics "c": c_1 – interface declaration, c_2 – declaration of operation, c_3 – interface realization, c_4 – implementation of operation, c_5 – context of the implementation of operation, c_6 – initialization and choice of strategy. Mentioned characteristics were detailed by 15 metrics $m_{i,j}$, where i stands for characteristics, and j means another metric. More important metrics in established quality assessment are enclosed in Table 1. In established quality assessment the highest result is required. Every lapse from such result indicates additional costs for the vendor.

Table 1. Selected metrics of strategy design pattern.

Metric	Values (points)
$m_{1,1}$ – access modifier	4 – public, 3 – internal or empty, 2 – protected, 1 – private
$m_{1,2}$ – kind of type	5 – interface, 2 – abstract class, 1 – class
$m_{3,1}$ – kind of type	3 – class, 1 – interface
$m_{1,2}$ – dependencies	5 – if each class realizes interface, 1 – if some class not realizes interface, 0 if any
$m_{4,1}$ – realization of an operation	5 – if class realizes (or override) operation, 3 – if class has similar operation, 0 – no operation
$m_{6,1}$ – strategy declaration and initialization	5 – declaration as interface and next initialization of concrete strategy, 4 – declaration and initialization inline, 1 – no declaration
$m_{6,2}$ – covering of initialized strategy	3 – if all strategies are initialized in one code unit, 1 – if only some

4.3 Quality Assessment of Strategy Pattern

Program used in the research is an application project created by students, which is navigating the stepper motors. The Strategy pattern has been used to adapt the application to different actuators, i.e. every student has been using different microcontroller board and developed application should run identically, independently to the equipment. Operation shared by microcontroller boards is: `byte RunCommand(byte cmd, byte args)`, which should be declared in the interface DeviceStrategy. RunCommand operation accepts two byte type parameters and gives back operation result also byte type. For the experiment there were implemented defects: d1 – strategy declaration is a class instead of interface, d2 – one of the classes inherits from strategy declaration (primary class) without implementation of operation, so with calling of this strategy the empty primary method is made, d3 – one of the calling of specific strategy consists of class declaration and direct initialisation (so-called inline).

In the first simulation process every participant received program code with defects and two tasks to solve, which was the simulation of iteration B1 from the 2 picture. For further description code in this form was called C1. As part of first task the participants had to add the code responsible for support of new microcontroller patterned upon legacy code. Besides, the added code should be integrated with the rest of the program (context of implementation). Within the second task students had to fix two defects: d2 and d3.

In the second simulation process participants received the same source code as in the beginning of the first process, i.e. C1. Code C1 as a result of quality assessment of implementation of pattern got 182/310 points. Decomposing the summary result on the particular component grades, there were noticed metric results, which were in the area of acceptable results in the accepted assessment criteria. Based on obtained information every participant made initial improvement of the code, which is iteration A3 from the picture 2. Improved code, hereinafter called C2, got summary assessment result of 310/310, which is enough to begin with further part of experiment.

Comparison of work made by every participant is presented in the Table 2. It follows that when enhancing the development process with proposed method temporal profit is obtained, despite additional time dedicated in iteration A3. Effort of making the assessment is negligible because it can be largely automated.

Table 2. Experiment results

Participant	Timing of the tasks C1	Timing of the tasks C2	Profit after using the method
1	3,5	2	43%
2	5,5	4	28%
3	7	6	14%

Relatively the highest profit, up to 43% got the first participant. It is because the fact that the participant knew the Strategy pattern before. Better quality in this case contributed to easier understanding the rule of action and knowing the structure of legacy code.

Easy to notice is also that the other two participants have lower skills than the first one, which resulted in bigger effort needed to make the same tasks. Work made by them should be evaluated by suggested method, under control of a person with more skills, which in case of delegating tasks to people with lower skills, gives an opportunity of avoiding different, unexpected results, which later get to people with more skills in order to further development.

Presented results, in the development cost perspective, are showing temporal profit of average 25% with using the method. This result confirm that suggested method provides solution of problem depicted previously and makes profits related with implementation of design patterns possible. It should be pointed out that in presented results the costs of construction of models and introducing it is not taken into consideration, because it is non-recurring cost. However it is a very laborious process, which set limitations of the method and requires to be automated in the future. In addition, complexity of the model is a disadvantage which can lead to rearranging the contradictions of data included in the model.

5 Summary

In the article chosen problems related to costs of software development were briefly discussed and there were also presented solutions supporting to lower the costs. On the example of strategy pattern there were presented benefits introduced by this pattern.

Presented results, in the development cost perspective, are showing temporal profit of average 25% with using the method. This result confirm that suggested method provides solution of problem depicted previously and makes profits related with implementation of design patterns possible.

Further research provide development of the model for information about the quality to give more details and to be comprehensible for the vendor. There is also expected assessment of a few patterns in different iterations of software reference.

References

1. Blewitt, A., Bundy, A., Stark, I.: Automatic verification of Java design patterns. In: Conference on 16th Annual International Automated Software Engineering, 2001 Proceedings, ASE 2001 (2001)
2. Czyczyn-Egird, D., Wojszczyk, R.: Determining the popularity of design patterns used by programmers based on the analysis of questions and answers on Stackoverow.com social network. In: 23rd Conference Computer Networks. Communications in Computer and Information Science, vol. 608, pp. 421–433. Springer International Publishing, Brunów (2016)
3. De Lucia, A., et al.: Design pattern recovery through visual language parsing and source code analysis. J. Syst. Soft. Arch. **82**(7) (2009). Elsevier Science Inc., New York
4. Gamma, E., et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Boston (1994)
5. Khaer, M.A., et al.: An empirical analysis of software systems for measurement of design quality level based on design patterns. In: 10th International Conference on Computer and Information Technology, Dhaka, pp. 1–6 (2007)
6. Martin, R.C., Martin, M.: Agile Principles, Patterns, and Practices in C#. Prentice Hall, New Jersey (2006)
7. McConnell, S.: Code Complete: A Practical Handbook of Software Construction, 2nd edn. Microsoft Press, Washington, DC (2004)
8. Mehlitz, P.C., Penix, J.: Design for verification using design patterns to build reliable systems. In: Proceedings of the Workshop on Component-Based Software Engineering (2003)
9. Nicholson, J., et al.: Automated Verification of Design Patterns: A Case Study, Science of Computer Programming, vol. 80, pp. 211–222. Elsevier, Amsterdam (2014)
10. Plecka, P., Bzdyra, K.: Algorithm of selecting cost estimation methods for ERP software implementation. In: Applied Computer Science, vol. 9, no 2, pp. 5–19. Politechnika Lubelska, Lublin (2013)
11. Rasool, G.: Customizable feature based design pattern recognition integrating multiple techniques. Ph.D. thesis, Technische Universitat Ilmenau, Ilmenau (2010)
12. Wojszczyk, R., Wójcik, R.: The model of quality assessment of implementation of design patterns. In: Advances in Intelligent Systems and Computing, vol. 474, pp. 515–524. Springer International Publishing, Switzerland (2016)
13. Wojszczyk, R., Khadzhynov, W.: The process of verifying the implementation of design patterns – used data models. In: Advances in Intelligent Systems and Computing, vol. 521, pp. 103–116. Springer International Publishing, Switzerland (2017)

Development of a Support System for Managing the Cyber Security of Information and Communication Environment of Transport

Valeriy Lakhno[1], Alexander Petrov[2(✉)], and Anton Petrov[3]

[1] European University, Kiev, Ukraine
[2] AGH University of Science and Technology, Krakow, Poland
aspl951@gmail.com
[3] National Aviation University, Kiev, Ukraine

Abstract. The operation of critical computer systems (CCS) in industry, energy, transport and communications, etc. requires constant monitoring of cyber threats, as well as vulnerabilities in the technical components and the software. The information object cyber security (CS) operational management system and the formation of the protection methods rational sets model which is based on a morphological approach is developed. This model allows us to generate different variants of protection sets that are compliant with a critical computer system (CCS) of transport branch taking into account morphological matrices for each security perimeter prepared with the intelligent decision support system (DSS or intelligent decision support system – IDSS). It will find an optimal variant of the cyber security perimeter sets using an CCS that maximizes the correlation of a consolidated figure of "information security" (IS) to consolidated figure "costs". A program set for IDSS in circuits of organizational-technical and operational management of the CCS security system is developed. It is proven that using the developed IDSS allows us to reduce the cost of developing an information security system and to shorten the time for informing some responsible individual about information security incidents.

Keywords: Information security · Information management · Transport · Decision support system · Mathematical model

1 Introduction

Active expansion of information-communication environment of critical computer systems (industry, communication and transport, etc.) (CCS), especially in the segment of mobile, distributed and wireless technologies, accompanied by the emergence of new threats to cyber security (CS), as evidenced by the growing number of incidents related to information security and protection of information and discovered vulnerabilities in CCS (or information objects (IO)) and automated control systems. The threats are real [1]. The rapid increase of incidents in the field of informational security (IS) has shown that existing information security systems (ISS), which are built on the

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_11

basis of known cyber-threats and emerging attacks, are not always effective in cases of new cyber-attacks (C-A) which are created against the widespread enterprise information system, automated control systems in electronics, industry, transport, the banking system etc. [2].

The structure of the technological complex of critical computer systems (CSS) of transport may include various technical systems and tools [1]. Information security of CCS of transport has never been released as a separate type of national security. Moreover, the IS and CS of CCS cannot exist outside of national security. As part of a whole, it carries heredity conceptual approaches to ensure security at the micro and macro levels, continuity of relationships, common principles and methods. Moreover, CS of CCS of transport usually has its own characteristics and specific, reflecting the industry direction and defining its place, role and importance in the structure of national security.

2 Problem Statement

For the successful usage of modern information-communicative systems (ICS) it is necessary not only to know how to manage all the functional resources but to create an effective information security management system (ISMS). As management objects – ISMS are difficult organizational-technical structures, which function in conditions of uncertainty. Effective management of such systems has to be based on innovative ITs aided by decision support which considers both IS and CS.

One of the variants of this solution is the usage of a decision support system (DSS) in CS on the basis of intelligent information technologies. Research into improving existing and the development of new methods, mathematical models and software for operational management of information protection (IPR) in IO, particularly in conditions of uncertainty, inconsistency and lack of knowledge about ICS status becomes highly relevant.

Goal of the research – developing a cyber-threat counterwork model using DSS, choosing rational variants of reactions on the occurrences in CS, and taking into account current operational IO data.

It is necessary to solve the following tasks to reach the goal of the research:

1. To develop a model of operational management of CS IO which allows us to increase IS management efficiency in conditions of information environment status uncertainty, and efficiency of the ISS rational structure planning process.
2. To develop the intelligent decision support system (IDSS) program of IO cyber-security management and to investigate the efficiency of the offered model.

3 Review of the Literature

The structure of the technological complex CCS of transport sector may include various technical systems and tools [1–4]. The increasing number of IS and CS threats has given rise to the surge of research in the field of development of uncovering and

preventing C-A systems [5–8], and also DSS [9, 10] and expert systems (ES) [11–13] in this field. But the majority of existing articles, devoted to the problem of DSS and ES, address only the basic features of C-A. Publication analysis [14, 15], allows us to uncover the increasing popularity of ISS risk assessment automated methods [16] and program sets of IS and CS risk management [17]. Unfortunately, this set of instruments of this approach has not been developed enough to be broadly applied. It was mentioned in the works [18, 19] that ISMS, in which intelligent technologies of cyber-threats identification and reacting to occurrences of IS breaches are realized, are products of private companies, and that a customer in general doesn't have any information about methods and models of leading effects forming in systems [20]. In the works [21, 22] all the disadvantages of many DSS and ES systems in the field of IS are shown. It is shown in the works [21, 23, 24] that it is appropriate to equip existing DSS and ES in field of IS (excluding tasks of cyber-security management) with functional models that allow us to increase efficiency of enumeration and investigation of illegitimate interferences to the work of ICS crimes.

On the opinion of some authors [16–18, 21–24], existing standards in the field of IS management don't form effective methods of CS management.

In such a way, according to the disputes in publications [12, 14, 21], dedicated to the potential of using integrated DSS and ES [9–11, 13, 20] in ISMS, which realize a precautionary strategy of IO CS [19, 24], the task of developing methods, models and applied software for using them in practice in intelligent support of ISS rational structure planning and the task of assessment and prediction of IS and CS risks and also of IPR management in conditions of uncertainty of potential cyberterrorists influences became relevant.

A model of a problematic situation in the field of CS consists of the three interacting systems: firstly, ISS which potentially can lead to problems with IS; secondly, controlling system with IS and CS (cyber-security), which is being developed for solving IPR problems; thirdly, the environment in which ISS operates. In this case the environment is understood to be a huge number of potential vulnerabilities and threats for IS and CS.

There is one main problem creating the CS – development of the threat model [11, 19, 25], which is connected to the specification of a management object interaction – ISS IO with the environment. IDSS, which develops a threat model building method, is based on a qualified scheme of goal-oriented destructive influences on IS and CS [26–28].

4 Materials and Methods

For managing of ISS of transport is characterized by the following kinds [2, 4]: onboard equipment installed on moving objects ICS; means mounted on fixed infrastructure; remote-controlled executive and indicative devices; servers for processing and storage; situational, operational and dispatch center's; means of communication; information and telecommunication equipment, providing a secure information interaction with external information systems.

A generalized architecture of ISMS and CS is offered according to the results of the control strategy in conditions of uncertainty analysis.

Level of safety (SL) is used in the capacity of an operated variable [25, 29]. The SL value depends on the maximum level of information urgency which is being updated according to recent changes in ICS.

Mechanisms of IPR control are created in the circuit with organizational-technical control governing changing business applications, data array processing plans, infrastructure, and all the corresponding requests to the information safety level. The circuit contains: IDSS in regards to choosing a security strategy and a system of safety level assessment. Managing influence in the circuit is realized by the staff of the IS department. Command information is formed in the process of a goal-oriented choice of an information security method complex rational structure (ISMC).

Operational command information is formed in the CS and IS IO operational management circuit. This information is distributed to the management object by a security manager or automatically with the help of managing influences realization methods.

The safety level of i - IO crosspoint is determined by the following formula:

$$SL_i = 1 - IMI_i = 1 - (C_{ICR} \cdot At_i \cdot As_i \cdot TL_i \cdot SM_i), \quad (1)$$

where IMI_i – IS incident importance in i - IO crosspoint; C_{ICR} – coefficient that allows to represent a result in a range [0; 1]; At_i – IS breach level in i - IO crosspoint; As_i – data asset criticality in i - IO crosspoint; TL_i – trust level of a device, which reports about IS breaches in i - IO crosspoint; SM_i – security measures level in i - IO crosspoint.

Quantity assessment of IO safety can be found the following way:

$$SL_{OI} = \prod_{i=1}^{n} (1 - IMI_i), \quad (2)$$

where $i = \overline{1, n}$ – quantity of crosspoints in ICS (or IO) structure.

Quantity of insider and external attacks against IO are given in the form of tuples:

$$RCA = \langle EUM, EST, CE, SS_{ne}, SS_h, PP, O(NN) \rangle, \quad (3)$$

$$IAA_{l(m)} = \left\langle EUM, IST_l^{k-1}, CE, SS_{ne}, SS_h, PP, O^k\left(NN_m^k\right) \right\rangle, \quad (4)$$

where RCA – remote attack against IO; $IAA_{l(m)}$ – internal attack against IA with k-level of criticality, which are being processed in NN_m crosspoint, when an attacker has a user account with an access permission to a data, of which level of criticality is not higher than $(k - 1)$, and he is trying to increase his level of privileges; EUM – set of entities, comprising: a subset of nodes ICS of transport $-um$ (potential vulnerabilities) [1, 2]; EST – external source of threat; IST_l^{k-1} – insider source of threat; CE – communication equipment in an information channel; SS_{ne}, SS_h – security services against the method of an attack spreading (networked and host; PP – protocols and packets; O – access object; NN_m^k – IO crosspoint, on which information with the highest level of criticality (k) is processed; l, m – numbers of crosspoints.

It is proven in works [8, 14, 16, 24] that the only effective way to identify an attack is in the analysis of a combination of unusual events. That is why in IDSS, an attack spreading WCA possible ways, quantity is compared to a quantity of indicators *IND*. The probability of the fact that suspicious action is a C-A is assessed with the indicators quantity which reacted against the attack spreading method. Crossing $\tau_a(p_i)$ determines an indicators set. We get the following expression:

$$\zeta_a \subseteq PSW \cap IND = \left\{ \left(psw_i, ind_j\right) : psw_i \in PSW \cap ind_j \in IND \right\} \tag{5}$$

where $IND = \left\{ind_j, ind_j\right\}$ – a network or ICS of transport perimeter indicator; *PSW* – possible spreading ways of C-A against IO crosspoints; $\zeta_a(psw_i)$ – crossing that determines an indicators set which reacts against the attack on the recent method.

In conditions when the status of the information environment is unknown, the threat counteraction model is enabled in IDSS which has an opportunity to choose a controlling influence that better corresponds to the management object status. A process of choosing an optimal safety events reaction variant are given in a form of a tuple:

$$\left\langle RV_i, RE_j, DA(RE_j), P_{CA}, P(z_l), OF, RV^{rat}(P_{CA}) \right\rangle \tag{6}$$

where RV_i – a reaction variant; RE_j – a result; DA_j – a damage assessment; z – environment status uncertainty characteristic; $P(z_l) - l$ environment status probability; *OF* – object function of choice; $RV^{rat}(P_{CA})$ – rational variant of reaction; P_{CA} – attack probability.

Safety events [8, 10, 13] reaction variants probability analysis $\{RO_i\}$ has shown that the number of control influences for each situation is limited $i \in [1, 3]$. An alternative advantages evaluation with a damage assessment model is used in IDSS $- \left\{RE_j\right\}, j \in [1, 4]$ taking into account that the IS events reaction variants choice is made in conditions of a potential C-A: no harm, losses for a certain user, losses for a group of users, loss for all ICS from attack realization.

Define a function with which we choose an optimal reaction variant [2, 29, 32]:

$$OF(RV_i, z) = \sum_{l=1}^{s} DA_j\left(RE_j(RV_i, z_l)\right) \cap \left(\prod_{i=1}^{I} p_{ij}\left(RE_j(RV_i), P_{CA}\right) \right), \tag{7}$$

where $i = \overline{1, I}$ – reaction variant; $l = \overline{1, S}$ – functional subsystem of ITS.

The probability p_{ij} of getting every j - result choosing every i - reaction variant is determined the following way:

$$p_{ij} = p_{ij}\left(RE_j(RV_i), P_{CA}\right), \quad \forall i : \sum_j p_{ij} = 1. \tag{8}$$

Control influence rational variant $RV^{rat}(P_{CA})$ is determined this way:

$$RV^{rat}(P_{CA}) = RV\left(\arg \min_i (OF(RV_i, z)) \right). \tag{9}$$

Threat counteraction methods are developed by an ISS analyst taking into account the possible cases of their spreading. They are developed on the basis of IDSS decision making method choice that is adapted to the reaction optimal variant choice: distant invasion through free-to-join networks, local network invasion, through a radio channel using a wireless hot spot or other [8, 12, 13, 16, 17, 19].

During the organizational-technical management process, the stage of planning of storage for information security tools, and the process of gradual removal of uncertainty about the structure and the storage of information security tools in the information security system is being considered. The process of planning *PL* rational sets of means of information protection (MIP) is described with the formula

$$PL = SFS \rightarrow CS_{al}, \tag{10}$$

where *SFS* – the starting data for the synthesis of variants of information security tools sets: $SFS = \{SFS_1, \ldots, SFS_L\}$, $SFS_l = \{DPT_{l1}, \ldots, DPT_{lK_l}\}$ – the range, which consists of IST functional subsystem *l*.

With the help of the system for intelligent support, the process of choosing optimal variant of means of information protection setup for perimeters of information security is considered as the formation of a sub-range for the best variants of setup $CS' \subseteq CS$. The range of the setup variants is described as $CS = \{CS_1, \ldots, CS_{AL}\}$. For the choice of the optimal variant of information security tools the objective function *OF* is used: $CS_{al} = OF(CS)$.

The population of data, which make it possible to compare variants of setups, includes two sub-ranges: $MA_{LS_l} \subset MA_l$, where MA_{LS_l} – indicator of "information security"; MA_l – data for the choice of rational variants [32].

Usage a morphological approach, the model of decision making regarding the choice of the optimal variant of information security tools, is presented in the form of the sequence:

$$RUL : \langle PUR, SFS, RUL_s, CS, MA_l, OF, CS_r(CS') \rangle \tag{11}$$

where *PUR* – the purpose of decision making; RUL_s – a set of rule generation options, which can be presented in an analytical form as the vector product of range $CS = SFS_1 \times \ldots \times SFS_L$.

The choice of the rational variants of information security tools setups is realized on the basis of experts' knowledge in the field of information security. The process of formation of a rational structure of information security tools is divided into five stages.

1. The variants of MIP setup are put under development. The range of possible variants of the solution is set with the help of the morphological matrix. For the considered perimeters of information security, the morphological matrices of information security tools are developed.

2. The intermediary matrices are filled in, which indicate compatibility with the firmware. For each couple of information security tools for different functional subsystems their compatibility is defined. The result is filled in the table. If MIP are consistent, then the function of compatibility is $s(DPT_{lm}, DPT_{pr}) = 1$, in other case $s(DPT_{lm}, DPT_{pr}) = 0$.
3. The range of decisions concerning the choice of variants of setup MIP is generated. The range of tool setups is reduced to the sub-range which is known to be compatible with each other. The range $CS = \{CS_1, \ldots CS_r, \ldots, CS_R\}$, which consists of all possible variants of formulation of setup MIP for the considered perimeter, is the Descartes's production of setups of alternatives (morphological matrix ranges).

The element of the range is presented as follows:

$$CS_r = \left\{(DPT_{1i}, \ldots, DPT_{Ln}) : DPT_{lm} \in SFS_l, \forall l = \overline{1,L}\right\} \tag{12}$$

Let's suppose that it was made a series of N measurements of the controlled variables of the state of CS, in the result it was received the matrix (SQ) of

$$SQ = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1i} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2i} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{l1} & x_{l2} & \ldots & x_{li} & \ldots & x_{ln} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{N1} & x_{N2} & \ldots & x_{Ni} & \ldots & x_{Nn} \end{pmatrix} \tag{13}$$

Here, the vector $X_l = (x_{l1}, x_{l2}, \ldots, x_{\ln})$ corresponds to the results of conducted l-the experiment to study the extent of cyber defence ICS (based on the works [8, 31, 32]). Every meaning x_{li} the input variable x_i we will give a short m number $\left(z_i^{1l}, z_i^{2l}, \ldots, z_i^{jl}, \ldots, z_i^{ml}\right)$, $i = \overline{1,n}$, where z_i^{jl} – the number, that establishes the extent to which the value x_{li} changing x_i in l-th the experiment, for example, when performing penetration tests [19, 32], taken for realization j-th options for cyber defence of ICS, $z_i^{jl} \in [0, 1]$. At the same time the vector X_l we will give a short m number $\left(d_{1l}, d_{2l}, \ldots, d_{jl}, \ldots, d_{ml}\right)$, $l = \overline{1,N}$, where d_j^l – the degree of feasibility using the j-th options for cyber defence in a situation where a set of controlled variables forms a vector X_l, $d_j^l \in [0, 1]$.

For j-th variant for cyber defence of ICS we introduce the matrix (SQ_j)

$$SQ_j = \begin{pmatrix} z_1^{j1} & \ldots & z_i^{j1} & \ldots & z_n^{j1} & z_1^{j1}z_2^{j1} & z_1^{j1}z_3^{j1} & \ldots & z_{n-1}^{j1}z_n^{j1} \\ z_1^{j2} & \ldots & z_i^{j2} & \ldots & z_n^{j2} & z_1^{j2}z_2^{j2} & z_1^{j2}z_3^{j2} & \ldots & z_{n-1}^{j2}z_n^{j2} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ z_1^{jl} & \ldots & z_i^{jl} & \ldots & z_n^{jl} & z_1^{jl}z_2^{jl} & z_1^{jl}z_3^{jl} & \ldots & z_{n-1}^{jl}z_n^{jl} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ z_1^{jN} & \ldots & z_i^{jN} & \ldots & z_n^{jN} & z_1^{jN}z_2^{jN} & z_1^{jN}z_3^{jN} & \ldots & z_{n-1}^{jN}z_n^{jN} \end{pmatrix}, \tag{14}$$

where $D_j^T = (d_j^1 d_j^2 \ldots d_j^l \ldots d_j^N)$, $A_j^T = (a_1^j a_2^j \ldots a_n^j a_{12}^j \ldots a_{i1i2}^j \ldots a_{n-1n}^j)$, $j = \overline{1,m}$.

Next, we introduce the model

$$y_j^l = \sum_{i=1}^{n} a_i^j z_i^{jl} + \sum_{i_1=1}^{n} \sum_{i_2 \neq i_1} a_{i_1 i_2}^j z_{i_1}^{jl} z_{i_2}^{jl}, \tag{15}$$

that specifies the degree of feasibility of using j-th variant of means of cyber protection in l-th situation, for example, in the implementation of the attack of class DDoS or DoS, $j = \overline{1,m}$ [6, 12, 19, 31, 32]

To solve the assessments problem of the appropriateness D_j degree of options for cyber defence for any set of monitored parameters in determining the vectors of parameter estimates of equations [8, 19], the methodology used for compiling and solving the system of fuzzy logical equations [8, 19, 27]. The most natural approach to solving the problem of calculating the components of vectors $z_j = \left(z_1^j, z_2^j, \ldots, z_n^j\right)$ for each set of values of the controlled variables $X_l = (x_{l1}, x_{l2}, \ldots, x_{\ln})$ is the following. For each variable generates a set of membership functions $\psi_j(x_i)$, $j = \overline{1,m}$, $i = \overline{1,n}$, where $\psi_j(x_i)$ – is the membership function of the controlled variable x_i of fuzzy set of values M_{ij} conducive to the realization of j-th version of the defence CS.

The introduction of the aggregate of such membership functions allows to interpret the measured value of each controlled variable x_i as a fuzzy number, whose degree of belonging to each of fuzzy sets $M_{i1}, M_{i2}, \ldots, M_{im}$ is determined by the corresponding values $\psi_j(x_i)$ of membership functions.

Then the calculated numbers $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_j, \ldots, \hat{y}_m$ determine the fuzzy values of the degree of appropriateness of the use of appropriate cyber defence in ICS for a set of measured values of controlled variables of the state information security of IO.

Applying the rules to perform operations on fuzzy numbers, when the membership function of the controlled parameter fuzzy set of values conducive to the realization of the j-th variant is described by the function (L-R)-type [31, 32], obtain the membership functions of fuzzy numbers $\hat{y}_j$, $j = \overline{1,m}$, determining the degree of desirability of selecting a specific solution for logical procedures of detection of C-A (LPDCA) [19, 32]. Where available, modelling will be based on the results of works [8, 19, 26, 32]. The corresponding number for the j-th variant of cyber defence in a particular situation of decision making in the vector of controlled variables $X^* = \left(x_1^*, x_2^*, \ldots, x_n^*\right)$ is equal to the equation [32].

$$\psi_j(X^*) = \begin{cases} L\left(\dfrac{\sum\limits_{i=1}^{n} \hat{a}_i^j x_i^j + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j x_{i_1}^j x_{i_2}^j - \left(\sum\limits_{i=1}^{n} \hat{a}_i^j x_i^* + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j x_{i_1}^* x_{i_2}^*\right)}{\sum\limits_{i=1}^{n} \hat{a}_i^j \alpha_{ij} + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j \left(x_{i_1}^j \alpha_{i_2 j} + x_{i_2}^j \alpha_{i_1 j}\right)}\right), & j = \overline{1,m} \\ R\left(\dfrac{\left(\sum\limits_{i=1}^{n} \hat{a}_i^j x_i^* + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j x_{i_1}^* x_{i_2}^*\right) - \left(\sum\limits_{i=1}^{n} \hat{a}_i^j x_i^j + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j x_{i_1}^j x_{i_2}^j\right)}{\sum\limits_{i=1}^{n} \hat{a}_i^j \beta_{ij} + \sum\limits_{i_1=1}^{n} \sum\limits_{i_2 \neq i_1} \hat{a}_{i_1 i_2}^j \left(x_{i_1}^j \beta_{i_2 j} + x_{i_2}^j \beta_{i_1 j}\right)}\right), & \alpha > 0, \beta > 0. \end{cases} \tag{16}$$

Types of decisions in the relevant policy CS of IO are selected as follows (based on the works [12, 13, 17, 31, 32]): protection for CCS is not needed (d1); protection of CCS is not required to update the system software CCS (d2); need update anti-virus protection (d3); need to update the technical means of information protection (TMIP) for CCS (d4); requires installation of firewall (FW) (d5); need to establish C-A detection system (d6); need update modules CCS (d7); necessary organizational measures to distribute access to CCS (d8); requires installation of means of protection against information leakage through other sources (d9). The factors influencing the choice of solutions on the cyber defence ICS represented as linguistic variables (Table 1) for which the selected set and universal terms. Then the need to use a specific strategy for the protection of IO may be described by the following dependencies:

Table 1. Factors influencing the choice of strategy of cyber defence of CCS (or ICS)

Partial setting of the state	Universe
x_1 – the level of secrecy of information resources	[0, 1]
x_2 – mode access to hardware CCS	[0, 1]
x_3 – level of protection against the unauthorized (UNA) coping	[0, 1]
x_4 – cases UNA data destruction on the CCS	[0, 1]
x_5 – cases of incorrect work of the software CCS	[0, 1]
x_6 – access control	[0, 1]
x_7 – condition novelty of system software	[0, 100], %
x_8 – availability of cryptographic tools in CCS	[0, 100], %
x_9 – number of incidents by information security for the CCS	[0, 1]
x_{10} – qualified staff in CCS	[0, 1]
x_{11} – intervention in the work of the CCS from the outside	[0, 100], %
x_{12} – availability of funds reservation of important information in the CCS	[0, 100], %
x_{13} – information loss due to the failure of ICS	[0, 1]
x_{14} – availability of systems to combat cyber attacks	[0, 1]
x_{15} – type of antivirus software	[0, 1]
x_{16} – availability of audit procedures for CCS	[0, 1]
x_{17} – the availability of means of identification and authentication of users of CCS	[0, 1]
x_{18} – active TMIP in CCS	[0, 1]
x_{19} – passive TMIP in CCS	[0, 1]

$$D = f_D(x_{19},\ y_4,\ y_5), y_1 = f_1(x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ y_3),$$
$$y_2 = f_2(x_9,\ x_{10},\ x_{11},\ x_{12}), y_3 = f_3(x_8,\ x_{13},\ x_{14}),$$
$$y_4 = f_4(x_{15},\ x_{16},\ x_{17},\ x_{18}), y_5 = f_5(x_1,\ x_2,\ y_1,\ y_2),$$

where D is the state of cyber defence of IO, $y_1,\ y_2, y_3, y_4, y_5$ – intermediate, General variables: y_1 – the state of cyber defence of IO {below the critical (bc), critical (cr), above the critical (ac), high (h)}; y_2 – the influence of external factors {adverse (a)

blank;moderate (m), enabling (e)}; y_3 – the level of security of IO {below the critical (ba), critical (c), above the critical (ac), high (h)}; y_4 – personnel qualifications {low (l), below average (ba) average (a) above average (aa), high (h)}; y_5 – the need to improve cyber defence ICS {doesn't need it, update system software (uss), virus protection (vp), installation of ICS (isw)} [26, 31, 32].

The software package "Decision Support System of Management protection of information – DMSSCIS" [8, 19, 26, 32], designed for reasoned choice of a rational set of information security in the process of designing information security systems of information objects. DMSSCIS was also used in the modernization of existing information security systems in data centers of transport companies in Salt (2015–2016, Jordan), Kyiv (2016–2017, Ukraine).

5 Results

On the software DMSSCIS [29], that particular selection method implemented an efficient option for responding to security events. The results are shown in Table 2.

Figure 1 shows examples of results of simulating the rational sets of information protection means (IPM) received using DMSSCIS. As can be seen from the graphs in Fig. 1(a), an increase in appropriations of information protection means (IPM) [29, 32] around the perimeter of information and communication systems of transport enterprises (ICSTE), for example, a large railway junction or motor transport enterprise, leads to a reduction in all probabilistic indicators of the possibility to successfully implement the attack.

In Fig. 1(a) the graphs for the perimeters I, II, IV are shown. For example, with the total costs for an ICSTE node at the IPM perimeter IV (PNE (IV) – the perimeter of the network equipment [1, 2, 31, 32]), is about 5000 to 5300 conventional monetary units (cvu) the probability of the offender achieving all of his goals is $P_a = 0.00082$. The curve in Fig. 1(a) has a pronounced exponential character with a negative coefficient.

As can be seen from the graphs, the increase in costs for IPM of perimeters above a certain level (for example, for the perimeter PNE (IV) > 12900 cvu) is not appropriate. This is due to the fact that after the subsequent increase in the amount of appropriations for the IPM perimeter, the probability of the offender reaching all his goals is practically not reduced.

Figure 1(b) shows the indicator of the total cost of ICS related to the actions of the attacking party, as well as the allocations for the MIP, on the likelihood of successful implementation of IPR activities. The graph has a pronounced minimum. Thus, starting from the extremum point, the cost level $C_{\sum}$ exceeds the loss of losses from C-A to ICSTE.

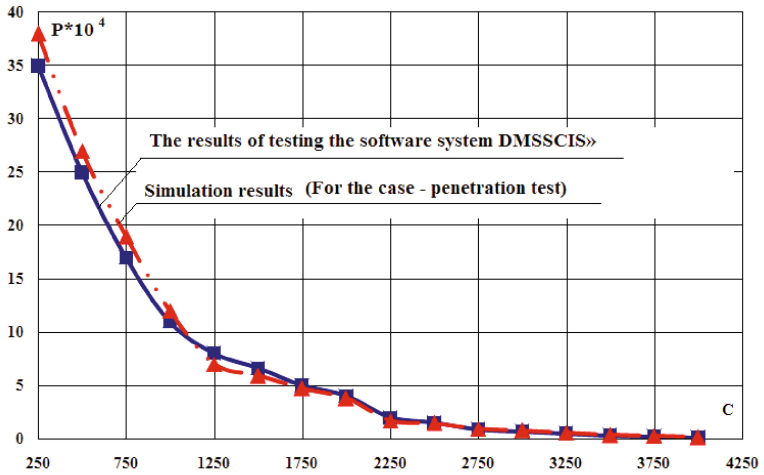
During the research the possibility was taken into account of an attack that implements remote intrusion through the perimeter, the availability of internal and external users, and abusers that have high privileges and violate the safety of information. After the formation of efficient information security in enterprises which took part in the study, with the help of intelligent decision support "DMSSCIS" the predicted value was 1,78–1,91% risk that there was an average value of 5,9–6,2 times less risk to information security systems compared to before.

Table 2. The results of testing the software system DMSSCIS

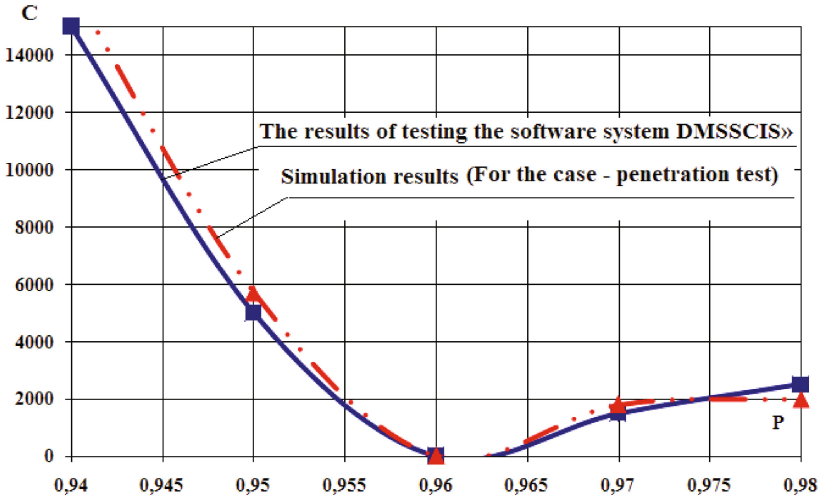
Class of cyber-attacks (C-A)	Options for responding to the current settings information of environment of information object		
	There are some linguistic variables added to IDSS: A – «number of unusual events in network against the spreading of attack», B – «number of unusual events in the host», C – «number of unusual events in IO perimeter», P_a – «probability of the fact that found unusual activity is an attack»		
U2R	$A = 2$, $B = 3$, $P_a = 0{,}5,\ldots,0{,}6$	$A = 1$, $B = 1$, $P_a = 0{,}25,\ldots,0{,}3$	
	The end of session attack source node	Sending a warning message to the user	
R2L	$A = 1$, $B = 3$, $P_a = 0{,}4,\ldots,0{,}65$	$A = 1$, $B = 1$, $P_a = 0{,}2,\ldots,0{,}4$	
	The end of session attack source node	Sending a warning message to the user	
DOS/DDOS	$A = 2$, $B = 3$, $C = 2$, $P_a = 0{,}5,\ldots,0{,}7$	$A = 1$, $B = 1$, $C = 2$, $P_a = 0.4,\ldots,0.65$	
	The end of session attack source node	Sending a warning message to the user	
The external attack via Wi Fi	$A = 3$, $B = 2$, $C = 3$, $P_a = 0{,}6\ldots0{,}7$	*$A = 1$, $B = 1$, $C = 2$, $P_a = 0{,}4\ldots0{,}6$*	*$A = 1$, $B = 1$, $C = 1$, $P_a = 0{,}3\ldots0{,}5$*
	Blocking access point	DOS-attack on stations	The lack of response
A remote attack via lines on the perimeter	$A = 3$, $B = 4$, $C = 2$, $P_a = 0{,}6\ldots0{,}8$	$A = 1$, $B = 1$, $C = 1$, $P_a = 0{,}2\ldots0{,}4$	$A = 1$, $B = 2$, $C = 1$, $P_a = 0{,}07\ldots0{,}25$
	Blocking access to the server in the network	Reconfiguration of security services to block IP	Sending a warning message to the user
The cost of optimal variants of the IS sets using an object function – $\psi_j(x_i)$			
Example. The perimeter of the network equipment 5000–5300 cvu. (For a large information node of a transport enterprise)			

The proposed approach to constructing a comprehensive information security management system for IO allowed us to reduce expenditures for IPM by 32–35% compared to alternative methods [6, 10, 14, 30, 31].

A certain shortcoming of the "DMSSCIS" is the requirement to engage at the initial stage of examination a few independent experts for the construction of membership functions and compiling production rules. At the present stage of research, for this purpose we employed tools from the Fuzzy Toolbox (Matlab), which computes such indicators of MIP as "protection of information" for each involved perimeter of protection.



a) The dependence of probability (P) of the realization of all the goals of C-A against the ICSTE from the cost (C) of the hardware-software set (HSS) IPM on the perimeter of a separate node



b) The dependence of the total costs (C) on the HSS IPM for the ICSTE node on the system probability (P)

Fig. 1. The results using DMSSCIS of the rational sets of IPM for IO

Further development of the work may lay in the improvement of the interaction of traditional cybersecurity mechanisms used in transport and the proposed decision support system – DMSSCIS.

In general, based on the conducted numerical experiments in the MatLab environment, and the results obtained for the actual operating conditions of "DMSSCIS", we can ascertain the effectiveness of the proposed model.

6 Conclusion

1. We improved a model for the operational management of ICS CS and the formation of a balanced complex of means of protection. The model is based on the morphological approach. In contrast to the existing solutions, the model with regard to the morphological matrices for each of the perimeters of protection of ICS prepared by DMSSCIS allows us to generate variants of sets of means of protection, which take into account the compatibility of software and hardware tools.
2. A developed software suite was made for intelligent decision support circuits organizational, technical and operational management of information system protection facilities. It confirmed the adequacy of the proposed models and algorithms. By using the developed system of intelligent decision support, networks of enterprises using DMSSCIS reduced the projected cost of the planned system of protection to 35%.

References

1. Al Hadidi, M., Ibrahim, Y., Lakhno, V., Korchenko, A., Tereshchuk, A., Pereverzev, A.: Intelligent systems for monitoring and recognition of cyber attacks on information and communication systems of transport. Int. Rev. Comput. Softw. **11**(12), 1167–1177 (2016)
2. Lakhno, V., Grabarev, A.: Improving the transport cyber security under destructive impacts on information and communication systems. Eastern-Eur. J. Enterp. Technol. **1/3**(79), 4–11 (2016). doi:10.15587/1729-4061.2016.60711
3. Dunn, W.: Practical Design of Safety-Critical Systems. Reliability Press, Cambridge (2002). ISBN 0971752702
4. El Hassani, A.A., El Kalam, A.A., Bouhoula, A., Abassi, R., Ouahman, A.A.: Integrity-OrBAC: a new model to preserve critical infrastructures integrity. Int. J. Inf. Secur. **14**(4), 367–385 (2014). doi:10.1007/s10207-014-0254-9
5. Zhang, Y., Wang, L., Sun, W., Green, R.C., Alam, M.: Distributed intrusion detection system in a multi-layer network architecture of smart grids. IEEE Trans. Smart Grid **2**(4), 796–808 (2011). doi:10.1109/tsg.2011.2159818
6. Al-Jarrah, O., Arafat, A.: Network intrusion detection system using attack behavior classification. In: 2014 5th International Conference on Information and Communication Systems (ICICS), pp. 1–6 (2014). doi:10.1109/iacs.2014.6841978
7. Louvieris, P., Clewley, N., Liu, X.: Effects-based feature identification for network intrusion detection. Neurocomputing **121**(9), 265–273 (2013). doi:10.1016/j.neucom.2013.04.038
8. Lakhno, V.: Creation of the adaptive cyber threat detection system on the basis of fuzzy feature clustering. Eastern-Eur. J. Enterp. Technol. **2**(9(80)), 18–25 (2016). doi:10.15587/1729-4061.2016.66015

9. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. Int. J. Very Large Data Bases **16**(4), 507–521 (2007). doi:10.1007/s00778-006-0002-5
10. Cavusoglu, H., Srinivasan, R., Wei, T.Y.: Decision-theoretic and game-theoretic approaches to IT security investment. J. Manag. Inf. Syst. (ACySe) **25**(2), 281–304 (2008)
11. Chang, L.-Y., Lee, Z.-J.: Applying fuzzy expert system to information security risk assessment: a case study on an attendance system. In: 2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY), pp. 346–351 (2013). doi:10.1109/iFuzzy.2013.6825462
12. Atymtayeva, L., Kozhakhmet, K., Bortsov, G.: Building a knowledge base for expert system in information security. In: Soft Computing in Artificial Intelligence. Advances in Intelligent Systems and Computing, vol. 270, pp. 57–76 (2014). doi:10.1007/978-3-319-05515-2_7
13. Kanatov, M., Atymtayeva, L., Yagaliyeva, B.: Expert systems for information security management and audit. Implementation phase issues. In: Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS), 3–6 December 2014, pp. 896–900 (2014). doi:10.1109/SCIS-ISIS.2014.7044702
14. Yang, Y.P.O., Shieh, H.M., Tzeng, G.H.: A VIKOR technique based on DEMATEL and ANP for information security risk control assessment. Inf. Sci. **232**, 482–500 (2013). http://dx.doi.org/10.1016/j.ins.2011.09.012S
15. Pawar, N.: Intrusion detection in computer network using genetic algorithm approach: a survey. Int. J. Adv. Eng. Technol. **6**(2), 730–736 (2013)
16. Linda, O., Manic, M., Vollmer, T., Wright, J.: Fuzzy logic based anomaly detection for embedded network security cyber sensor. In: IEEE Symposium on Computational Intelligence in Cyber Security (CICS), 11–15 April 2011, pp. 202–209 (2011). doi:10.1109/cicybs.2011.5949392
17. Mashkina, I.V., Guzairov, M.B., Vasilyev, V.I., Tuliganova, L.R., Konovalov, A.S.: Issues of information security control in virtualization segment of company information system. In: XIX IEEE International Conference on Soft Computing and Measurements, pp. 161–163 (2016). doi:10.1109/SCM.2016.7519715
18. Oglaza, A., Laborde, R., Zarate, P.: Authorization policies: using decision support system for context-aware protection of user's private data. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 16–18 July 2013 (2013). doi:10.1109/TrustCom.2013.202
19. Lakhno, V., Kazmirchuk, S., Kovalenko, Y., Myrutenko, L., Zhmurko, T.: Design of adaptive system of detection of cyber-attacks, based on the model of logical procedures and the coverage matrices of features. Eastern-Eur. J. Enterp. Technol. **3/9**(81), 30–38 (2016). doi:10.15587/1729-4061.2016.71769
20. Gamal, M.M., Hasan, B., Hegazy, A.F.: A security analysis framework powered by an expert system. Int. J. Comput. Sci. Secur. (IJCSS) **4**(6), 505–527 (2011)
21. Goztepe, K.: Designing fuzzy rule based expert system for cyber security. Int. J. Inf. Secur. Sci. **1**(1), 13–19 (2012)
22. Gutzwiller, S.R., Hunt, S.M., Lange, D.S.: A task analysis toward characterizing cyber-cognitive situation awareness (CCSA) in cyber defense analysts. In: IEEE International Multi-disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 21–25 March 2016. doi:10.1109/COGSIMA.2016.7497780
23. Reesa, L.P., Deanea, J.K., Rakesa, T.R., Bakerb, W.H.: Decision support for cybersecurity risk planning. Decis. Support Syst. **51**(3), 493–505 (2011). doi:10.1016/j.dss.2011.02.013
24. Paliwal, S., Gupta, R.: Denial-of-Service, probing & remote to user (R2L) attack detection using genetic algorithm. Int. J. Comput. Appl. **60**(19), 57–62 (2012)

25. Ericsson, N.G.: Cyber security and power system communication-essential parts of a smart grid infrastructure. IEEE Trans. Power Deliv. **25**(3), 1501–1507 (2010). doi:10.1109/tpwrd.2010.2046654
26. Storchak, A.: Model assessment of information security based on multi-step process driven decision. Spec. Telecommun. Syst. Inf. Protect. **2**(24), 112–117 (2013)
27. Atymtayeva, L., Kozhakhmet, K., Bortsova, G.: Building a knowledge base for expert system in information security. In: Soft Computing in Artificial Intelligence. Advances in Intelligent Systems and Computing, vol. 270, pp. 57–76 (2014). doi:10.1007/978-3-319-05515-2_7
28. Valenzuela, J., Wang, J., Bissinger, N.: Real-time intrusion detection in power system operations. IEEE Trans. Power Syst. **28**(2), 1052–1062 (2013). doi:10.1109/TPWRS.2012.2224144
29. Lakhno, V.A., Kravchuk, P.U., Mekhed, D.B., Mohylnyi, H.A., Donchenko, V.U.: Development of a support system for managing the cyber protection of an information object. J. Theor. Appl. Inf. Technol. **95**(6), 1263–1272 (2017)
30. Kritikos, K., Massonet, P.: Security-based adaptation of multi-cloud applications. In: Data Privacy Management, and Security Assurance. Lecture Notes in Computer Science, vol. 9481, pp. 47–64 (2016). doi:10.1007/978-3-319-29883-2_4
31. Burachok, V.: Algorithm for evaluating the degree of protection of special information and telecommunication systems. Inf. Secur. **3**, 19–27 (2010)
32. Petrov, O., Borowik, B., Karpinskyy, M., Korchenko, O., Lakhno, V.: Immune and Defensive Corporate Systems with Intellectual Identification of Threats. Śląska Oficyna Drukarska, Pszczyna (2016)

Brainstorming Sessions – Towards Improving Effectiveness and Assessment of Ideas Generation

Adrian Andrzejewski(✉), Kordian Kręcisz, Mariusz Matusiak, Andrzej Romanowski, and Laurent Babout

Institute of Applied Computer Science, Lodz University of Technology, ul. Stefanowskiego 18/22, 90-924 Lodz, Poland
pramuspl@gmail.com, korkrec@live.com, azamennace@gmail.com, androm@kis.p.lodz.pl, lbabout@iis.p.lodz.pl

Abstract. The project was intended to develop a way of improving outcomes of brainstorming sessions and acquiring participants involvement during those appointments. We analyzed existing team work schemes and most commonly faced issues. Having examined hallmarks of available solutions we proposed a prototype of a desktop program paired with a mobile application that automatizes a flow of a session by significantly reducing amount of required on-the-fly noting during the ideation process thanks to an electronic record system that collects both ideas and a user activity. This helps keeping team members not distracted from the actual brainstorming process. Proposed tool enhances both performance and experience as well as enables deeper ex-post analysis of a whole brainstorming session flow. Several groups of students have participated in a user study on the influence of the tool. Favorable reports and feedback have been received stating that the prototype has a positive impact both on users work and the sessions outcome.

Keywords: Teamwork · Assessment · Brainstorming session · Design observatory · Electronic brainstorming · Nominal brainstorming

1 Introduction

The possibilities of enhancing interpersonal relations during brainstorming sessions have been widely researched from both psychological and practical perspective. Design Thinking Observatories carefully monitor the behavior of team members, registering their actions and events to analyze, whether the discussion is heading in the right direction [2, 3]. The observatories are handled by trained people and utilized a set of cameras, microphones and some sort of a registering application. For the brainstorming sessions one of the challenges ahead, apart from the direct relations among the team members, is to create an insight for the team members to synthesize the ideas, and organize the data in use during the discussion. The data should be legible to create a shared understanding among all team members, because people might have diverse individual perspectives that guide their interpretations [1]. In our work, we wanted to

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_12

make a combination of both approaches. Although, we did not monitor the exact behaviors of the members via cameras and microphones, but rather their activity through the number of ideas given during the brainstorming session. Our goal was to develop a way of assessing and organizing the discussion data in a clean way.

The following article is divided into several sections. First section takes a look at related work in the topic of electronic brainstorming solutions. Then the proposed system design is introduced. Before the final section where results of performed tests are formulated, a short section concerns also the technology used during development process.

2 Related Work

There has been much research conducted in the field of electronic brainstorming. The experiments considered several types of brainstorming sessions with varying number of people in the teams and other factors influencing the sessions. In these studies the people were usually introduced to a kind of EBS (Electronic Brainstorming System), which helped them register the ideas into a common database. The EBS were usually applications installed on the computers. As stated in [4], when exposed to the EBS, the subjects of the research concentrated mainly on the computer instead of face-to-face communication. "Approximately 96.6% of the planned time was spent working on the computer. In contrast time spent in group interaction accounted for only 3.4% of the session time". Thus, the use of electronic brainstorming systems in the discussion resulted in peculiar case of nominal brainstorming. Nominal brainstorming is a type of a session in which each and every person has the time for their own evaluation of the problem and individual gathering of the ideas. The downhill of this method is usually the lack of dynamic idea exchange between the people and therefore lack of mutual inspiration. On the other hand, participants could stay focused on the task at hand and are less prone to distraction. Both types of brainstorming, whether nominal or face-to-face have their pros and cons. In the other research from R. Brent Gallupe [7] and Hamit Coskun [8] more variables were added to the idea of electronic brainstorming. Firstly, the experiments were conducted several times with varying group sizes to check the influence of the application usage on the effectiveness of the session. Moreover in the second research the subjects were told in one of the experiments to memorize all of the ideas that were given.

The results showed that with small groups the technology did not enhance the productivity of the members. However the situation looks differently for larger teams. "In the electronic groups, performance increased substantially with group size. In contrast, performance in the non-electronic brainstorming groups did not increase as group size increased" [8]. "The findings of all three experiments have consistently demonstrated that unique and original ideas increased as the group size increased" [7]. Moreover in the latter article, the researchers told the subjects to memorize all the ideas during the session. It resulted in an interesting outcome that for short-timed sessions it turned out to be an obstacle. However for longer discussions the productivity increased, as the subjects have a better outlook on the possible solutions.

Some work also highlights the importance of visual materials during the discussion and ways of improving people's attention during the discussions. "Two factors in

particular seem to contribute to losses from inefficient attention processes: 1. attention diversion because of excessive exposure to other people's ideas, and 2. lack of attention on other people's ideas" [5]. It is also proven that team members can concentrate more effectively if they can easily view the ideas of other people, inspire themselves and deliver new ideas based on the already existing ones. "The productivity gain was mainly associated with higher elaboration on others' ideas rather than with an increase in the number of unique ideas" [5]. Other study took the above-mentioned point into consideration, while developing a sticky notes interface for teams [9]. The study discovered that there are three main factors contributing to the reduction of effectiveness of brainstorming sessions:

1. Fear of negative evaluation of the idea, resulting in people withholding their ideas to themselves.
2. The members not being fast enough to write all the ideas during a discussion, which in some cases made these ideas no longer relevant or original.
3. Shallow elaboration on the ideas and lack of sources, which as a result leads to difficulty in understanding and accepting the ideas by other members [9].

One of the solutions introduced by the study involved the idea of private sticky note sections. The sections were utilized by the team members in order to store their temporary ideas. They were not exposed to public knowledge, which enabled the users to develop the ideas with no pressure. Thus, in our project we have high regard for anonymity of users and ease of use of the resulting tool.

To conclude, what we aim for is finding a way of fusing nominal and face-to-face brainstorming, so that people can discuss the matters and at the same time register their own ideas. Nowadays it is not a common practice to have every team member work with their laptop during a brainstorming session and not everyone can carry it around all the time. We decided that our solution has to be more mobile to fit in current needs. The EBS developed in this project cannot interfere in the course of discussion, so that we have as much face-to-face interaction as possible. Our goal was also to display the ideas in a legible way, provide an easy mean of idea evaluation and allow to extend the already existing ideas at any point.

3 System Design and Principles of Operation

Main assumption was to provide an easy to use and independent solution which would not require involvement of anyone not particularly interested in designing process and which would not distract from the process itself. Eventually it should also be possible to generate a plain and approachable report for the supervisor for evaluation purposes. Video and sound recordings as well as real-time analysis from the outside were all abandoned as a result of conclusion from previous research by Stanford University team [2, 3]. Moreover, the recording sessions have been criticized to have a negative impact on participants' performance – they reported that they feel embarrassed about expressing their ideas due to constant pressure and invigilation of the system. We came up with a solution, which meets the requirements listed above but still respects conclusions from cited research. In order to deliver self-assessment utility we decided to

provide team members with a mobile application, which is simple enough not to disturb the natural flow of conversation. The entire user activity during a session should be synchronized and gathered to create visualization for possible further analysis. To take advantage of the existing solutions we decided to take the presentation of ideas using sticky notes to another level. Thus, the application prototype consists of two main components:

- Desktop application – run on a Personal Computer serves as host for client mobile applications used by participants of the meeting. Meeting leader notes incoming ideas which are then processed and used for visualization and report generation as well as post-session analysis. It also serves as an interface to monitor individual users' activities.
- Mobile application – client application used by each meeting participant for submission of new ideas, expanding the existing ones and expressing attitude towards each of them.

The basic idea behind the tool is the following: the meeting starts with users logging with their mobile devices into the system set up by the desktop application. The desktop application is being operated by an administrator of the meeting, usually a scrum master or a secretary. Once they are logged in, the list of users is displayed and the meeting can commence. The mobile application offers a simple interface, enabling the user to express a new idea simply by pressing a button.

Firstly, the user presses the "New idea" button, which sends a notification to the host application to open an interface for idea registration. Secondly, the administrator types in the author's idea. While new ideas flow in, the list of submissions is dynamically refreshed and displayed to all participants. Once the submission stage is finished, the interface alters a bit. The initial evaluation phase starts in which users can assess the ideas. The amount of ideas, votes and user activity over time especially are being registered in the system constantly (see Fig. 1).

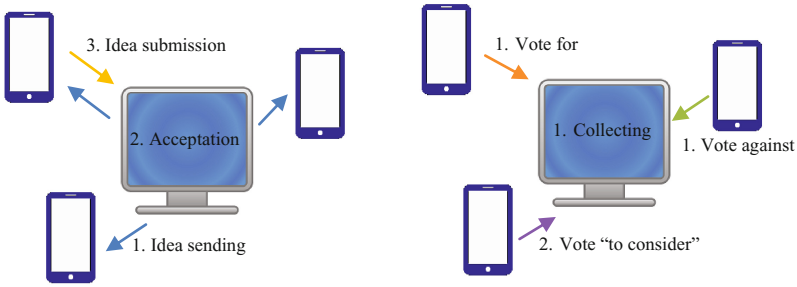


Fig. 1. Diagrams for basic mechanism of submitting an idea during a brainstorming session (left), and for mechanism of voting on ideas after the session (right).

Collected data is used for generation of a virtual whiteboard filled with sticky notes. It is a visual and fully interactive presentation of a brainstorming session outcome – the notes can be freely dragged, grouped, prioritized and customized. They contain not

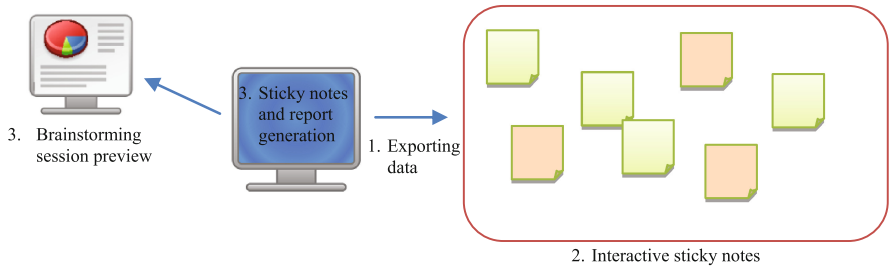


Fig. 2. Interactive presentation of ideas and user activity report generation

only ideas descriptions but also their popularity and a reference to a base idea if present (see Fig. 2).

Gathered information would also be used for generation of another report for the team supervisor – unlike the previous one, this would include the performance of each participant and introduce the timeline presenting the whole course of the discussion.

The list below summarizes main features of the system:

- Mobile application is intended to be easy to use and non-distracting.
- Desktop application serves as a meeting journal/notepad and collects data for further processing.
- Collected and voted ideas are presented in a form of an interactive board.
- This data might also be used to generate a report incorporating time factor which could be used to performance measuring.
- Portability of the system is assured due to usage of the latest technologies available on the market. Desktop application can be run on all popular operating systems like Windows, Linux and OS X. Mobile application is prepared for Android, Windows Phone and iOS.

4 Technology

In accordance to the proposed requirements our solution had to be developed using technologies allowing it to be easily available across all popular devices and systems. Several possible approaches included native platform development using three programming languages simultaneously (Java, C#, Objective-C), low-level programming in C/C++ or writing a web application partially in HTML and JavaScript. Most of these technologies could come in handy in particular situations like game design (native code) or displaying static web content (HTTP requests). The latter scenario has become very popular these days and is widely supported by lots of new frameworks like Rails, Node.js, AngularJS, Apache Cordova etc. Our approach focused however on creating real-time dynamic application allowing the highest possible efficiency to be maintained during development process. Software portability and an offered set of features making us capable of designing fully functional system led us to choose Java 8

combined with the NetBeans Platform as the build tools for the desktop part of our application and Xamarin.Forms with C# for the mobile part.

As it can be seen on the Fig. 3, the desktop part consists of 4 major modules. Java application is the heart of the entire system. Its role is to provide an interface for the session supervisor for setting up brainstorming meetings, introducing and updating users data into database, generating web reports for the team and giving detailed information about users activity in customizable and intuitive form. For data acquisition we stood in front of a wide variety of relational database systems to choose from. In terms of portability our aim was to provide users with an application that would not require any extra components to be installed. That is why Apache Derby engine has been embedded into our system. The database stores all necessary information including users' activity and outcomes of all performed sessions. To bind all these components together with the mobile application a local server has been set up also on the desktop. Data transmission is handled through two types of sockets. UDP socket broadcasts information about host PC (IP address) in a local network. That data is then received by mobile clients and used to establish end-to-end dedicated connections between devices through TCP socket. When a session is over, the supervisor is able to generate a report with interactive sticky notes that can be displayed on large surfaces like a screen or Surface Hub. That part has been designed using leading front-end web technology AngularJS. The most important part of the project was to deliver a mobile application with highly intuitive and simple interface. That is why it is presented on just two main views – one for authentication process and the other for brainstorming activities. When pairing with host is successful a connected user is allowed to submit new ideas as well as, according to brainstorming rules, extend each of them (the so called "Yes, and…" behavior). List of all proposed ideas is displayed instantaneously at the top of the screen. When the voting phase is reached new buttons appear and user can share his opinion about every proposal (see Fig. 4). Given the fact that mobile smart devices are getting more and more popular nowadays and almost every functionality of additional electronic equipment is now replaced with a mobile application we believe that using participants' own personal devices is much more comfortable than introducing lots of new, single-use hardware interfaces, especially in the studied topic.

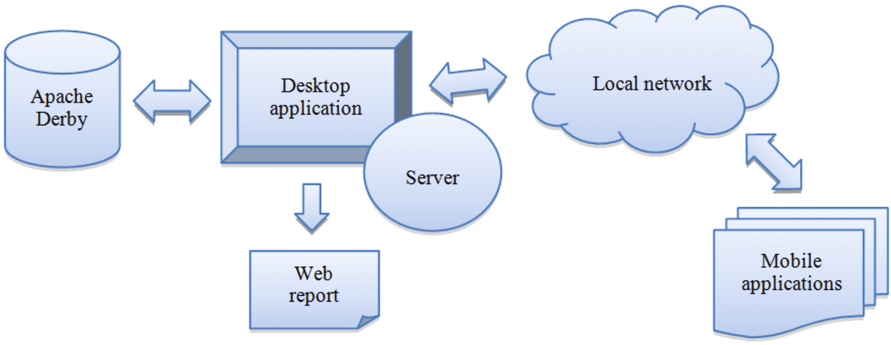


Fig. 3. Block scheme of the application

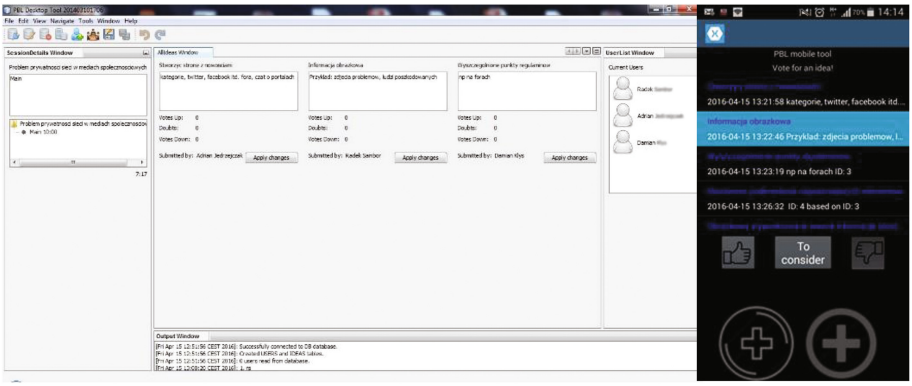


Fig. 4. Screens with the main views of the desktop application (left) and the mobile application (right)

5 Testing and Results Discussion

In contrast to many related studies [1, 9] our solution does not change the way how brainstorming sessions flow e.g. by dropping user-to-user verbal interactions completely. Instead it enhances them by increasing participants activity and bringing tools for later investigation and finding events that might have not been caught during meetings. In early stage of the project we presented the mobile application prototype to fellow students asking for their feedback towards the interface. It resulted in several improvements including separation of session into stages; expanding voting possibilities; 'look&feel' improvements as well as in providing the features that would not likely become available in traditional brainstorming sessions such as providing participants with multimedia materials related to a session's topic instantly.

Once we developed a fully functional prototype in order to measure the effectiveness of proposed solution we carried out several iterations of field studies in form of a comparative analysis – we wanted to measure how our tools stand confronted with a conventional approach. In order to do that we inquired several groups of students (members of Team Project; a Problem Based Learning method-based projects run at the International Faculty of Engineering, Lodz University of Technology, Poland) to participate in two subsequent brainstorming sessions. At the beginning of each session, the participants were given the discussion topic. The difference of the half of experimental group lied in method of noting the ideas – the first was a traditional "sticky notes and pens" session. The second group of participants were provided with smartphones having the proposed application installed. At the same time, one of the research group members took the role of a meeting secretary working in the desktop mode. This is illustrated on Fig. 5.

At the end of first phase we asked the groups about how they had felt about their performance as well as how technology might had been be used to boost it. After second part we repeated the question about performance and requested feedback on

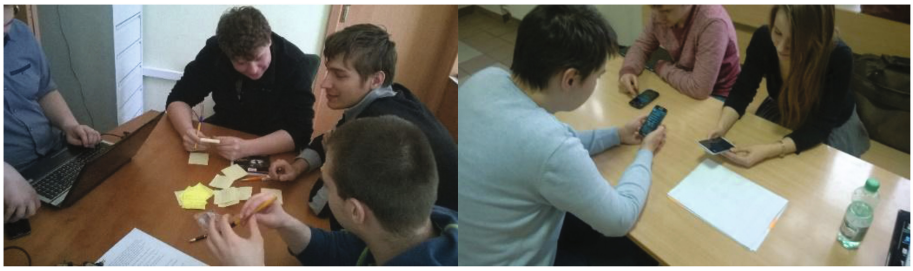


Fig. 5. 1st phase of testing – pen and paper session (left) and the 2nd phase of testing – session with the tool (right).

Table 1. User satisfaction.

Questions	Rating (average, scale 0–10)			
	Group 1	Group 2	Group 3	Group 4
1st session – pen and paper				
How would you rate your performance*?*	5.00	7.00	6.33	6.33
2nd session – E-brainstorming tool				
How would you rate your performance*?*	8.00	5.33	7.00	7.33
How helpful was the tool?	8.33	7.33	7.33	7.33

how useful our tool was compared to the traditional sticky notes. For this purpose we used a range of values starting from 0 to 10. The results are listed in Table 1.

Most of the groups stated that their performance was somewhat better when using the mobile application. Only one group stated otherwise; yet at the same time they pointed out that they found the main topic of the discussion difficult to discuss. For this reason we put more focus on the answers how the tool helped them to communicate and to express their ideas.

All the participants were generally satisfied with the proposed application and they reported improved experience in their performance. Apart from statistical data we asked few open questions about general feeling about proposed solution. Common opinion stated that the presence of the secretary is invaluable for assuring better order of the session course and for providing participants with more time to focus on developing ideas rather than writing them down. The mobile application was praised for eliminating the problem of occurrence of repetitive ideas in fast-paced sessions as well as 'biased voting' – making it anonymous prevents voting on personal preference rather than by the value of idea itself.

The participants also stated that the straightforward interface and its different look in consecutive phases makes it easy to understand the rules of brainstorming concept for people unfamiliar with it. It keeps the ideas tidy and clean and the system allows exporting and sharing the results of the session. The virtual sticky notes board (see Fig. 6) was also well received. The participants valued its aesthetics – in case of numerous real-life notes, the whole board becomes unreadable and very often hard to

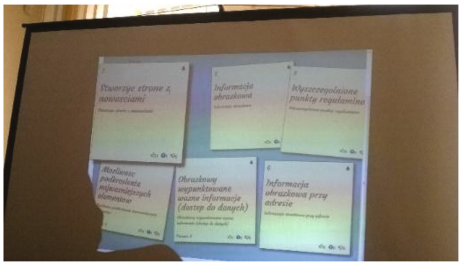


Fig. 6. Brainstorming session review with the use of interactive sticky notes.

manipulate while the virtual counterpart makes it simpler and well-organized. We observed that using the application stimulates activity of previously passive team members. People, whose productivity was clearly much lower in the regular brainstorming session, became more creative and expressive in the second phase. This might be caused by collecting data about each user's input.

We also received some hints about further improvement of the system including adapting it for the design teams located in distant places, more detailed ideas view in the mobile application, introduction of folders in the interactive board and some other minor interface changes.

6 Conclusions and Future Work

This paper presents a successful approach enhancing participants' performance during brainstorming sessions with the aid of a computer-aided ideas recording system. Instead of noting down the ideas onto the post-it notes, users can focus on the ideas generation with no distraction associated with writing; since the system managed by a third person takes care of session's results maintenance. Recorded ideas can then be used to build ideas further on, or to vote and judge on the ideas as well as to display them eventually as virtual post-it-based ideation results. Proposed tool being a tandem of desktop and mobile applications has been successfully implemented and tested by several groups of students. Comparative analysis proved that it had positive influence on their performance what has been proved both in number of generated ideas and their subjective opinion. What is more, the mobile application has been well received and the participants valued its potential. This led us to decision of further development of this tool. The most interesting fields of further development is the multitouch large surface implementation such as presented here [10], as well as to consider crowdsourcing as a possible way of interaction within the ideation meetings such as for scientific, tomography purposes [11, 12] provided careful interface design is assured as pointed out in [13].

Acknowledgements. We are much obliged to Przemysław Kucharski and Piotr Duch from the Institute of Applied Computer Science for their help during the last phase of our project. We also

would like to thank all the students that took part in the tests and provided us with valuable comments and suggestions.

References

1. Gumienny, R., Dow, S., Meinel, C.: Supporting the synthesis of information in design teams. In: Proceedings of the 2014 Conference on Designing Interactive Systems, pp. 463–472. ACM, Vancouver (2014)
2. Sonalkar, N., Mabogunje, A., Leifer, L.: Analyzing the display of professional knowledge through interpersonal interactions in design reviews. In: DTRS 10: Design Thinking Research Symposium 2014, Purdue University (2014)
3. Torlind, P., Sonalkar, N., Bergstrom, M., Blanco, E., Hicks, B., McAlpine, H.: Lessons learned and future challenges for design observatory research. In: International Conference on Engineering Design, ICED 2009 (2009)
4. Nunamaker, J., Applegate, L., Konsynski, B.: Facilitating group creativity: experience with a group decision support system. J. Manag. Inf. Syst. **3**(4) (1987)
5. Javadi, E., Mahoney, J., Gebauer, J.: The impact of user interface design on idea integration in electronic brainstorming: an attention-based view. J. Assoc. Inf. Syst. **14**(1), 1–21 (2013)
6. Chen, C., Trotman, K., Zhou, F.: Nominal versus interacting electronic fraud brainstorming in hierarchical audit teams. Account. Rev. (2015)
7. Gallupe, R., Dennis, A., Cooper, W., Valacich, J., Bastianutti, L., Nunamaker, J.: Electronic brainstorming and group size. Acad. Manag. J. **35**(2), 350–369 (1992)
8. Coskun, H.: The effects of group size, memory instruction and session length on the creative performance in electronic brainstorming groups (2011)
9. Widjajaa, W., Yoshiia, K., Hagaa, K., Takahashia, M.: Discusys: multiple user real-time digital sticky-note affinity-diagram brainstorming system. In: 17th International KES 2013 Conference (2013)
10. Sielski, D., Kozakiewicz, W., Basiuras, M., Klaudia, G., Santorek, J., Kucharski, P., Grudzien, K.: Comparative analysis of multitouch interactive surfaces. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) Proceedings of the 2017 FedCSIS Conference, ACSIS. IEEE (2017, in press)
11. Chen, C., Wozniak, P., Romanowski, A., Obaid, M., Jaworski, T., Kucharski, J., Grudzien, K., Zhao, S., Fjeld M.: Using crowdsourcing for scientific analysis of industrial tomographic images. ACM Trans. Intell. Syst. Technol. **7**(4), 25p (2016). Article no. 52
12. Romanowski, A., Grudzień, K., Woźniak, P.: Contextual processing of ECT measurement information towards detection of process emergency states. In: Proceedings of the Thirteenth International Conference on Hybdrid Intelligent Systems (HIS 2013), Tunisia, pp. 292–298 (2013)
13. Jelliti, I., Romanowski, A., Grudzien, K.: Design of crowdsourcing system for analysis of gravitational flow using x-ray visualization. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) Proceedings of the 2016 FedCSIS Conference ACSIS, vol. 8, pp. 1613–1619. IEEE (2016)

Analysis of IMS/NGN Call Processing Performance Using Phase-Type Distributions Based on Experimental Histograms

Sylwester Kaczmarek(✉) and Maciej Sac

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 11/12 Narutowicza Street, 80-233 Gdańsk, Poland
{kasyl, msac}@eti.pg.gda.pl

Abstract. The paper describes our further research done with the proposed analytical and simulation traffic models of the Next Generation Network (NGN), which is standardized for delivering multimedia services with strict quality and includes elements of the IP Multimedia Subsystem (IMS). The aim of our models of a single IMS/NGN domain is to evaluate two standardized call processing performance parameters, which appropriate values are very important for satisfaction of users and overall success of the IMS/NGN concept. These parameters are mean Call Set-up Delay $E(CSD)$ and mean Call Disengagement Delay $E(CDD)$. Our latest investigations concern improving the conformity of the analytical results and experimental results obtained using the simulation model, which implements the operation of real network elements according to current standards and research. In this paper the results of calculations using PH/PH/1 queuing systems are presented, in which arrival and service distributions are phase-type distributions computed using maximum likelihood and distance minimization fitting algorithms based on experimental histograms. Presented latest results are compared to these obtained using other, previously investigated queuing systems. Additionally, computational complexity of all examined queuing systems is analyzed. As a result, some general remarks concerning all tested queuing systems and their applicability to NGN are provided.

Keywords: IMS · NGN · Call processing performance · Phase-type distributions · PH/PH/1 · Traffic model

1 Introduction

Our research concerns the Next Generation Network (NGN) [1], which is a standardized network architecture based on the IP Multimedia Subsystem (IMS) [2] (hence the names "IMS-based NGN" and "IMS/NGN" are commonly used), proposed for fulfilling the needs of current and future information society concerning delivering multimedia services. For a commercial success this new concept of telecommunication network should be properly designed, to guarantee the values of quality parameters, which are satisfactory for users (e.g. standardized call processing performance parameters [3, 4], which include mean Call Set-up Delay $E(CSD)$ mean Call

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_13

Disengagement Delay $E(CDD)$). To achieve this aim, proposition and application of appropriate traffic models is necessary to investigate the relations between network parameters and guaranteed quality parameters.

Performed review indicated that research on these aspects is in early phase and the majority of available traffic models do not include the operation of all standardized IMS/NGN elements (such as RACF, Resource and Admission Control Functions, responsible for resource control) or service scenarios and do not allow assessment of call processing performance parameters. The fact that IMS/NGN is now in the implementation stage and the problem of ensuring quality becomes more and more important led us to start investigations in this area.

In the paper we continue our research concerning the formerly proposed simulation [5] and analytical [6] traffic model of a single domain of IMS/NGN, which allow evaluation of $E(CSD)$ and $E(CDD)$. Our latest investigations focus on finding the best queuing system for the analytical model to achieve the best conformity with the experimental results provided by the simulation model. In the simulation model not theoretical queuing systems but the operation of real network elements and standardized call scenarios are accurately implemented (the simulated elements process SIP and Diameter messages according to standards). Consequently, the simulation model reflects the phenomena taking place in real IMS/NGN network and can be considered as a reference for evaluation of quality of the analytical results. In the previous stages of our research we examined the following queuing systems in our analytical model (as IMS/NGN elements are servers with lots of memory available for message queues and in all examined cases high queue lengths are very improbable – due to constraints on $E(CSD)$ and $E(CDD)$ – the analysis of the system can be readily done using queuing models with infinite waiting room capacity):

- M/G/1 based on two moments of service distribution – used in our first approach [6] although were aware of the fact that intervals between messages at the inputs of IMS/NGN elements are generally not exponential; the results were under many conditions acceptable but poor conformity of calculations and simulations could be observed under high load and also when IMS/NGN elements were connected using links with relatively low bandwidth,
- G/G/1 based on two or three moments of arrival distribution and two moments of service distribution [7] – used in the next step of our research; they did not significantly improve the results,
- PH/PH/1 with arrival and service distributions represented by phase-type distributions [8–11] obtained using moment matching algorithms [12]; good conformity between analytical and simulation results were observable but we still did not find one universal queuing system for all considered data sets.

The completion of the above mentioned research, introduced in this paper, are investigations with PH/PH/1 queuing systems obtained using maximum likelihood and distance minimization fitting algorithms, which base on whole experimental histograms (not only on their moments) and due to this fact can lead to better results than for the previously examined queuing systems. In this paper the results for PH/PH/1 queuing systems obtained using maximum likelihood and distance minimization fitting algorithms are compared to the best results achieved in the previous stages. Moreover, we

present the analysis of computational complexity for the queuing systems used in all stages of our research (M/G/1, G/G/1, PH/PH/1 with all investigated fitting algorithms), which has not been evaluated before. The rest of the text is organized as follows. Basic details about the proposed IMS/NGN traffic models are presented in Sect. 2. Elementary information about phase-type distributions, algorithms for fitting this type of distributions to arrival and service distributions in IMS/NGN and analyzing PH/PH/1 queuing systems is provided in Sect. 3. The results of the performed investigations are presented and discussed in Sect. 4. Summary and future work are described in Sect. 5.

2 Traffic Model of IMS/NGN

Our research concerns the ITU-T NGN architecture, which is the most advanced of all available NGN solutions [13]. Based on current standards and scientific work we have proposed an analytical [6] and simulation [5] model of a single ITU-T NGN domain, which contain the elements depicted in Fig. 1, performing the following roles [6, 7, 12]:

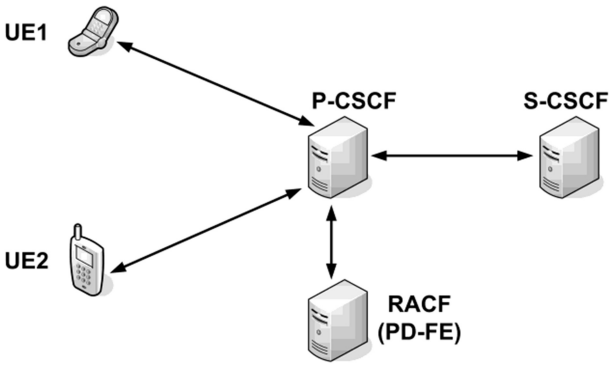


Fig. 1. Model of a single domain of IMS/NGN [6, 14, 15].

- User Equipments (UEs): user terminals generating call set-up and disengagement requests, registered in the domain,
- P-CSCF (Proxy – Call Session Control Function): the server responsible for receiving all messages from UEs and forwarding them to the S-CSCF server,
- S-CSCF (Serving – Call Session Control Function): the main server handling all calls,
- RACF (Resource and Admission Control Functions): the unit of the transport stratum, which allocates resources for a new call and releases resources during call disengagement.

The above mentioned network elements participate in a standardized voice call scenario [14–18] depicted in Fig. 2. For resource operations (allocation, release) Diameter protocol [19] messages are exchanged between P-CSCF and RACF. Other

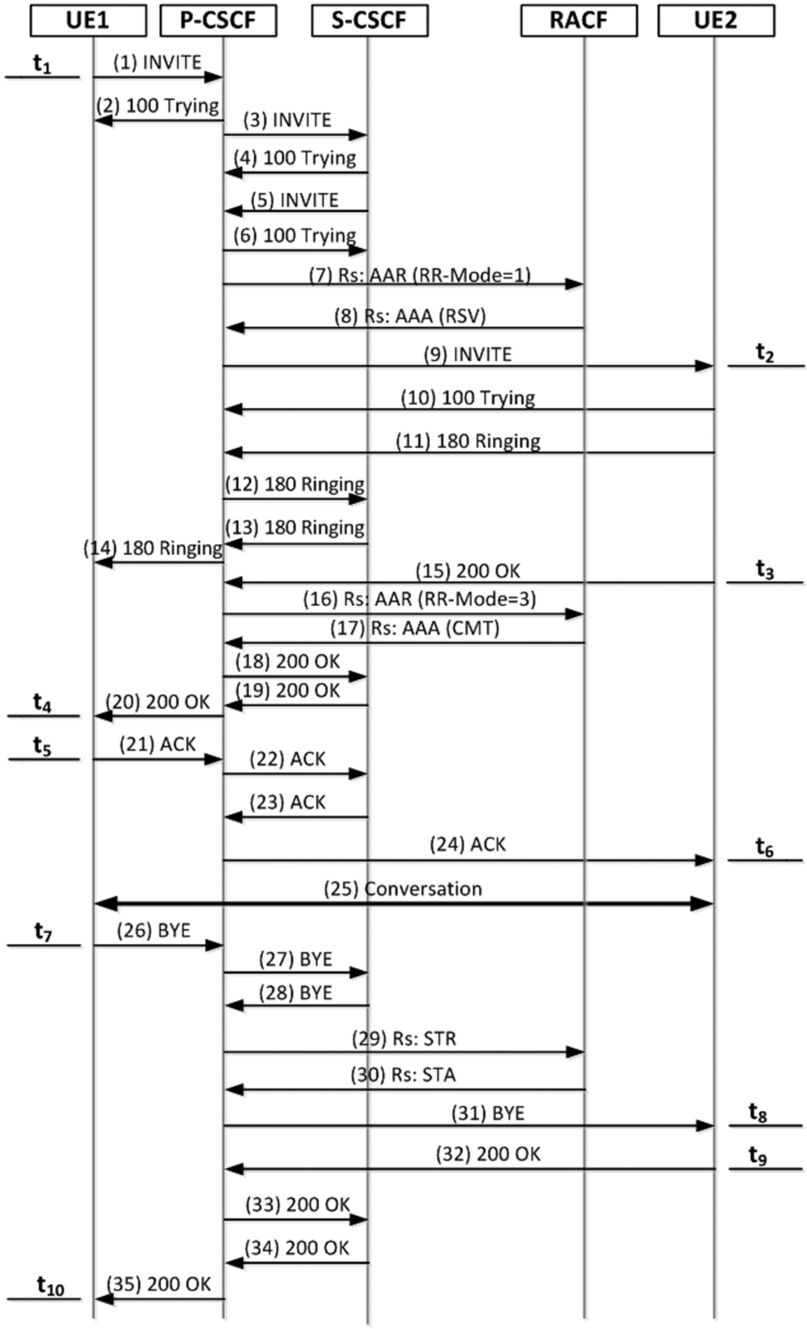


Fig. 2. Call set-up (messages 1–24) and call disengagement (messages 26–35) scenario in a single domain of IMS/NGN [6].

elements communicate using SIP protocol [20]. More details regarding the call scenario are out of the scope of this paper and can be found in [6].

Based on the t_1–t_{10} times illustrated in Fig. 2 and the definitions given by the ITU-T [3, 4], Call Set-up Delay (*CSD*) and Call Disengagement Delay (*CDD*) for a single call can be calculated as follows:

$$CSD = (t_2 - t_1) + (t_4 - t_3) + (t_6 - t_5) \quad (1)$$

$$CDD = (t_8 - t_7) + (t_{10} - t_9) \quad (2)$$

Formulas (1) and (2) are used to obtain the output variables evaluated by our traffic models, which are mean *CSD*, *E*(*CSD*), and mean *CDD*, *E*(*CDD*) [3, 4]. The analytical and simulation models represent different approaches to assess *E*(*CSD*) and *E*(*CDD*) (more details will be provided later). However, both of them use a similar set of the input variables, from which the most vital are [5, 6]:

- λ_{INV}: the intensity of aggregated call set-up requests (SIP INVITE messages from many terminals) generated by UE1 with exponential intervals,
- T_{INV1} and T_{INV2}: the times of processing SIP INVITE messages by P-CSCF and S-CSCF correspondingly,
- a_k (k = 1, 2, …, 8): the factors determining times of processing other SIP and Diameter messages by CSCF servers according to Table 1,
- T_X: the time of processing messages by RACF,
- lengths and bandwidths of optical links, lengths of transmitted messages: values necessary to calculate communication times between network elements.

Table 1. Message processing times of CSCF servers.

Message	P-CSCF processing times	S-CSCF processing times
SIP INVITE	T_{INV1}	T_{INV2}
SIP 100 Trying	$T_{TR1} = a_1 \cdot T_{INV1}$	$T_{TR2} = a_1 \cdot T_{INV2}$
SIP 180 Ringing	$T_{RING1} = a_2 \cdot T_{INV1}$	$T_{RING2} = a_2 \cdot T_{INV2}$
SIP 200 OK (answer to INVITE)	$T_{OK1} = a_3 \cdot T_{INV1}$	$T_{OK2} = a_3 \cdot T_{INV2}$
SIP ACK	$T_{ACK1} = a_4 \cdot T_{INV1}$	$T_{ACK2} = a_4 \cdot T_{INV2}$
SIP BYE	$T_{BYE1} = a_5 \cdot T_{INV1}$	$T_{BYE2} = a_5 \cdot T_{INV2}$
SIP 200 OK (answer to BYE)	$T_{OKBYE1} = a_6 \cdot T_{INV1}$	$T_{OKBYE2} = a_6 \cdot T_{INV2}$
Diameter AAA	$T_{AAA1} = a_7 \cdot T_{INV1}$	$T_{AAA2} = a_7 \cdot T_{INV2}$
Diameter STA	$T_{STA1} = a_8 \cdot T_{INV1}$	$T_{STA2} = a_8 \cdot T_{INV2}$

The simulation model [5, 6], developed in the OMNeT++ framework [21], implements the operation of the network elements and call scenario depicted in Figs. 1 and 2. CSCF servers in the model include Central Processing Units (CPUs), which are responsible for handling messages incoming from other elements according to the assumed call scenario (Fig. 2). When CPUs are busy incoming messages are stored in CPU queues. Other network elements in the model respond with a particular delay

(RACF responsible for resource allocation and release as well as UEs representing many user terminals). Each element of the model includes communication queues (one for each outgoing link), which buffer messages before sending them through links when links are busy. During simulation the t_1–t_{10} times (Fig. 2) are stored for all performed call scenarios and used to obtain *CSD* (1) and *CDD* (2). At the end of simulation *E*(*CSD*) and *E*(*CDD*) are calculated along with corresponding confidence intervals for a defined confidence level [5, 6].

In the analytical model mean values of theoretical delays introduced by the elements depicted in Fig. 1 are calculated and appropriately summed to obtain mean *CSD* and mean *CDD* [6, 7, 12]. These component delays include:

- mean message waiting times in communication queues; for calculation of these times mathematical models of queuing systems are used,
- message transmission times (message lengths divided by links bandwidths),
- propagation times (equal to 5 μs/km – optical links are assumed),
- mean message waiting times in CSCF servers CPU queues; for calculation of these times mathematical models of queuing systems are used,
- message processing times by CSCF servers CPUs (Table 1) and RACF (T_X input variable).

Due to the fact that the simulation model implements real IMS/NGN elements and call scenario, it reflects the phenomena occurring in the real network (message transmission, queuing, handling, etc.) and can be used to assess the quality of the analytical model based on theoretical calculations. More details about both models cannot be provided in this paper due to lack of space and can be found in [5–7, 12].

3 Phase-Type Distributions

Phase-type distributions [8–11] are probability distributions that result from a system of one or more inter-related Poisson processes occurring in sequence or phases. Special cases of continuous phase-type distributions are [8–11]: exponential distribution, Erlang distribution, Coxian distribution, Hyperexponential distribution (also called a mixture of exponential) and Hypoexponential distribution.

Our interest in phase-type distributions is related to their very important feature. As the set of phase-type distributions is dense in the field of all positive-valued distributions [8–11], they can represent or approximate (with any accuracy) any positive valued distribution. There are several algorithms for fitting different subsets of phase-type distributions to experimental data based on a specified number of first moments [8–11] or whole experimental histograms [11, 22–24].

Using such algorithms phase-type distributions can be fitted to all arrival and service distributions in the network [25]. After that, some characteristics of queues, like mean message waiting times (required in our research), can be calculated through analysis of PH/PH/1 queuing systems. Such analysis can be done by solving quasi-birth-and-death (QBD) Markov chains using matrix-analytic methods [26, 27]. Alternatively, formulas for analyzing queues with MAP (Markovian Arrival Process) or BMAP (Batch Markovian Arrival Process) distributions [28, 29] can be used as

these types of distributions are supersets of phase-type distributions. Several solvers are available and we have tested two [30, 31] designed for the MATLAB environment used in our research. Both of the tested solvers provided very similar results, however the second one [31] was chosen for further use due to faster and direct calculation of mean waiting time [12].

As already mentioned, in this paper we perform investigations with PH/PH/1 queuing systems, in which phase-type distributions are fitted using maximum likelihood and distance minimization algorithms based on whole experimental histograms. The set of investigated algorithms included (the first three belong to the family of maximum likelihood algorithms, while the last two are distance minimization algorithms):

- EMpht [11] – fitting general phase-type distributions using Expectation-Maximisation (EM) algorithm; in our research distributions of order 4 were fitted with the number of iterations equal to 6000; due to large computation times the number of samples of each experimental histogram was limited to 100000,
- GFIT1 [22] – fitting Hyper-Erlang phase-type distributions using Expectation-Maximisation (EM) algorithm; the order of fitted distributions was set to 4; the number of samples of each experimental histogram was limited to 1000000,
- GFIT2 [22] – the same algorithm as in GFIT1, however, here distributions of order 10 were fitted and the number of samples of each experimental histogram was limited to 700000,
- PhFit1 [24] – fitting acyclic phase-type distributions using distance minimization algorithm; distributions of order 4 were fitted with total number of iterations equal to 3000 (3 rounds with 1000 iterations); the investigated distance metric was relative entropy; the number of samples of each experimental histogram was limited to 200000,
- PhFit3 [24] – the same algorithm as in PhFit1, however, here distributions of order 10 were fitted and the number of samples of each experimental histogram was limited to 100000.

The number of iterations was chosen experimentally for each of the algorithms according to its convergence speed.

To apply the above mentioned fitting algorithms in the analytical model, a full description of arrival and service distributions for all IMS/NGN elements was necessary. This was not a problem for service distributions as they result from times of processing individual messages by network elements (defined by the input variables; for links they are determined by message lengths and link bandwidths) and the set of messages handled by each element (given by the assumed call scenario).

When it comes to arrival distributions, only theoretical intensity of messages handled by all networks elements is known from the call scenario and λ_{INV}. Therefore, in this case phase-type distributions were fitted to experimental histograms recorded using the simulation model [25]. The results of the performed investigations are described in the next section.

4 Results

In order to compare the latest $E(CSD)$ and $E(CDD)$ results with these obtained in our previous papers [6, 7, 12], the same input data sets (presented in Table 2) were used. We also assumed identical message lengths (Table 3) as well as the a_k factors: $a_1 = 0.2$, $a_2 = 0.2$, $a_3 = 0.6$, $a_4 = 0.3$, $a_5 = 0.6$, $a_6 = 0.3$, $a_7 = 0.6$, $a_8 = 0.6$ (Table 1).

Table 2. Input data sets.

Data set	λ_{INV} [1/s]	T_{INV1} [ms]	T_{INV2} [ms]	T_X [ms]	Links
1a	5–250	0.5	0.5	3	No links
1b	5–250	0.5	0.5	3	300 km, 10 Mbit/s
1c	5–250	0.5	0.5	3	300 km, 100 Mbit/s

Table 3. Message lengths [32].

Message	Length in bytes
SIP INVITE	930
SIP 100 Trying	450
SIP 180 Ringing	450
SIP 200 OK (answer to INVITE)	990
SIP ACK	630
SIP BYE	510
SIP 200 OK (answer to BYE)	500
Diameter messages	750

The values of the input variables were taken according to current research (e.g. message lengths, which were measured experimentally in paper [32]) and our experience regarding telecommunication networks (e.g. the length of 300 km assumed for all links represents a typical distance between larger cities in Poland). Due to the fact that in our traffic models quite a large number of the input variables is used (Sect. 2), in the research presented in this paper we focused on call set-up request intensity, which is a typical and very commonly investigated parameter for telecommunication networks. Additionally, we examined different parameters of links (length and bandwidth), which have a significant impact on the final results.

As already mentioned in Sect. 3, in the research described in this paper phase-type distributions were fitted to message arrival and service distributions for all network elements, using maximum likelihood and distance minimization algorithms. In the next step mean message waiting times were calculated through analysis of PH/PH/1 queuing systems. The resultant mean waiting times were applied to compute the considered total system responses – $E(CSD)$ and $E(CDD)$.

Apart from fitting phase-type distributions to both arrival and service distributions of all elements (PH/PH/1 queuing systems), we also assumed that either arrival or service distributions are exponential (special cases of phase-type distributions), which resulted in M/PH/1 and PH/M/1 queuing systems correspondingly.

The above mentioned computations required full description of arrival and service distributions for all network elements. As arrival distributions are not fully described by the input variables (Sect. 3), they were obtained experimentally, using our simulation model. The simulations were performed with the following parameters [7, 12]:

- warm-up period: 1500 s,
- 5 measurement periods,
- 0.95 confidence level,
- simulation is finished when confidence intervals for $E(CSD)$ and $E(CDD)$ do not exceed 5% of mean Call Set-up Delay and mean Call Disengagement Delay or when total simulation time exceeds 10000 s; with such stop conditions at least 10000 message inter-arrival times were obtained during each simulation.

The results obtained for data sets 1a–1c (Table 2) are presented in Figs. 3, 4 and 5 as $E(CSD)$ and $E(CDD)$ versus call set-up request intensity curves. In each subfigure (Figs. 3, 4 and 5 contain six subfigures) we present results of simulations, calculations

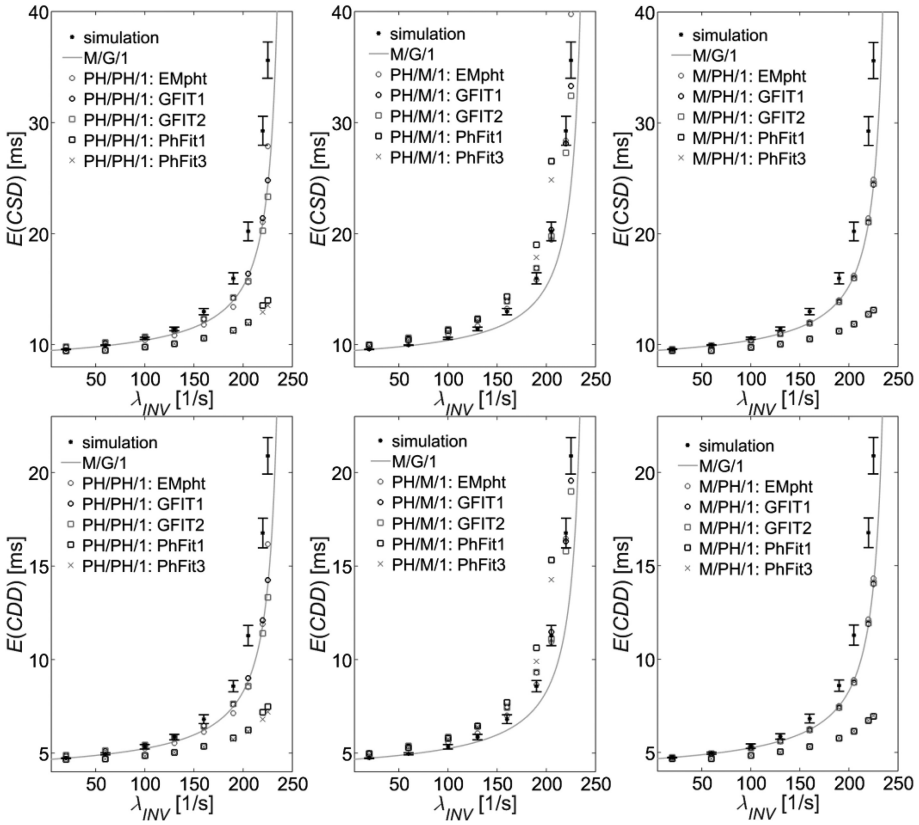


Fig. 3. Results for data set 1a (no links); left: PH/PH/1 queuing systems, center: PH/M/1 queuing systems, right: M/PH/1 queuing systems; results at the top refer to $E(CSD)$ and at the bottom – to $E(CDD)$.

using M/G/1 queuing systems (our initial approach [6]) and analytical approaches based on phase type distributions (PH/PH/1, PH/M/1, M/PH/1) obtained using maximum likelihood and distance minimization algorithms (Sect. 3). The names of the fitting algorithms are included in legends (after colons).

Generally, it can be noticed that conformity of calculations and simulations of call processing performance parameters is dependent on many factors, including queuing systems used in the analytical model (Figs. 3, 4 and 5). Very important is also call set-up request intensity as well as available link bandwidth (links involve communication queues, which load is related to message intensity and link bandwidth; in data set 1a network elements are connected directly without links, thus, there are no communication queues).

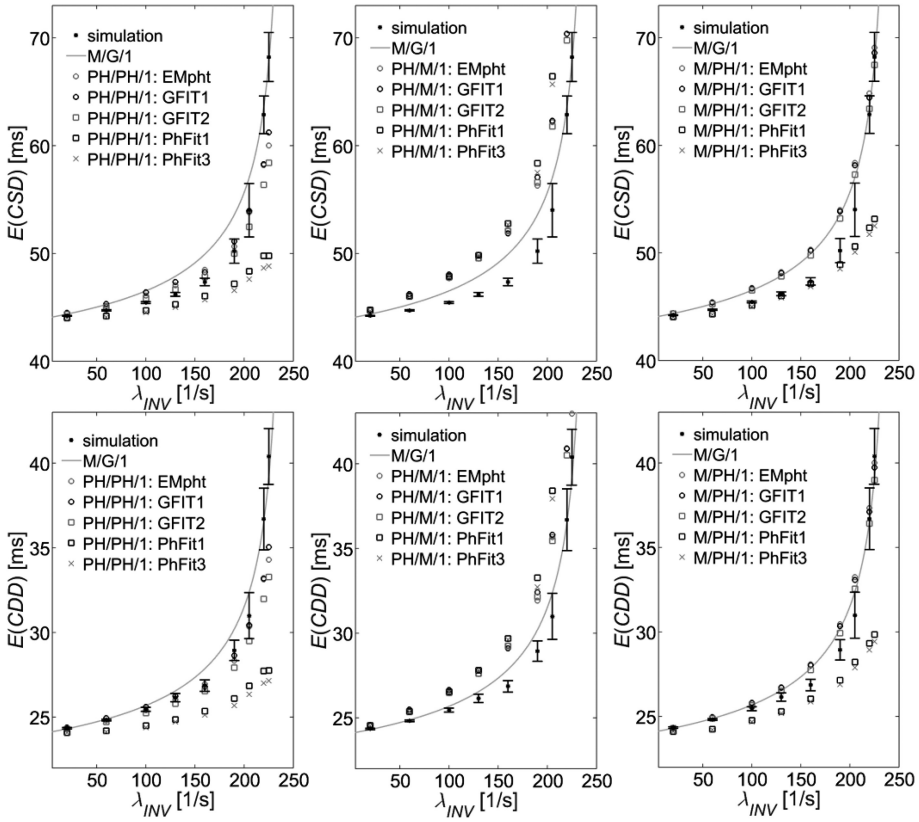


Fig. 4. Results for data set 1b (link length 300 km and bandwidth 10 Mbit/s); left: PH/PH/1 queuing systems, center: PH/M/1 queuing systems, right: M/PH/1 queuing systems; results at the top refer to *E*(*CSD*) and at the bottom – to *E*(*CDD*).

We can observe that the relations between analytical and simulation results are similar for $E(CSD)$ and $E(CDD)$, when the same queuing systems are investigated. However, $E(CSD)$ values are always larger than $E(CDD)$, since the process of call set-up is much more complicated, comparing to call disengagement, and involves more messages. For each analyzed data set (Table 2) PhFit1 and PhFit3 algorithms give analytical results very far from simulations and other calculations. This rule applies especially for PH/PH/1 and M/PH/1 queuing systems with PhFit1 and PhFit3 algorithms. As a result, these algorithms are not useful in practice.

Based on Figs. 3, 4 and 5, it can be also noticed that in the majority of analyzes cases PH/M/1 queuing systems overestimate $E(CSD)$ and $E(CDD)$, while M/PH/1 queuing systems give results very similar to M/G/1 (this does not apply to PhFit1 and PhFit3 algorithms).

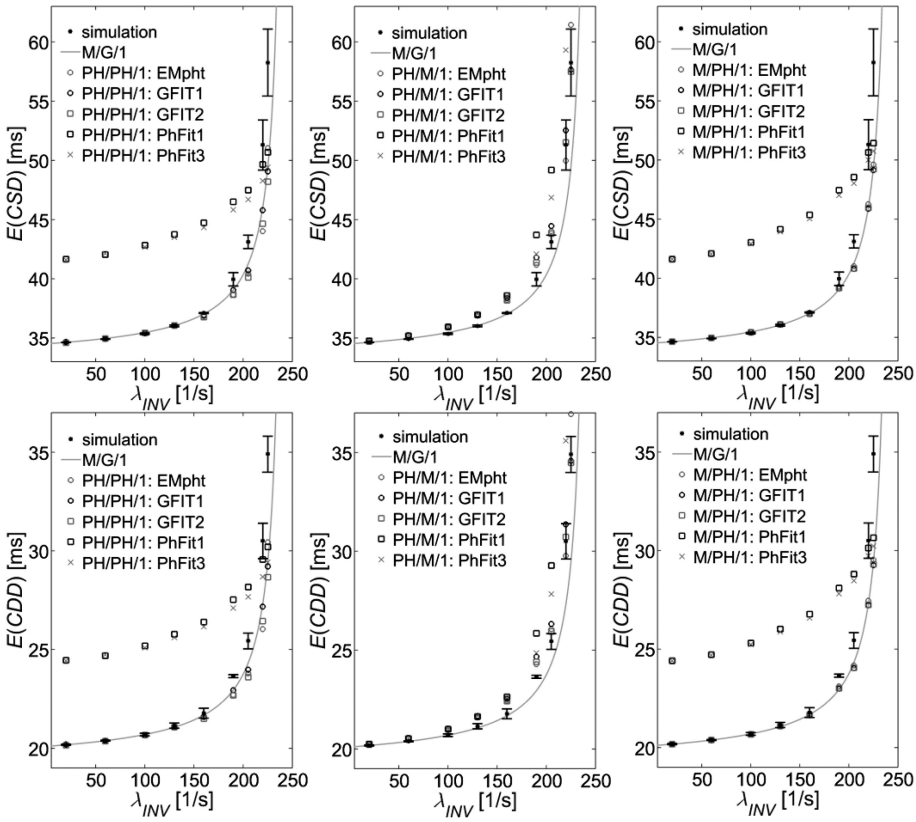


Fig. 5. Results for data set 1c (link length 300 km and bandwidth 100 Mbit/s); left: PH/PH/1 queuing systems, center: PH/M/1 queuing systems, right: M/PH/1 queuing systems; results at the top refer to $E(CSD)$ and at the bottom – to $E(CDD)$.

In order to thoroughly examine the subject of the paper, we have also done some more extended research with the root-mean-square error (RMSE) [7, 12] defined as follows:

$$\mathrm{RMSE} = \sqrt{\underset{\lambda_{INV} \in \Lambda}{E} \left(Y_{\mathrm{simulation}} - Y_{\mathrm{analytical}}\right)^2}, \tag{3}$$

where Y is either $E(CSD)$ or $E(CDD)$ and $E()$ is the averaging operator over a particular set of call set-up request intensities $\lambda_{INV} \in \Lambda$. To investigate different network conditions, for all data sets the following sets of λ_{INV} were assumed [7, 12]:

- "green" – IMS/NGN elements are low loaded and $E(CSD)$, $E(CDD)$ change very little with call set-up request intensity (λ_{INV} = 20, 60, 100 [1/s]),
- "yellow" – IMS/NGN elements are quite highly loaded and $E(CSD)$, $E(CDD)$ start noticeably increasing with call set-up request intensity (λ_{INV} = 130, 160, 190 [1/s]),
- "red" – IMS/NGN elements are overloaded and $E(CSD)$, $E(CDD)$ start going to infinity (λ_{INV} = 205, 220, 225 [1/s]),
- "gr-yel" – the set including all call set-up request intensities from the "green" set and almost all call set-up request intensities from the "yellow" set (λ_{INV} = 20, 60, 100, 130, 160 [1/s]),
- "all" – the set containing all call set-up request intensities from the "green", "yellow" and "red" sets (λ_{INV} = 20, 60, 100, 130, 160, 190, 205, 220, 225 [1/s]).

The results of RMSE computations are presented in Tables 4, 5 and 6. The RMSE values were calculated for the analytical results presented in Figs. 3, 4 and 5 as well as for other queuing systems investigated in our previous papers – G/G/1 based on moments [7] ("G/G/1" in Tables 4, 5 and 6), PH/PH/1 with phase-type distributions obtained using moment matching algorithms [12] ("PH/PH/1 moments" in Tables 4, 5 and 6). Two best results for each of these previous queuing systems are included along with the names of G/G/1 formulas and PH/PH/1 fitting algorithms based on moments. The names are provided in the brackets below the results. In each column of Tables 4, 5 and 6 we marked two best (bold and underlined font) and two worst RMSE results (bold and italic font with gray background).

The obtained RMSE values demonstrate that for various IMS/NGN parameters the best analytical $E(CSD)$ and $E(CDD)$ results can be achieved using different queuing systems (PH/PH/1 with phase-type distributions based on whole experimental histograms or only on their moments; G/G/1; M/G/1). Moreover, particular fitting algorithms for PH/PH/1 and G/G/1 formulas offer different efficiency. It is also important that often several of the above mentioned approaches offer very good performance (RMSE about 0.5 ms or even smaller), however, our aim is to find the best one.

In the majority of cases PH/PH/1 queuing systems (and their special cases – PH/M/1, M/PH/1) with proper fitting algorithms based on whole experimental histograms (EMpht, GFIT1, GFIT2) are the best or almost the best in terms of RMSE. However, the differences to the other queuing systems are often minimal. To decide which queuing system is the most suitable for IMS/NGN, we also investigated computational complexity (the time required to obtain analytical results for particular queuing systems). It is especially important in engineering applications, in which both

Table 4. RMSE for data set 1a (no links).

	RMSE for $E(CSD)$, [ms]					RMSE for $E(CDD)$, [ms]				
	green	*yellow*	*red*	*gr-yel*	*all*	*green*	*yellow*	*red*	*gr-yel*	*all*
M/G/1	**0,110**	1,317	8,399	0,506	4,909	**0,056**	0,727	5,089	0,290	2,968
PH/PH/1: EMpht	0,134	1,656	7,031	0,592	4,171	0,082	0,941	4,225	0,346	2,499
PH/PH/1: GFIT1	0,192	1,099	8,041	0,329	4,687	0,130	0,593	4,871	0,191	2,834
PH/PH/1: GFIT2	0,214	1,081	9,179	0,343	5,337	0,145	0,581	5,580	0,201	3,240
PH/PH/1: PhFit1	0,644	3,203	16,209	1,375	9,547	0,396	1,916	9,983	0,839	5,873
PH/PH/1: PhFit3	0,661	***3,263***	16,622	1,386	9,788	0,406	***1,953***	10,240	0,846	6,023
PH/M/1: EMpht	0,401	**0,224**	2,482	0,351	1,457	0,264	**0,172**	1,588	0,241	0,935
PH/M/1: GFIT1	0,622	0,942	**1,507**	0,770	**1,087**	0,403	0,672	**0,821**	0,509	**0,655**
PH/M/1: GFIT2	0,653	0,926	**2,203**	0,771	**1,430**	0,422	0,661	**1,232**	0,508	**0,843**
PH/M/1: PhFit1	0,542	1,982	***201,18***	0,835	***116,15***	0,353	1,336	***125,84***	0,550	***72,658***
PH/M/1: PhFit3	0,508	1,362	***71,779***	0,748	***41,450***	0,332	0,945	***44,967***	0,495	***25,968***
M/PH/1: EMpht	**0,102**	1,301	8,020	0,481	4,691	**0,051**	0,717	4,852	0,275	2,832
M/PH/1: GFIT1	0,126	1,411	8,395	0,533	4,915	0,067	0,787	5,087	0,307	2,972
M/PH/1: GFIT2	0,118	1,388	8,330	0,520	4,876	0,061	0,772	5,045	0,299	2,947
M/PH/1: PhFit1	***0,693***	***3,221***	16,832	***1,403***	9,903	***0,426***	***1,926***	10,370	***0,856***	6,094
M/PH/1: PhFit3	***0,688***	3,218	16,827	***1,399***	9,899	***0,423***	1,924	10,366	***0,854***	6,092
G/G/1 1st [7] (name of formula in brackets)	0,113 (Diff22)	0,802 (My2)	6,656 (King)	**0,265** (My1)	5,119 (My2)	0,080 (Diff22)	**0,439** (My2)	3,986 (King)	**0,172** (My1)	3,106 (My2)
G/G/1 2nd [7] (name of formula in brackets)	0,266 (My1)	0,925 (My1)	7,029 (Page)	**0,306** (KLB)	5,310 (King2)	0,181 (My1)	0,489 (My1)	4,227 (Page)	**0,188** (KLB)	3,225 (King2)
PH/PH/1 moments 1st [12] (name of algorithm in brackets)	0,111 (M/PH/1: PH2 (3))	**0,700** (PH/M/1: PH2 (3))	2,519 (PH/M/1: PH2 (5))	0,343 (PH/PH/1: ME (5))	1,542 (PH/M/1: PH2 (5))	0,057 (M/PH/1: PH2 (3))	0,515 (PH/M/1: PH2 (3))	1,431 (PH/M/1: PH2 (5))	0,192 (PH/PH/1: ME (5))	0,902 (PH/M/1: PH2 (5))
PH/PH/1 moments 2nd [12] (name of algorithm in brackets)	0,111 (M/PH/1: PH2 (5))	0,719 (PH/M/1: PH2 (5))	2,529 (PH/M/1: PH2 (3))	0,357 (PH/PH/1: PH2 (5))	1,545 (PH/M/1: PH2 (3))	0,057 (M/PH/1: PH2 (5))	0,526 (PH/M/1: PH2 (5))	1,437 (PH/M/1: PH2 (3))	0,204 (PH/PH/1: PH2 (5))	0,903 (PH/M/1: PH2 (3))

accuracy and complexity count. In the next part we present some details about calculation times in the analytical model for one data set (Intel Core i7-2670QM CPU 8 × 2.20 GHz, 8 GB RAM, Windows 7 Professional SP1 64-bit):

- for M/G/1 [6] computations are the fastest of all (several seconds) and require only the input variables (there is no need to use experimental histograms),
- for G/G/1 [7] computations last about one minute (experimental histograms are needed to obtain second and third moments of arrival distributions, which are used in calculations),
- for PH/PH/1 (and their special cases – PH/M/1, M/PH/1) with moment matching algorithms [12] computations take from one to several minutes (experimental histograms are needed to obtain second, third, fourth and fifth moments of arrival distributions, which are used in calculations),

Table 5. RMSE for data set 1b (link length 300 km and bandwidth 10 Mbit/s).

	RMSE for *E*(*CSD*), [ms]					RMSE for *E*(*CDD*), [ms]				
	green	*yellow*	*red*	*gr-yel*	*all*	*green*	*yellow*	*red*	*gr-yel*	*all*
M/G/1	0,705	2,637	**1,805**	1,477	1,889	0,133	0,954	**1,059**	0,489	**0,826**
PH/PH/1: EMpht	0,460	0,852	5,450	0,724	3,196	**0,031**	0,356	4,095	**0,068**	2,373
PH/PH/1: GFIT1	0,666	0,990	4,834	0,834	2,875	0,111	**0,187**	3,702	0,099	2,141
PH/PH/1: GFIT2	**0,290**	**0,410**	6,839	**0,374**	3,959	0,138	0,658	5,011	0,242	2,919
PH/PH/1: PhFit1	1,201	2,274	13,657	1,341	8,024	***1,104***	2,216	9,396	1,355	5,610
PH/PH/1: PhFit3	1,311	2,744	14,599	1,547	8,610	***1,177***	2,511	***9,999***	1,488	***5,991***
PH/M/1: EMpht	1,533	4,848	7,132	2,863	5,057	0,655	2,368	3,929	1,359	2,675
PH/M/1: GFIT1	***1,759***	5,178	7,782	2,934	5,492	0,805	2,580	4,315	1,398	2,940
PH/M/1: GFIT2	1,565	4,974	6,860	2,862	4,975	0,675	2,450	3,745	1,357	2,613
PH/M/1: PhFit1	***1,622***	***6,031***	***19,388***	***3,175***	***11,76***	0,714	***3,129***	***11,619***	***1,563***	***6,959***
PH/M/1: PhFit3	1,591	***5,618***	***16,077***	***3,131***	***9,875***	0,694	***2,869***	9,521	***1,538***	5,755
M/PH/1: EMpht	0,864	3,001	2,807	1,727	2,424	0,226	1,170	1,364	0,631	1,046
M/PH/1: GFIT1	0,839	2,894	2,566	1,674	2,285	0,210	1,102	1,292	0,598	0,988
M/PH/1: GFIT2	0,705	2,414	**1,976**	1,415	**1,846**	0,128	0,796	1,217	0,438	0,843
M/PH/1: PhFit1	0,979	0,870	10,786	0,812	6,273	0,949	1,343	7,589	0,974	4,483
M/PH/1: PhFit3	1,013	1,163	11,352	0,896	6,614	0,979	1,535	7,957	1,054	4,713
G/G/1 1st [7] (name of formula in brackets)	0,550 (Diff22)	1,042 (Diff22)	4,729 (Daley)	0,660 (Diff22)	3,229 (My2)	0,123 (KLB)	**0,300** (My1)	2,625 (Daley)	0,206 (My1)	2,223 (My2)
G/G/1 2nd [7] (name of formula in brackets)	0,698 (KLB)	1,075 (Diff21)	4,867 (My2)	1,028 (My1)	3,307 (King2)	0,166 (King2)	0,331 (Diff21)	2,742 (Page)	0,280 (KLB)	2,381 (King2)
PH/PH/1 moments 1st [12] (name of algorithm in brackets)	**0,405** (PH/PH/1: APH2 (3))	**0,515** (PH/PH/1: APH2 (3))	2,025 (M/PH/1: PH2 (3))	**0,502** (PH/PH/1: APH2 (3))	**1,874** (M/PH/1: PH2 (3))	**0,038** (PH/PH/1: PH2 (5))	0,428 (PH/PH/1: ME (3–4))	**1,181** (M/PH/1: PH2 (3))	**0,069** (PH/PH/1: PH2 (5))	**0,834** (M/PH/1: PH2 (3))
PH/PH/1 moments 2nd [12] (name of algorithm in brackets)	0,500 (PH/PH/1: ME (5))	0,613 (PH/PH/1: ME (5))	2,025 (M/PH/1: PH2 (5))	0,602 (PH/PH/1: ME (5))	**1,874** (M/PH/1: PH2 (5))	0,060 (PH/PH/1: APH1 (3))	0,474 (PH/PH/1: APH1 (3), PH2 (3))	**1,181** (M/PH/1: PH2 (5))	0,071 (PH/PH/1: APH1 (3), PH2 (3))	**0,834** (M/PH/1: PH2 (5))

– PH/PH/1 (and their special cases – PH/M/1, M/PH/1) with maximum likelihood and distance minimization fitting algorithms are the worst case; fitting phase-type distributions to whole experimental histograms lasts from 30 min to several hours with parameters of the algorithms described in the previous section; therefore, fitting was performed offline and its results were saved to disk for further use.

Taking into account all the aspects presented in the previous part of this section (accuracy and complexity), we propose to use the following queuing systems in the analytical model of a single domain of IMS/NGN. For data sets 1a (no links; Table 4) and 1c (links with relatively high bandwidth; Table 6) we suggest using simple M/G/1

Table 6. RMSE for data set 1c (link length 300 km and bandwidth 100 Mbit/s).

	RMSE for *E*(*CSD*), [ms]					RMSE for *E*(*CDD*), [ms]				
	green	*yellow*	*red*	*gr-yel*	*all*	*green*	*yellow*	*red*	*gr-yel*	*all*
M/G/1	0,069	0,518	6,014	**0,064**	3,485	**0,003**	0,410	3,701	0,058	2,150
PH/PH/1: EMpht	**0,020**	0,758	6,114	0,120	3,557	0,031	0,564	3,758	0,112	2,194
PH/PH/1: GFIT1	0,053	0,529	6,329	0,093	3,667	0,016	0,421	3,900	0,085	2,265
PH/PH/1: GFIT2	**0,040**	0,783	7,166	0,163	4,162	0,021	0,579	4,421	0,129	2,574
PH/PH/1: PhFit1	7,045	7,232	5,149	7,265	6,543	4,248	4,343	3,188	4,380	3,961
PH/PH/1: PhFit3	7,004	6,826	5,783	7,115	6,560	4,223	4,086	3,584	4,286	3,974
PH/M/1: EMpht	0,347	1,083	2,071	0,689	1,364	0,176	0,584	1,311	0,380	0,835
PH/M/1: GFIT1	0,407	1,415	**1,101**	0,783	**1,061**	0,215	0,788	**0,738**	0,439	**0,636**
PH/M/1: GFIT2	0,384	1,171	**0,605**	0,712	**0,793**	0,200	0,636	**0,395**	0,392	**0,447**
PH/M/1: PhFit1	0,386	2,381	***131,94***	0,835	***76,188***	0,201	1,394	***82,503***	0,473	***47,640***
PH/M/1: PhFit3	0,391	1,594	***33,468***	0,836	***19,346***	0,204	0,900	***20,958***	0,473	***12,112***
M/PH/1: EMpht	0,069	**0,390**	5,903	0,076	3,416	**0,002**	**0,324**	3,630	**0,035**	2,104
M/PH/1: GFIT1	0,045	0,486	6,266	0,073	3,629	0,014	0,390	3,858	0,067	2,239
M/PH/1: GFIT2	0,052	0,469	6,206	0,072	3,593	0,009	0,378	3,819	0,060	2,216
M/PH/1: PhFit1	***7,123***	***7,905***	5,039	***7,539***	6,798	***4,289***	***4,741***	3,132	***4,539***	4,110
M/PH/1: PhFit3	***7,079***	***7,609***	5,221	***7,411***	6,715	***4,259***	***4,551***	3,248	***4,456***	4,058
G/G/1 1st [7] (name of formula in brackets)	0,111 (KLB)	0,584 (My2)	5,089 (Page)	0,121 (KLB)	4,132 (My2)	0,035 (KLB)	0,419 (My2)	3,120 (Page)	0,098 (KLB)	2,548 (My2)
G/G/1 2nd [7] (name of formula in brackets)	0,150 (My1)	0,774 (King2)	5,173 (Daley)	0,156 (King2)	4,341 (King2)	0,055 (My1)	0,573 (King2)	3,174 (Daley)	0,110 (King2)	2,686 (King2)
PH/PH/1 moments 1st [12] (name of algorithm in brackets)	0,051 (PH/PH/1: APH2 (3))	**0,426** (M/PH/1: PH2 (3))	1,556 (PH/M/1: PH2 (3))	**0,070** (M/PH/1: PH2 (3))	1,137 (PH/M/1: PH2 (5))	0,005 (M/PH/1: PH2 (3))	**0,350** (M/PH/1: PH2 (3))	0,947 (PH/M/1: PH2 (3))	**0,041** (PH/PH/1: ME (3–4))	0,664 (PH/M/1: PH2 (5))
PH/PH/1 moments 2nd [12] (name of algorithm in brackets)	0,058 (M/PH/1: PH2 (3), PH2 (5))	**0,426** (M/PH/1: PH2 (5))	1,556 (PH/M/1: PH2 (5))	**0,070** (M/PH/1: PH2 (5))	1,138 (PH/M/1: PH2 (3))	0,005 (M/PH/1: PH2 (5))	**0,350** (M/PH/1: PH2 (5))	0,947 (PH/M/1: PH2 (5))	0,049 (M/PH/1: PH2 (3), PH2 (5))	0,664 (PH/M/1: PH2 (3))

queuing systems, which give acceptable results (comparable to the best results), when "green" and "yellow" sets of λ_{INV} are considered. For "red" and "all" sets of call set-up request intensity PH/M/1 queuing systems are the best.

When links with relatively low bandwidth are used (data set 1b; Table 5) the situation is different. PH/PH/1 queuing systems give the best results for "green" and "yellow" sets of λ_{INV}. For "red" and "all" the smallest and very comparable RMSE values are offered by M/G/1 and M/PH/1 queuing systems. Due to much simpler calculations, we propose to use M/G/1.

For the above mentioned PH/PH/1 and PH/M/1 queuing systems RMSE results obtained using maximum likelihood and distance minimization algorithms operating on whole experimental histograms are in most cases not significantly better than for

algorithms based only on moments [12]. As only slightly better results do not compensate for the high computational complexity (many hours of calculations for maximum likelihood and distance minimization algorithms), for engineering purposes more applicable are moment matching algorithms.

The necessity to choose different queuing systems in the analytical traffic model for different data sets and call set-up request intensities results from the complexity of IMS/NGN elements and call scenario. In the network there are many instances of call scenarios performed simultaneously, each of them concern sending and processing a set of signaling messages (Fig. 2). During a single instance most of the network elements is used several times. Our previous research [25] indicated that in many cases even a small change in load of one IMS/NGN element results in modification of types of message inter-arrival time distributions in the whole system. Consequently, it is very difficult to select a single queuing system, which would be efficient for all sets of investigated input variables.

5 Conclusions and Future Work

In the paper we continue our work on analytical and simulation traffic model of a single domain of IMS/NGN. Our aim is to find the best queuing system for the analytical model, to achieve possibly the best conformity with experimental results. These are provided by our simulation model, which does not implement theoretical queuing systems, but reflects the operation of real IMS/NGN network elements and standardized call scenarios.

In this paper investigations using PH/PH/1 queuing systems (and their special cases – PH/M/1, M/PH/1) are performed, in which arrival and service distributions are represented by phase-type distributions obtained using maximum likelihood and distance minimization algorithms. The obtained results are compared to these calculated using previously investigated queuing systems (M/G/1; G/G/1 based on two or three moments of arrival distribution and two moments of service distribution; PH/PH/1 with phase-type distributions obtained using moment matching algorithms). The compared parameters are total system responses – mean Call Set-up Delay $E(CSD)$ as well as mean Call Disengagement Delay $E(CDD)$, standardized by the ITU-T.

The performed research demonstrates that performance of the tested queuing systems in IMS/NGN depends on several network parameters, including load of network elements and links (for links load results from message intensity and available bandwidth). Often for the assumed set of parameters more than one of the investigated queuing systems offer accuracy, which can be considered as good or even very good. However, due to specific nature of IMS/NGN elements and call scenario, it is very difficult to point one universal queuing system, appropriate for all sets of network parameters.

Apart from accuracy, in engineering applications very important is also computational complexity (the time necessary to obtain analytical results for particular queuing system), which has not been analyzed in our previous papers. Considering both accuracy and complexity, and assuming that network is not overloaded (“green” and “yellow” sets of call set-up request intensity; Sect. 4), we suggest using M/G/1 and

PH/PH/1 queuing systems in the analytical model of a single domain of IMS/NGN. The first type of queuing systems should be applied when IMS/NGN elements are connected directly or when links with relatively high bandwidth are used (data sets 1a, 1c; Table 2). The second one performs the best for links with relatively low bandwidth (data set 1b; Table 2). Due to smaller time necessary to fit phase-type distributions, for PH/PH/1 queuing systems we propose to use moment matching algorithms, which offer final results only slightly worse than maximum likelihood and distance minimization algorithms operating on whole experimental histograms.

In the next step we are going to perform similar research to that described in this paper, but in a multidomain IMS/NGN environment (appropriate analytical and simulation traffic models are under development [33–35]).

References

1. General overview of NGN, ITU-T Recommendation Y.2001, December 2004
2. IP Multimedia Subsystem (IMS); Stage 2 (Release 14), 3GPP TS 23.228 v14.3.0, March 2017
3. Call processing performance for voice service in hybrid IP networks, ITU-T Recommendation Y.1530, November 2007
4. SIP-based call processing performance, ITU-T Recommendation Y.1531, November 2007
5. Kaczmarek, S., Kaszuba, M., Sac, M.: Simulation model of IMS/NGN call processing performance. Gdańsk Univ. Technol. Faculty ETI Ann. **20**, 25–36 (2012)
6. Kaczmarek, S., Sac, M.: Traffic model for evaluation of call processing performance parameters in IMS-based NGN. In: Grzech, A., et al. (eds.) Information systems architecture and technology: networks design and analysis, pp. 85–100. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2012)
7. Kaczmarek, S., Sac, M.: Analysis of IMS/NGN call processing performance using G/G/1 queuing systems approximations. Telecommun. Rev. Telecommun. News (8–9), 702–710 (2012)
8. Osogami, T., Harchol-Balter, M.: Closed form solutions for mapping general distributions to quasi-minimal PH distributions. Perform. Evaluat. **63**(6), 524–555 (2006)
9. Bobbio, A., Horvath, A., Telek, M.: Matching three moments with minimal acyclic phase type distributions. Stochastic Models **21**(2–3), 303–326 (2005)
10. Telek, M., Horvath, G.: A minimal representation of Markov arrival processes and a moments matching method. Perform. Evaluat. **64**(9–12), 1153–1168 (2007)
11. Asmussen, S., Nerman, O., Olsson, M.: Fitting phase-type distributions via the EM algorithm. Scand. J. Stat. **23**(4), 419–441 (1996)
12. Kaczmarek, S., Sac, M.: Analysis of IMS/NGN call processing performance using phase-type distributions. In: Grzech, A., et al. (eds.) Information systems architecture and technology: network architecture and applications, pp. 23–39. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2013)
13. Kaczmarek, S., Sac, M.: Traffic modeling in IMS-based NGN networks. Gdańsk Univ. Technol. Faculty ETI Ann. **1**(9), 457–464 (2011)
14. Functional requirements and architecture of next generation networks, ITU-T Recommendation Y.2012, April 2010
15. IMS for next generation networks, ITU-T Recommendation Y.2021, September 2006
16. Resource and admission control functions in next generation networks, ITU-T Recommendation Y.2111, November 2011

17. Resource control protocol no. 1, version 3 – Protocol at the Rs interface between service control entities and the policy decision physical entity, ITU-T Recommendation Q.3301.1, August 2013
18. Pirhadi, M., Safavi Hemami, SM., Khademzadeh, A.: Resource and admission control architecture and QoS signaling scenarios in next generation networks. World Appl. Sci. J. **7** (Special Issue of Computer & IT), 87–97 (2009)
19. Fajardo, V., et al.; Diameter Base Protocol, IETF RFC 6733, October 2012
20. Rosenberg, J., et al.: SIP: Session Initiation Protocol, IETF RFC 3261, June 2002
21. OMNeT++ Discrete Event Simulator – Home. http://www.omnetpp.org. Accessed 31 May 2017
22. Thümmler, A., Buchholz, P., Telek, M.: A novel approach for fitting probability distributions to real trace data with the EM algorithm. In: Proceedings of the 6th IEEE International Conference on Dependable Systems and Networks, Yokohama, Japan, pp. 712–721 (2005)
23. Wang, J., Liu, J., She, C.: Segment-based adaptive hyper-erlang model for long-tailed network traffic approximation. J. Supercomput. **45**(3), 296–312 (2008)
24. Horvath, A., Telek, M.: PhFit: a general phase-type fitting tool. In: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools, TOOLS 2002, London, United Kingdom, pp. 82–91 (2002)
25. Kaczmarek, S., Sac, M.: Approximation of message inter-arrival and inter-departure time distributions in IMS/NGN architecture using phase-type distributions. J. Telecommun. Inform. Technol. (3), 9–18 (2013)
26. Pérez, J.F., van Velthoven, J., van Houdt, B.: Q-MAM: a tool for solving infinite queues using matrix-analytic methods. In: Proceedings of the SMCTools 2008, Athens, Greece (2008)
27. Gross, D., Shortle, J., Thompson, J., Harris, C.: Fundamentals of Queueing Theory, 4th edn. Wiley, New York (2008)
28. Lucantoni, D.M., Meier-Hellstern, K.S., Neuts, M.F.: A single-server queue with server vacations and a class of nonrenewal arrival processes. Adv. Appl. Probab. **22**(3), 676–705 (1990)
29. Lucantoni, D.M.: New results on the single server queue with a batch Markovian arrival process. Commun. Stat. Stochast. Models **7**(1), 1–46 (1991)
30. PATS: Performance analysis of telecommunication systems. http://win.ua.ac.be/∼vanhoudt/ . Accessed 31 May 2017
31. TELCOM 2825 Information systems and network infrastructure protection. http://www.sis.pitt.edu/∼dtipper/2130_notes.html. Accessed 31 May 2017
32. Abhayawardhana, V.S., Babbage, R.: A traffic model for the IP Multimedia Subsystem (IMS). In: IEEE 65th Vehicular Technology Conference, VTC2007, Dublin, Ireland (2007)
33. Kaczmarek, S., Sac, M.: Traffic model of a multidomain IMS/NGN. Telecommun. Rev. Telecommun. News (8–9), 1030–1038 (2014)
34. Kaczmarek, S., Sac, M.: Call processing performance in a multidomain IMS/NGN with asymmetric traffic. In: Grzech, A., et al. (eds.) Information Systems Architecture and Technology: Selected Aspects of Communication and Computational Systems, pp. 11–28. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2014)
35. Kaczmarek, S., Sac, M.: Verification of the analytical traffic model of a multidomain IMS/NGN using the simulation model. In: Grzech, A., et al. (eds.) Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part II. Advances in Intelligent Systems and Computing, vol. 430, pp. 109–130. Springer, Cham (2016)

Motion Repeatability of Tennis Forehand Preparation Phase Without the Ball Using Three Dimensional Data

Maria Skublewska-Paszkowska(✉)

Institute of Computer Science, Lublin University of Technology,
Nadbystrzycka 38D, 20-618 Lublin, Poland
maria.paszkowska@pollub.pl

Abstract. The forehand is one of the key shot in tennis. Its reparability is a very important aspect of gaining points. The first purpose of this paper is to create the algorithms for calculating the motion parameters for the preparation phase of the forehand, such as: the position of the head of the tennis racket relative to the player's body, the elbow angle (i.e. whether the arm is bent or straight) and the placement of the player's feet towards the front of the pelvis. The second aim of the study is to calculate the repeatability of the selected motion parameters (both for the body and the racket) of the preparation phase of the tennis forehand using coefficient of variation. The shots were performed in the motion capture laboratory using the Plug-in Gait biomechanical model. The described parameters are computed from three dimensional data, stored in C3D file.

The obtained results indicate that the placement of the head of the tennis racket towards the player's body is repeatable only for first two defined positions (above the forehead and between it and the shoulder). The participant puts his feet repeatable, after average 46% time he puts his left leg forward. The arrangement of the arm during the preparation phase of the forehand is also repeatable.

Keywords: Tennis forehand · Repeatability · Motion capture · 3D algorithms

1 Introduction

Tennis forehand is one of the basic strokes used in every game. In this shot two phases may be specified: the preparation and the moment of impact [1]. Each step is an important aspect of a properly performed stroke. The aspect of the repeatability of tennis shots plays an important role in archiving better results, both for professional and amateur players. Each time the players should be able to perform strikes with the same or similar parameters, including both the body and the racket.

The first purpose of this paper is to create the algorithms for calculating the motion parameters for the preparation phase of the forehand. They are: the position of the head of the tennis racket relative to the player's body and the placement of the player's feet towards the front of the pelvis. The elbow angle (i.e. whether the arm is bent or straight) is read from the biomechanical data. The second aim of the study is to calculate the repeatability of the selected motion parameters (both for the body and the

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_14

racket) of the preparation phase of the tennis forehand using numerical evaluation. In this research the preparation phase begins when the racket's position reaches the maximum along XY plane and ends before the moment of impact.

The shots were performed in the motion capture laboratory using the Plug-in Gait biomechanical model and a racket model [2]. The described parameters are computed from three dimensional data, stored in C3D file. The above-described parameters were computed for ten consecutive forehand strikes of a professional tennis coach.

2 Research Overview

Much research concerns analysis of the serve as the main stroke making an advance during the game. The motion repeatability of the tennis serve of three experienced and three novice players was studied in [3]. The participants were asked to serve a ball into another side of the tennis court 10 times using an inertial motion capturing system consisting of 17 sensors. The sampling rate was 40 fps. Three-dimensional position fluctuations of the lumbar vertebra were measured as the centre of the body. Three-dimensional angular velocities of 23 segments and 22 joints were captured. The tennis player's serve analysis (during the backswing preparation as well as the forward swing motions) using accelerometers is presented in [4]. The kinematic chain pattern and correct angular and translational movements of the knee, waist and wrist of the player are discussed.

The created model for the tennis serve is presented in [5]. The authors created a tool for evaluating the individual technique of a tennis player based on the kinetic energy transmission pattern using a complete mechanical body model. The model, created for the purpose of the research, consisted of 28 markers with five solid-rigid ones. The tool treated the individual technique of a player as a mathematical function. The significant discriminant function evaluated the tennis player's serve as "good" or "bad". The kinetic energy is composed of a linear velocity of a body segment and a rotational component which considers the angular velocity of the segment.

The sport technique in male tennis was studied in [6], for three shots. The analysis was done in four key points: when the time of maximum acceleration of upper tennis racket occurred, when the maximum speed of upper tennis racket was recorded, when the time of contact between the tennis racket and the ball occurred, and for the highest and lowest values of kinematic parameters of all segments of the dominant upper extremity (right). The kinematic parameters chosen for the study were: velocity, acceleration of the dominant upper extremity, centre of gravity and upper racket, height, lateral and forward-backward distance of the centre of gravity, angles in both knees, elbows and body bending.

Three aspects of the racket are analysed: its velocity, swing and orientation. Both motion capture systems (optical and markerless) and video cameras are used for the research. The motion and pose in 3 dimensions were computed from a video recording [7]. This method also allows for computing racket performance. The swing of the racket was investigated in [8]. The speed of a tennis racket (as well as that of a golf club and a baseball bat) was analysed using 3-dimensional images obtained from two cameras. Reflective tape was placed on the elbow, around the wrist and around the

racket's handle in two places. The maximum effort (both in standing and sitting positions) was calculated for each swing. The swing speed was approximately constant during the test.

Tracking a tennis racket using a markerless system (based on multiple cameras) and a visual hull is presented in [9]. Views of a tennis racket were recorded and segmented into binary images. The racket's shape was obtained. The visual hull of the racket was created using the intersection of the volume of space formed by back-projecting the silhouettes from all input views. In [10] the velocity of the tennis racket was calculated based on a monochrome recording made using a high-speed camera.

The research containing a tennis racket tracking and its analysis using Vicon system is presented in [2]. Seven retro-reflective markers were attached to it, so the racket's movement could be calculated. The racket's velocity in the consecutive frames, the orientation of the racket in both sagittal and axial planes of the racket were indicated. The racket motion was studied in [11].

This paper describes the algorithms of four parameters of the preparation phase of the forehand strike. The position of the head of the tennis racket relative to the player's body was presented in [6], where the position is specified as below or above the head. In this paper the position of the tennis racket is indicated as: above the forehead, between the forehead and the shoulder, between the shoulder and the pelvis and below the pelvis. The elbow angle was also studied in [6]. In this paper the angle is computed by the Plug-in Gait model. The setting of the feet was studied in [6]. In this paper a new algorithm is presented which indicates the feet's positions.

3 Method

3.1 Participant

One right-handed 30 year-old tennis coach was the participant in this study. He was 1.74 m tall and weighed 71 kg. He signed the ethical approval form for this research.

3.2 Motion Capture System

A passive optical motion capture system was used to track the participant and the racket while performing tennis strokes at the Laboratory of Motion Analysis and Interface Ergonomics at the Lublin University of Technology in Poland where interdisciplinary tests are performed [12]. The motion capture system consisted of: eight NIR T40S cameras operating in near infrared, two reference video Bonita cameras, a Giganet hub collecting data, a desktop computer and a set of accessories (e.g. markers, a calibration wand, double-sided tape). The system recorded the positions of the markers placed on the subject's body (each marker must be seen by at least two cameras). The equipment was supplied with Vicon's Nexus 2.0 software, used for system calibration, data recording and data processing. The movements were registered with 100 Hz.

3.3 Experimental Procedure

The participant was prepared for the experiment according to the Plug-in Gait Model [13]. Thirty-nine retroreflective markers were attached to the participant using hypoallergenic double-sided tape as specified in the model. This model allows for calculating angles, torques and forces in a subject's joints. Then, the person was measured for the purpose of creating and scaling a new subject in the Nexus software (height, weight, leg length, arm offset, knee, ankle, elbow and the thickness of both hands). The subject's calibration was performed as the next step in preparation due to the verification that the markers are visible and correctly attached.

After the warm-up, the participant was told to perform ten forehand strikes without the ball while running and avoiding a bollard placed on the floor. Because the participant was running, the strokes were more natural than hitting the ball from a standing pose. The racket was tracked during the research. Seven retro-reflective markers were attached to the racket according to the scheme presented in Fig. 1. They allow one to reconstruct the shape of the racket.

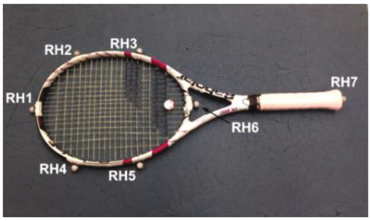


Fig. 1. Tennis racket with seven markers attached [2].

3.4 Post-processing

Each 3D recording was post-processed using the Vicon Nexus software that consisted of four main steps: marker labelling, gap filling using interpolation methods, data cleaning (e.g. deleting all unlabelled markers) and applying the Plug-in-Gait model (only for the human body). A new subject was created for the racket. It consisted of seven markers [2]. The post-processed recordings were exported as C3D files. Each C3D file contains ten consecutive forehand strikes. For the purpose of this research the preparation phases were designated. These files were used for further analysis. The files were processed by the author's own piece of software created in C++ using the Biomechanical toolkit (b-tk) and Eigen libraries.

3.5 Coefficient of Variation

The coefficient of variation – CV (see formula 1) [14] was used for computing the repeatability of tennis forehand preparation phase for selected parameters.

$$CV = \frac{\sigma}{\mu} * 100 \quad (1)$$

where σ denotes the standard deviation and μ – mean.

The parameter was considered as repeatable when CV was lower than ten.

4 Algorithms

Two algorithms calculating the selected motion parameters were implemented for the purpose of this research. They base on three dimensional positions of the markers attached to the body and the racket stored in the C3D file. The first indicates the racket's positions relative to the player's body in the consecutive frames of the recording (Fig. 2). The second describes the positions of the feet and the distance from the frontal part of the pelvis in the consecutive frames (Fig. 3). The right elbow angles (RElbowAngle) were read from the biomechanical data given by implementing the Plug-in Gait model.

```
headOfRacketPlacement(vectHeadRacqueHight, vectHeadHight,
                      vectShoulderHight, vectPelvisHeight){
for(int i ← 0; i < vectHeadRacqueHight.size(); i++){
  if(vectHeadRacqueHight(i) > vectHeadHight(i))
              racketPosition(i) ← 0;
  else if (vectHeadRacqueHight(i)>=vectShoulderHight(i))
              racketPosition(i) ← 1;
  else if (vectHeadRacqueHight(i) >= vectPelvisHeight(i))
              racketPosition(i) ← 2;
  else racketPosition(i) ← 3;
}
return → racketPosition;
}
```

Fig. 2. Algorithm indicating the position of the head of the tennis racket towards the player's body.

4.1 Head of the Tennis Racket Towards the Player's Body

The pseudocode, presented in Fig. 2, is a method that inputs the height from the ground of the specified markers in a form of the vectors so that the data for all frames can be stored. The height is expressed as the Y coordinates of the markers. The markers that were taken into consideration are: RH1, RFHD, RSHO and RASI. Each iteration of the *for* loop indicates the position of the head of the tennis racket towards parts of the body

in the current frame as an integer value, defined in Table 1. The algorithm returns an array (*racketPosition*) storing the positions of all frames of the recording.

4.2 Feet Positions

The pseudocode, presented in Fig. 3, is a method that computes the distance between the line indicated by two markers placed on the frontal pelvis (RASI and LASI) and two markers attached to the toes (RTOE and LTOE) projected on the XY plane. For the consecutive frames, using *for* loop, the line is defined based on A, B and C coefficients. Then, the distances from the right and the left toe are calculated using Formula 2. The integer value is calculating which represents which leg is placed further from the pelvis. If the value is set to one, the left leg is further from the pelvis. If the value is set to two, the right leg is further from it. Otherwise, two legs equidistant from the pelvis. The algorithm returns these values in the *dist* vector.

$$\frac{|A \cdot x + B \cdot y + C|}{\sqrt{A \cdot A + B \cdot B}} \tag{2}$$

where A, B, C stand for the line coefficients and x, y are the marker coefficients.

```
distanceFromPelvis(frameNo, rPelvis, lPelvis, lToe,
                   rToe, further, distLeft, distRight){
vector dist(frameNo);
for(int i = 0; i < frameNo; i++){
  A(i) = rPelvis(i,Y) - lPelvis(i,Y);
  B(i) = lPelvis(i,X) - rPelvis(i,X);
  C(i) = lPelvis(i,Y)*rPelvis(i,X) rPelvis(i,Y)*lPelvis (i,X);
  distLeft(i)=distanceFromLineXY(A(i), B(i), C(i), lToe(i,X),
                        lToe(i,Y));
  distRight(i) = distanceFromLineXY(A(i), B(i), C(i),
                       rToe(i,X),rToe(i,Y));
  if(distLeft(i) > distRight(i))
     dist(i) = 1;
  else if (distLeft(i) < distRight(i))
     dist(i)  = 2;
  else dist(i) = 3;
   }
```

Fig. 3. The algorithm computing the distance between the frontal pelvis and both toes and indicating which leg is placed further.

5 Results

5.1 Head of the Tennis Racket Towards the Player's Body

The participant performed ten strikes (trials). The percentage share of each racket's position towards the body (defined in Table 1) was computed using algorithm presented in Fig. 2. Two extremes (Nos. 2 and 9) were excluded from the further research due to the average time phase criterion failure (furthest values relative to the average number of grates). The obtained results are gathered in Table 2.

Table 1. The racket positions towards body parts

Value	Definition
0	Head of tennis racquet (RH1) is higher than the forehead (RFHD)
1	Head of tennis racquet (RH1) is between the forehead and the shoulder (RSHO)
2	Head of tennis racquet (RH1) is between the shoulder and the pelvis (RASI)
3	Head of tennis racquet is below the pelvis (RASI)

Table 2. The percentage of the racket positions towards the body in the preparation phase

Trial	Frame numbers	Position 0 [%]	Position 1 [%]	Position 2 [%]	Position 3 [%]
1	76	64	8	11	17
2	89	65		13	16
3	67	54	9	18	19
4	69	52	6	25	17
5	80	59	5	14	23
6	75	55	5	17	23
7	73	58	5	16	21
8	76	58	5	16	21
9	96	72	5	9	14
10	70	57	6	19	19

The mean value, standard deviation (SD) and the coefficient of variation (CV) were computed for the time of the strikes (frame numbers) as well as for the racket's three positions (for percentages values). The results are gathered in Table 3.

Table 3. The statistical values for time and the percentage of the racket positions towards the body in the preparation phase

	Frame numbers	Position 0 [%]	Position 1 [%]	Position 2 [%]	Position 3 [%]
Mean	75.00	58	6	16	19
SD	4.05	4	1	4	2
CV	5.41	6.08	21.97	23.07	10.33

5.2 Feet Positions

The feet positions are calculating using algorithm presented in Fig. 3. At the beginning stage of the preparation phase the participant placed his right foot ahead of the left one. It was calculated after what time (percentage) he reversed the feet order. The results are presented in Table 4. The statistical values are: mean – 46.05, SD – 3.67 and CV – 7.97.

Table 4. The percentage time of the left foot being ahead of the right one

Trial	Time of placing the left foot ahead of the right one [%]
1	45
3	43
4	43
5	45
6	45
7	55
8	46
10	46

5.3 Elbow Angle

The average elbow angles in the following positions indicated by the head of the tennis racket towards the player's body for preparation phase are gathered in Table 5. The statistical values are presented in Table 6.

Table 5. The average elbow's angle in the following positions of forehand preparation phase

Trial	Average elbow angle in position 0 [°]	Average elbow angle in position 1 [°]	Average elbow angle in position 2 [°]	Average elbow angle in position 3 [°]
1	91.35	63.59	58.35	56.34
3	88.41	60.58	56.02	56.30
4	81.95	55.07	45.75	46.01
5	85.89	55.96	50.13	47.62
6	84.83	56.84	50.20	50.80
7	92.17	64.41	55.94	52.77
8	85.66	54.44	48.23	47.49
10	85.23	57.77	51.87	51.60

Table 6. The statistical values for the average elbow angle in the following positions of the forehand preparation phase

	Average elbow angle in position 0 [°]	Average elbow angle in position 1 [°]	Average elbow angle in position 2 [°]	Average elbow angle in position 3 [°]
Mean	86.94	58.58	52.06	51.12
SD	3.46	3.84	4.34	3.94
CV	3.98	6.55	8.34	7.71

6 Conclusions

The paper presents the idea of calculating the repeatability of selected parameters of the preparation phase of the tennis forehand without the ball. The head of the tennis racket towards the player's body, the feet positions and the elbow angles were analysed. The first aim of this paper was both to present the algorithms computing the selected parameters of the preparation phase of the tennis forehand and to verify them on the basis of C3D data of one participant (Tables 2, 4, and 5). The second purpose was to calculate the repeatability of the obtained parameters using the coefficient of variation.

The results were grouped according to the first parameter analysed – the head of the tennis racket towards the player's body. The statistical values, presented in Table 3, clearly show that only position 0 (above forehead) and 1 (between the forehead and shoulder) are repeatable. Other positions of the racket towards the body change differently in the following trials.

The results presented in Table 4 indicate that the participant begins the preparation phase with his right foot (put) forward. After average 46% of the phase time he puts his left leg forward. This parameter is repeatable. This is the correct arrangement of the lower limb in this part of the forehand.

The arrangement of the arm during the preparation phase of the forehand also plays an important role. It should be neither totally bent nor totally straight. The results gathered in Table 6 allow to draw the conclusions that in position 0 the arm is more straight (the elbow angle is higher) than in other positions. It seems that this shot is thoroughly trained.

This study may play a very interesting aspects for both beginners and professional players. The precise values, obtained by the algorithms presented here, may give directions for the future training in order to improve the player's performance and repeatability of the tennis forehand.

Acknowledgment. The research programme entitled "*Biomechanical parameters of athletes in the individual exercises*" based on the analysis of 3D motion data and EMG, realised in the Laboratory of Motion Analysis and Interface Ergonomics was approved by the Commission for Research Ethics, No. 2/2016 dated 8.04.2016. The authors would like to thank the Student Sports Club Tennis Academy POL-SART for their support.

References

1. Ivančević, T., Jovanović, B., Đukić, M., Marković, S., Đukić, N.: Biomechanical analysis of shots and ball motion in tennis and the analogy with handball throws. Facta Univ. Ser. Phys. Educ. Sport **6**(1), 51–66 (2008)
2. Skublewska-Paszkowska, M., Lukasik, E., Smolka, J.: Algorithms for tennis racket analysis based on motion data. Adv. Sci. Technol. Res. J. **10**(31), 255–262 (2016)
3. Kato, M., Sato, T., Shimodaira, Y.: Developing of evaluation method of motion repeatability by motion capturing. In: Kommers, P., Isaías, P., Fernandez Betancort, H. (eds.) International Conferences Interfaces and Human Computer Interaction 2015, Game and Entertainment Technologies 2015 and Computer Graphics, Visualization, Computer Vision and Image Processing, CGVCVIP Proceedings, pp. 325–327, Lisbon (2015)
4. Ahmadi, A., Rowlands, D.D., James, D.A.: Investigating the translational and rotational motion of the swing using accelerometers for athlete skill assessment. In: IEEE sensors 2006, EXCO, Daegu, 22–25 October 2006, pp. 980–983 (2006)
5. de López Subijana, C., Navarro, E.: Kinetic Energy transfer during the serve. J. Hum. Sport Exerc. **4**(2), 114–128 (2009). ISSN 1988-5202
6. Mudra, P., Psalman, V.: Evaluation of sport technique in tennis based on 3D kinematic analysis. In: Proceedings of the 10th International Conference on Kinanthropology: Sport and Quality of Life, pp. 406–414 (2016)
7. Elliott, N., Choppin, S., Goodwill, S.R., Allen, T.: Markerless tracking of tennis racket motion using a camera. Procedia Eng. **72**, 344–349 (2014). The 2014 Conference of the International Sports Engineering Association. Elsevier
8. Cross, R., Bower, R.: Effects of swing-weight on swing speed and racket power. J. Sports Sci. **24**(1), 23–30 (2006)
9. Elliott, N., Choppin, S., Goodwill, S.R., Allen, T.: Image-based visual hull of a tennis racket. Procedia Eng. **34**, 877 (2012). 9th Conference of the International Sports Engineering Association (ISEA)
10. Washida, Y., Elliott, N., Allen, T.: Measurement of main strings movement and its effect on tennis ball spin. Procedia Eng. **72**, 557–562 (2014). The 2014 Conference of the International Sports Engineering Association
11. Mitchell, S.R., Jones, R., King, M.: Head speed vs. racket inertia in the tennis serve. Sport Eng. **3**, 99–110 (2000). doi:10.1046/j/1460-2687.2000/00051.x
12. Skublewska-Paszkowska, M., Lukasik, E., Smolka, J., Milosz, M., Plechawska-Wojcik, M., Borys, M., Dzienkowski, M.: Comprehensive measurements of human motion parameters in research projects. In: Candel Torres, I., Gomez Chova, L., Lopez Martinez, A. (eds.) 10th International Technology, Education and Development Conference INTED, Conference Proceedings, 6–9 March 2016, Valencia, pp. 8597–8605. IATED Academy (2016)
13. Plug-in Gait Model. www.irc-web.co.jp/vicon_web/news_bn/PIGManualver1.pdf. Accessed 15 May 2017
14. Reed, J.F., Lynn, F., Meade, B.D.: Use of coefficient of variation in assessing variability of quantitative assays. Clin. Diagn. Lab. Immunol. **9**(6), 1235–1239 (2002). doi:10.1128/CDLI.9.6.1235-1239.2002

Towards Standardized Mizar Environments

Adam Naumowicz(✉)

Institute of Informatics, University of Bialystok, Bialystok, Poland
adamn@math.uwb.edu.pl

Abstract. The effectiveness of formalizing substantial parts of mathematics largely depends on the availability of relevant background knowledge. The bigger the knowledge library, however, the harder it is to specify what is or should be relevant. Even with today's size of the libraries available for various proof assistants, importing the whole library is not an option for practical performance reasons. On the other hand, too detailed import mechanisms are prone to dependency problems and pose certain difficulty for the user. In this paper we present the key ideas of a project aimed at generating standardized formalization environments which could be used to facilitate developing new formalizations based on the current content of the Mizar Mathematical Library. Showing the results of this research based on the library designed with great focus on reusability can also be insightful for designers of other formalization systems and libraries.

Keywords: Computerized proof assistants · Mizar Mathematical Library · Standardized formal environments

1 Introduction

Accumulating and reusing all previously formalized data in a form of a standard library is a necessity for today's proof assistant systems involved in large formalization projects. The way in which a given system allows one to import available data from its library varies across proof assistants with different foundations and implementations. But a common feature seems to be a file-based library items organization. For example, HOL Light has the import handled by its metalanguage OCaml's **`#use`** and the **`needs`** directive [1], there is the **`Require [Import|Export]`** in Coq [2], or the **`imports`** directive in Isabelle/Isar [3] and so on. In the current Mizar [4, 5] system[1] there are several importing directives with different roles and semantics. Especially for users of other systems, the original Mizar importing method based on a set of rather fine-grained import directives might seem cumbersome and unnecessarily complicated. This is the main motivation for the extension of the standard Mizar utilities and a corresponding language addition presented in this paper. However, the proposed solution is not meant to completely replace the current importing methods, but rather help the users to easily start writing Mizar formalizations without too much problems with setting-up a working environment containing required items from the standard library.

[1] http://mizar.org.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_15

Let us observe that the original Mizar importing mechanism comes from the times of rather scarce computer resources (RAM, storage, computational power), when it was designed as a practical solution to cope with these limitations. However, despite its intricacy, it still works reasonably well nowadays when the complexity of the formalized developments grows. Maintaining the collection of interdependent articles forming the Mizar Mathematical Library (MML)[2] based on these fine-grained dependencies is one of the factors that most of the current complex formalization developments are still processed in seconds or minutes rather than hours, which is sometimes the case with other proof environments. Note also, that even more fine-grained access to every particular definition and proof can sometimes be desirable, as shown by the **mizar-items** project [6] aimed at reconstructing the prerequisites of theorems in the spirit of reverse mathematics.

It should be underlined that the problems with altering the dependencies of articles from the MML, which we describe in the next sections, are not only Mizar-specific. Similar issues will inevitably appear in any library with a similarly rich dependence structure. Currently, the level of reusability in other large formalization libraries is significantly smaller than in the Mizar library. For example, the analysis of the Isabelle-based Archive of Formal Proofs [7] revealed that 106 out of its 215 articles were isolated nodes of the imports graph, i.e. they were not related to other articles in the library. Moreover, among the articles that had some non-trivial dependence, the maximal number of reused articles in one article was equal to 4, and the maximal number of articles directly depending on some other article reached 9. As far as Mizar is concerned, the graph of 1289 articles of the current MML version[3] has only one isolated node (for technical reasons preserved in the library, (article **SCHEMS_1**, cf. [8]), there is an article which imports data from 121 other articles (**JORDAN**, cf. [9]) and the elementary properties of subsets from **SUBSET** (cf. [10]) are imported in as many as 1250 articles.

2 Mizar Library Importing Specifics

Let us briefly recall here that an environment of a typical Mizar article contains a series of directives importing various sorts of notions from articles previously available in the MML (or a local data base). Basic information about each directive can be found in the Mizar user's manual [11].

The directives are **vocabularies**, **notations**, **constructors**, **registrations**, **requirements**, **definitions**, **equalities**, **expansions**, **theorems** and **schemes**. Apart from **requirements** and **vocabularies**, each directive contains a list of (usually a couple of dozens) article names whose data should be imported to give a certain meaning to the newly developed formalization. For a detailed explanation of the functional and implementational aspects of the more

[2] The processing and analysis of the Mizar library has been performed using the infrastructure of the University of Bialystok High Performance Computing Center.

[3] MML ver. 5.41.1289 distributed with the compatible Mizar system ver. 8.1.06, http://mizar.uwb.edu.pl/system/#download.

advanced directives **registrations**, **reductions**, **equalities** and **expansions**, the reader may consult [12–14], and [15], respectively.

The **requirements** directive accepts as its arguments specific names which do not directly correspond to any formalization article, but instead provide means to switch on special processing for selected highly-used notions, e.g. **BOOLE** for automating Boolean operations on sets [16], or **ARITHM** for automating the arithmetic of real and complex numbers [17].

Originally, the **vocabularies** directive was a means for introducing symbols in special files not associated with any particular Mizar article, so that the same symbols can later be shared by multiple articles and freely overloaded by various notations. However, today's practice of the Library Committee maintaining the MML is to name these files in accordance with the name of the first article introducing a given symbol. One might obviously come to the idea that this whole set of symbols should be represented as one common resource. However, when we made such an experiment, then it was shown immediately that some of the symbols which the authors are perfectly allowed to use in specific contexts, can cause syntactic problems in others. A simple example is the use of a symbol like **[x]** for denoting a product of objects in a category [18]:

```
definition
  let C be non void non empty ProdCatStr;
  let a,b be Object of C;
  func a [x] b -> Object of C equals
  (the CatProd of C).(a,b);
  ...
end;
```

and this obviously would cause a syntactic error with passing scheme arguments like e.g. in the very formulation of the separation scheme [19]:

```
scheme
  Separation { A()-> set, P[object] } :
  ex X being set st for x holds x in X iff x in A() & P[x]
...
end;
```

The **vocabularies** directive is quite different from other directives that syntactically look very similar. The point is that this directive allows to share the same symbols by many (sometimes) very different definitions, and so importing all their respective data when we just need to use a certain symbol would be completely unreasonable. On the other hand, we do not want the lexical analysis lose its flexibility by requiring that all symbols introduced in some article become at that point reserved for all further

library uses. With 8863 symbols in current use (1933 attributes, 4825 functors, 37 left brackets, 37 right brackets, 936 modes, 752 predicates, 175 selectors, and 168 structures)[4] syntactic clashes would be practically unavoidable.

The rest of the directives, at least from a syntactic point of view, behave in a similar way, so it is reasonable to combine them into one common **`import`** directive, which should work as a macro merging 'behind the scene' several different directives. This functionality has been implemented to reflect the situation that some resources with the common name might not necessarily exist in the library. One should be able to import all the possible items from an article even if some of the imported notions were not exported to the library (e.g. most articles contain theorems, but many do not provide schemes, or new notations, registrations, etc.).

Table 1 below shows some statistics of how the directives are used in the current MML articles.

Table 1. Average number of directives per article.

Directive	Avg. number of article names
`Constructors`	12.53760
`Definitions`	3.41117
`Equalities`	3.84251
`Expansions`	2.77502
`Notations`	26.87900
`Registrations`	15.49810
`Requirements`	4.05275
`Schemes`	2.56943
`Theorems`	24.95810
`Vocabularies`	29.31650

It should also be noted that for some of the Mizar environment directives the order in which their arguments appear in the article may be relevant, because of the concrete implementation limitations, e.g. registrations of adjectives and reductions or **`equalities`**. For **`notations`** and **`definitions`** the ordering is meaningful by design.

Preserving the ordering is useful to avoid errors caused by the overloading of popular symbols heavily used in the library. However, there might be cases of overly complicated environments that the current state of the library may not allow to apply this normalization to.

[4] According to the statistics provided by MML Query, http://fm.uwb.edu.pl/mmlquery/fillin.php?filledfilename=statistics.mqt&argument=number+1&version=5.40.1289.

In each MML version, such a reference list is available in the **mml.lar** file, which lists all the processable MML articles, starting from the axiomatic **TARSKI**. So, users are generally encouraged to construct their environments in accordance with the ordering given by this list, if it can be done without difficulty.

3 Generating Standardized Environments

The proposed improvement of the user interface for accessing the library is based on introducing a new **imports** directive to the Mizar language and implementing it in the relevant software (the ACCOMMODATOR, cf. [11]). Although the new directive is primarily devised to be useful when creating new formalizations from scratch, we should also be able to re-introduce it to the available MML articles to anticipate potential problems that may result from its use and propose ready solutions. Such standardized environments for MML articles can be prepared in two simple steps with the help of a set of PERL scripts which are distributed with Mizar [20] to facilitate the analysis of MML articles' environments and their transformations.

Firstly, we can replace all the underlying directives' names with the new **import** keyword and then use an adjusted version of the **sortenv.pl** script, to turn them into one **import** directive without any repetitions among its arguments. The sorting obviously helps to keep the article's new combined environment in sync with the natural ordering of how the MML has been built from the axiomatic notions provided by **mml.lar**. Comparing to the numbers presented in Table 1, we note that the new **imports** directives in MML articles have on average 38.5392 files as their arguments.

4 Examples

It turns out that if we automatically introduce the new **imports** directive to the current MML, 560 out of 1289 articles (43%) require some changes (in the environment ordering or adjusting the text proper part). We may, for example, look at selected files from the **YELLOW** series of articles that contain the introductory material formalized during the project of encoding in Mizar a handbook of continuous lattices (CCL) [21]. These articles are more or less in the middle of the **mml.lar** list, so they are quite advanced. On the other hand, they were developed to fill gaps from several theories (topology, lattices and ordered sets), so they can simulate quite well a typical formalization.

The first article from this series which does not proof check without errors after automatically generating its environment with the new **import** directive is **YELLOW_3** [22]. The original environment declaration in the current MML looks like this:

```
environ
 vocabularies XBOOLE_0, SUBSET_1, TARSKI, ORDERS_2, WAYBEL_0, XXREAL_0,
      ZFMISC_1, RELAT_1, MCART_1, LATTICE3, RELAT_2, LATTICES, YELLOW_0,
      EQREL_1, REWRITE1, ORDINAL2, FUNCT_1, STRUCT_0, YELLOW_3;
 notations TARSKI, XBOOLE_0, ZFMISC_1, XTUPLE_0, SUBSET_1, RELAT_1, RELAT_2,
      RELSET_1, MCART_1, DOMAIN_1, FUNCT_2, BINOP_1, STRUCT_0, ORDERS_2,
      LATTICE3, YELLOW_0, WAYBEL_0;
 constructors DOMAIN_1, LATTICE3, ORDERS_3, WAYBEL_0, RELSET_1, XTUPLE_0;
 registrations XBOOLE_0, SUBSET_1, RELSET_1, STRUCT_0, LATTICE3, YELLOW_0,
      ORDERS_2, WAYBEL_0, RELAT_1, XTUPLE_0;
 requirements SUBSET, BOOLE;
 definitions LATTICE3, RELAT_2, TARSKI, WAYBEL_0, ORDERS_2;
 expansions LATTICE3, RELAT_2, WAYBEL_0, ORDERS_2;
 theorems FUNCT_1, FUNCT_2, FUNCT_5, LATTICE3, MCART_1, ORDERS_2, RELAT_1,
      RELAT_2, RELSET_1, TARSKI, WAYBEL_0, YELLOW_0, YELLOW_2, ZFMISC_1,
      XBOOLE_0, BINOP_1, XTUPLE_0;
 schemes FUNCT_7, RELAT_1;
```

and the standardized version looks as follows (please mind that the directives **vocabularies** and **requirements** stayed intact):

```
environ
 vocabularies XBOOLE_0, SUBSET_1, TARSKI, ORDERS_2, WAYBEL_0, XXREAL_0,
      ZFMISC_1, RELAT_1, MCART_1, LATTICE3, RELAT_2, LATTICES, YELLOW_0,
      EQREL_1, REWRITE1, ORDINAL2, FUNCT_1, STRUCT_0, YELLOW_3;
 requirements SUBSET, BOOLE;
 imports RELAT_1, TARSKI, XBOOLE_0, XTUPLE_0, ZFMISC_1, SUBSET_1, FUNCT_1,
      RELAT_2, RELSET_1,  MCART_1, FUNCT_2, BINOP_1, DOMAIN_1, FUNCT_5, FUNCT_7,
      STRUCT_0, LATTICE3, YELLOW_0, ORDERS_2, ORDERS_3, WAYBEL_0, YELLOW_2;
```

Several errors reported in this article with the new environment are caused by the reorganized set of imported definitions. As a solution it is enough to move the **RELAT_1** article name before **TARSKI**, so that inclusion between two relations can be proved according to the original definition for simple sets rather than using a more specialized condition in the case of relations that comes from the redefinition in **RELAT_1**. Similarly, we need to swap the order of **ORDERS_2** and **YELLOW_0**, because the author of this formalization again preferred to 'unfold' the original definition of an antisymmetric relation. In these cases the relative distance between these two files in the list was rather small, so nothing was broken by this tiny disorder. However, trying to use the same approach to solve another definition-related error would require moving **RELSET_1** before **TARSKI** in the **imports** directive and this change would have worse consequences. Namely, the ANALYZER module reports an unknown functor, because the order of notations was changed at the same time, and now the **.:** function application symbol would not generate proper type information

(a redefinition of the form **`redefine func R.: A -> Subset of Y;`** gets overloaded and then unavailable):

```
      thus f.(x "/\" y) = f.inf {x,y} by YELLOW_0:40
        .= inf (f.:{x,y}) by A3,A2
::>         *103
        .= inf {f.x,f.y} by A1,FUNCT_1:60
        .= f.x "/\" f.y by YELLOW_0:40;
::> 103: Unknown functor
```

So a better solution for this kind of situation is to fix the proof which causes the proof checker to complain and make use of a better suited redefinition. In this case, it is indeed a better and shorter proof:

```
theorem Th1:
  for X, Y being set, D being Subset of [:X,Y:] holds D c= [:proj1 D, proj2 D:]
proof
  let X, Y be set, D be Subset of [:X,Y:];
  let x be Element of X, y be Element of Y;
  assume
A1: [x,y] in D;
  x in proj1 D & y in proj2 D by A1,XTUPLE_0:def 12,def 13;
  hence thesis by ZFMISC_1:def 2;
end;
```

instead of the original (here commented out with the error mark **`*52`** indicating a wrong definition order):

```
:: theorem Th1:
::   for X, Y being set, D being Subset of [:X,Y:] holds D c= [:proj1 D, proj2 D:]
:: proof
::   let X, Y be set, D be Subset of [:X,Y:];
::   let q be object;
::   assume
:: ::>    *52
:: A1: q in D;
::   then consider x, y being object such that
::   x in X and
::   y in Y and
:: A2: q = [x,y] by ZFMISC_1:def 2;
::   x in proj1 D & y in proj2 D by A1,A2,XTUPLE_0:def 12,def 13;
::   hence thesis by A2,ZFMISC_1:def 2;
:: end;
::> 52: Invalid assumption
```

The article **`YELLOW_9`** [23] with a new automatically generated environment contains a few analogous errors, but can also be used to illustrate a more interesting

kind of error which can happen when we merge importing directives and set a common ordering for all of them. In this case we have a space which is both topological, but also relational. The meaning of the attribute 'discrete' is different from the original and so we get:

```
registration
  cluster strict complete 1-element for TopLattice;
  existence
  proof
    take the strict reflexive 1-element discrete finite TopRelStr;
::>                                                  *136
    thus thesis;
  end;
end;
::> 136: Non registered cluster
```

as well as the following syntactically correct statement, but about a different notion of 'discrete', so the final proof step is not accepted by the CHECKER:

```
registration
  let R be RelStr;
  cluster correct discrete strict for TopAugmentation of R;
  existence
  proof reconsider BB = bool the carrier of R
    as Subset-Family of R;
    set T = TopRelStr(#the carrier of R, the InternalRel of R, BB#);
    the RelStr of R = the RelStr of T;
    then reconsider T as TopAugmentation of R by Def4;
    take T;
    T is discrete TopStruct by TDLAT_3:def 1;
    hence thesis;
::>            *4
  end;
end;
::> 4: This inference is not accepted
```

The solution, of course, is to put **TD_LAT** after **ORDERS_3** in the **imports** in order to reflect the author's original order of notations and so instead of importing the notion for relational structures:

```
definition
  let IT be RelStr;
  attr IT is discrete means
:: ORDERS_3:def 1
  the InternalRel of IT = id (the carrier of IT);
end;
```

force the system to use the version defined in topological terms for arbitrary topological structures:

```
definition
  let IT be TopStruct;
  attr IT is discrete means
:: TDLAT_3:def 1
  the topology of IT = bool the carrier of IT;
  attr IT is anti-discrete means
end;
```

Fortunately, the reordering of **TD_LAT** and **ORDERS_3** does not introduce other troublesome imports.

5 Concluding Remarks

The experimental versions of the Mizar ACCOMMODATOR (**accom** and **makeenv**) binaries precompiled for the main distribution platforms (Linux, Windows and MacOSX/Darwin), the adjusted **sortenv.pl** environment sorting script and relevant example Mizar articles can be found at the Mizar website: http://mizar.uwb.edu.pl/~softadm/imports/. Apart from the implementation related to adding a new importing directive, in order to accommodate successfully all the articles from the current MML some limits fixed in the current systems had to be increased, e.g. the **MaxAttrPattNbr** restricting the number of permittable attribute patterns. The new word **imports** should also be added to the list of reserved words of the language contained in the library file **mizar.dct**.

The new mechanism is inevitably going to produce some performance issues from time to time when various automation mechanisms would be overused indirectly by the users. It is, however, expected that the new mechanism can significantly help the users, especially the less experienced ones. And when their work is submitted to the MML, the best optimized environment could be restored by the Library Committee and preserved in the library.

It is also worthwhile to consider developing even more high-level environment importing directives. They can resemble current **requirements** by using a fixed name for a combination of carefully selected imports that enable jump start developments in specific theories, like the aforementioned continuous lattices theory, for instance.

The presented experiment sheds some light on how the granularity of import mechanisms could be further developed in Mizar to better serve the users. It also shows the users that it is beneficial to organize the file structure of their newly formalized theories in a more fine-grained way to avoid import conflicts during the reuse of their data. On the other hand, our research carried out over the Mizar library allows to anticipate similar potential problems with reorganizing the rich dependence structure of other formal mathematical libraries.

References

1. Harrison, J.: HOL Light tutorial. http://www.cl.cam.ac.uk/∼jrh13/hol-light/tutorial.pdf. Accessed 18 May 2017
2. Bertot, Y., Castéran, P.: Interactive theorem proving and program development - Coq'Art: the calculus of inductive constructions. In: Texts in Theoretical Computer Science. An EATCS Series. Springer (2004)
3. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, vol. 2283. Springer, New York (2002)
4. Grabowski, A., Korniłowicz, A., Naumowicz, A.: Four decades of Mizar. J. Autom. Reason. **55**(3), 191–198 (2015)
5. Bancerek, G., Byliński, C., Grabowski, A., Korniłowicz, A., Matuszewski, R., Naumowicz, A., Pąk, K., Urban, J.: Mizar: state-of-the-art and beyond. In: [24], pp. 261–279 (2015)
6. Alama, J.: Mizar-items: exploring fine-grained dependencies in the Mizar Mathematical Library. In: Davenport, J.H., Farmer, W.M., Urban, J., Rabe, F. (eds.) Intelligent Computer Mathematics - 18th Symposium, Calculemus 2011, and Proceedings of 10th International Conference, MKM 2011, Bertinoro, Italy, 18–23 July 2011. Lecture Notes in Computer Science, vol. 6824, pp. 276–277. Springer (2011)
7. Blanchette, J.C., Haslbeck, M., Matichuk, D., Nipkow, T.: Mining the Archive of formal proofs. In: [24], pp. 3–17 (2015)
8. Czuba, S.T.: Schemes. Formaliz. Math. **2**(3), 385–391 (1991)
9. Korniłowicz, A.: Jordan curve theorem. Formaliz. Math. **13**(4), 481–491 (2005)
10. Trybulec, Z.: Properties of subsets. Formaliz. Math. **1**(1), 67–71 (1990)
11. Grabowski, A., Korniłowicz, A., Naumowicz, A.: Mizar in a nutshell. J. Formaliz. Reason. **3** (2), 153–245 (2010)
12. Naumowicz, A.: Enhanced processing of adjectives in Mizar. Stud. Log. Gramm. Rhetor. **18** (31), 89–101 (2009)
13. Korniłowicz, A.: On rewriting rules in Mizar. J. Autom. Reason. **50**(2), 203–210 (2013)
14. Grabowski, A., Korniłowicz, A., Schwarzweller, C.: Equality in computer proof-assistants. In: 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Łódź, Poland, 13–16 September 2015, pp. 45–54 (2015)
15. Korniłowicz, A.: Definitional expansions in Mizar. J. Autom. Reason. **55**(3), 257–268 (2015)
16. Naumowicz, A.: Automating Boolean set operations in Mizar proof checking with the aid of an external SAT solver. J. Autom. Reason. **55**(3), 285–294 (2015)
17. Naumowicz, A.: Interfacing external CA systems for Gröbner bases computation in Mizar proof checking. Int. J. Comput. Math. **87**(1), 1–11 (2010)
18. Byliński, C.: Cartesian categories. Formaliz. Math. **3**(2), 161–169 (1992)
19. Trybulec, Z., Święczkowska, H.: Boolean properties of sets. Formaliz. Math. **1**(1), 17–23 (1990)
20. Naumowicz, A.: Tools for MML environment analysis. In: [24], pp. 348–352 (2015)
21. Bancerek, G., Rudnicki, P.: A compendium of continuous lattices in Mizar. J. Autom. Reason. **29**(3–4), 189–224 (2002)
22. Korniłowicz, A.: Cartesian products of relations and relational structures. Formaliz. Math. **6** (1), 145–152 (1997)
23. Bancerek, G.: Bases and refinements of topologies. Formaliz. Math. **7**(1), 35–43 (1998)
24. Kerber, M., Carette, J., Kaliszyk, C., Rabe, F., Sorge, V. (eds.): Intelligent Computer Mathematics - International Conference, CICM 2015, Proceedings. Washington, DC, USA, 13–17 July 2015. Lecture Notes in Computer Science, vol. 9150. Springer (2015)

Formalization of the Nominative Algorithmic Algebra in Mizar

Artur Korniłowicz[1(✉)], Andrii Kryvolap[2], Mykola Nikitchenko[2], and Ievgen Ivanov[2]

[1] Institute of Informatics, University of Białystok, Ciołkowskiego 1M, 15-245 Białystok, Poland
arturk@math.uwb.edu.pl

[2] Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Street, Kiev 01601, Ukraine
krivolapa@gmail.com, nikitchenko@unicyb.kiev.ua, ivanov.eugen@gmail.com

Abstract. We describe a formalization of the nominative algorithmic algebra in the Mizar proof assistant. This algebra is a generalization of Glushkov algorithmic algebras which is well suited for representing semantics and specifying properties of programs on complex data structures which interact with external environment in convenient and unified way. We describe formal definitions and formally proven and checked statements about the carrier sets and operations (program compositions) of this algebra. The obtained results are useful for implementing methods of automated verification of software, for example, for the Internet of Things (IoT) and ensuring safety of IoT systems.

Keywords: Formal methods · Proof assistant · Semantics · Algorithmic algebras · Formal verification

1 Introduction

Nowadays computer-controlled systems are becoming widely deployed and the idea of the Internet of Things [1], of connecting a variety of physical devices and virtual entities into a global network is turning into reality. Since embedded software plays an important role in such products, controls their behavior, and, in the case of incorrect implementation, can be the primary source of safety issues, it is highly desirable to incorporate at least some software verification processes into IoT device design practice to reduce the amount of such issues and their implications.

Generally, verification means checking that a system behaves in accordance with its specification. Formal verification uses mathematical methods for this purpose, usually relying on mathematical logic and/or type theory, and is considered to be the approach of choice for obtaining the strongest safety guarantees. However, in order to apply formal verification, a system designer needs to formulate a formal, mathematically rigorous specification of the intended behavior of the system and obtain an adequate mathematical model of its implementation. Then formal verification may be attempted using the relevant automated or semi-automated software tools [2] – automated

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_16

theorem provers, model checkers, proof assistants, etc., or manually. If formal verification succeeds, it provides a strong evidence that the implemented system has the safety properties implied by its specification.

Verification methods have been studied for a long time, and many of them were based on the Floyd–Hoare approach. In particular, the means of expressing formal specifications commonly included pre- and post-conditions. Such types of specifications also appeared in the domains of artificial intelligence and robotics, e.g. in action description languages such as STRIPS, PDDL, etc.

In recent years there has been a significant progress in tools that support application of formal methods to software development, some of which include

- *verified compilers* (CompCert [3] for C, CakeML [4] for ML, etc.) that generate assembly or machine code that is proven to be equivalent to the source code under the assumptions of the formalized source language semantics and realistic models of microprocessor instruction set architectures (ISAs) such as ARM, Intel x86, etc.;
- *microkernels* and *hypervisors* (e.g. seL4 [5], CertiKOS [6], etc.) that are formally verified with respect to high-level formalized specifications of system interface and behavior under the assumptions of realistic formal models of ISAs.

Usage of such tools in embedded software development for IoT allows one to eliminate some potential sources of safety issues (errors during type checking of the source code or translation of the source code to executable machine code, errors in implementation of scheduling, and, to some extent, resource management and access control). However, the following issues remain:

- *correctness of application-level IoT software* is not implied by the correctness of implementation of a compiler and a limited operating system kernel; application-level software still requires a certain form of verification;
- *robustness* of both formally verified system and application-level software components in the case of deviation of hardware and/or external software from their assumed behavior or hardware failures is usually not analyzed and formal proofs of software correctness (e.g. seL4 microkernel correctness proof using Isabelle/HOL proof assistant [7]) usually do not give usable information on such situations.

As a step towards development of a methodology and tools that allow solving such issues, in the previous works [8, 9] we proposed a logical framework for specifying properties of imperative programs, formalizing imperative programs, and reasoning about their behavior, based on ideas from the *composition-nominative approach to program formalization* [10], *Glushkov algorithmic algebras* [11], and the *Floyd–Hoare logic* [12, 13]. In particular, in these works we constructed a generalization of the Floyd–Hoare logic for partial pre- and post-conditions [8] and a special inference system for it [8, 9]. The approach can be supported by the following observations:

- many software verification projects, including the mentioned above, use extensions of the classical Floyd–Hoare logic to reason about program behavior, so it can be expected that a suitable extension of the Floyd–Hoare logic implemented in semi-automatic or automatic verification tools may provide a satisfactory basis for

formal verification of IoT software and also give information about robustness of the verified code in the case of failure of assumptions under which it was verified;
- application-level IoT software usually deals with sufficiently high-level communication with external IoT devices or web services on the Internet (and this agrees the idea of the Internet of Things in general), and many security and safety considerations for IoT devices are related to the questions of correctness of implementation of the corresponding communication protocols and, especially, correctness of the procedures that interpret and process the data that is sent and received;
- data formats commonly used in such communication, especially with web services, including JSON, XML and other related formats, have tree-like, hierarchical nature, so it is important to be able to specify in the convenient form the behavior and perform formal verification of code that processes hierarchically organized data;
- the cases of deviation of hardware and/or external software components from their assumed behavior, hardware failures, etc. can be represented as *undefinedness* of parts (predicates, operations, data components, etc.) of the formal specification of the preconditions of execution of software code when its behavior is analyzed in the context of such deviations/failures/phenomena.
- one can analyze how robust a verified software code is in cases of different types of deviations of its execution environment from the behavior assumed in its formal specification by analyzing what happens when different parts of preconditions of execution of its instructions become undefined; undefinedness of certain parts of preconditions of code execution may or may not (depending on how a deviation event affects code execution) lead to nondeterminism of code operation.

As a solution, we proposed an extension of the Floyd–Hoare logic which we described in [8] which allows one to

- reason about programs which operate on complex, tree-like data structures in convenient and high-level way;
- include partial predicates [14] in program pre- and postconditions.

This logic allows one to verify assertions (Hoare triples) of the form

$$\{Pre{-}condition\}Program\{Post{-}condition\}$$

which mean that if the *Pre-condition* holds (a condition on the state of the memory) immediately before the *Program* is run, and the *Program* terminates, then *Post-condition* holds immediately after *Program's* execution. Here *Pre-condition* and *Post-condition* can be specified by partial predicates on memory states (i.e. they are not necessarily defined on all possible data), which is one of the major differences between this logic and the classical Floyd–Hoare logic. The presence of partial pre- and post-conditions makes the inference system for the classical Floyd–Hoare logic unsound, so our logic requires a different (modified) sound inference system. We described this inference system in [8]. A *Program* in assertions is described in a language with the usual structural programming constructs of the sequential execution, branching, cycle, however, it can operate on complex hierarchically organized data structures as nominative data [10, 15] which allow convenient representation of various

commonly used data structures, like multidimensional arrays, tables, trees, etc. These representations were described in [15]. Nominative data have the form of name-value assignments in which values can be nominative data (name-value assignments) themselves:

$$[name1 \rightarrow value1, name2 \rightarrow [nested1 \rightarrow value2, nested2 \rightarrow value3], name3 \rightarrow value4].$$

Such nominative data can be interpreted as labeled oriented trees in which arcs are labeled with names (here *name1*, *name2*, *nested1*, *nested2*, *name3*), and leaf nodes contain values (here *value1*, *value2*, *value3*, *value4*). For example:

$$[x \rightarrow 1, y \rightarrow [x \rightarrow 2], z \rightarrow 2].$$

As a simple illustration consider the following program written in pseudocode, which reads (synchronously) a fragment of data from a location in an external storage—a Flash memory, denoted as *ExternalMem.Flash.page1,* into random access memory (RAM) location denoted as *RAM.region1*, performs some computation on *RAM.region1* (*computation*), and writes the result into *ExternalMem.Flash.page1:*

```
RAM.region1 := ExternalMem.Flash.page1;
RAM.region1 := computation(RAM.region1);
ExternalMem.Flash.page1 := RAM.region1;
```

Assume that the current execution state of the program is modeled as a nominative data of the form:

$$[RAM \rightarrow [region1 \rightarrow content1, region2 \rightarrow content2, ...],$$
$$ExternalMem \rightarrow$$
$$[Flash \rightarrow [page1 \rightarrow flash\_content1, page2 \rightarrow flash\_content2, ...]]],$$

where *RAM*, *region1*, *region2*, …, *ExternalMem*, *Flash*, *page1*, *page2*, … are names and *content1*, *content2*, …, *flash_content1*, *flash_content2*, … denote concrete values assigned to the corresponding names in the nominative data. Moreover, *content1*, *content2,* … mean the content currently stored in RAM, and *flash_content1*, *flash_content2* mean the content currently stored in the Flash memory. We use the dot notation like *RAM.region1* to denote the name *region1* inside the value corresponding to the name *RAM* in a nominative data (formally, the name sequences like *RAM.region1* are called *complex names* in [15], and the operation of finding the value corresponding to them in a nominative data is called *associative denaming* [15]).

During execution of the program the content of RAM and the Flash memory may change. We can model this change as transition from one nominative data to another. E.g., suppose that the initial state (before the execution of the program) is

$$[RAM \rightarrow [region1 \rightarrow 0, region2 \rightarrow 0],$$
$$ExternalMem \rightarrow [Flash \rightarrow [page1 \rightarrow 1, page2 \rightarrow 0]]].$$

Then after execution of the first instruction:

```
RAM.region1 := ExternalMem.Flash.page1;
```

the memory state can be described by the following nominative data:

$$[RAM \to [region1 \to 1, region2 \to 0], \\ ExternalMem \to [Flash \to [page1 \to 1, page2 \to 0]]].$$

This fact implies, e.g. the following assertion:

$$\{\ ExternalMem.Flash.page1 = 1\ \} \\ RAM.region1 := ExternalMem.Flash.page1; \\ \{\ RAM.region1 = 1\ \}$$

Similar assertions can be stated for other statements and can be combined using rules of the inference system to form an assertion about the whole program. The pre- and post-conditions can be considered as predicates on nominative data which represent memory state, or, more generally, the state of the execution environment of the program. Our extended Floyd–Hoare logic supports the situations when the values of these predicates are defined on some nominative data and are undefined on another nominative data. For example, suppose that we model a hardware fault in the Flash memory. This situation can be modeled as the case when the nominative data representing the memory state of system loses the name-value pair in which the name is *Flash*, so this nominative data becomes:

$$[RAM \to [region1 \to 1, region2 \to 0], ExternalMem \to [\ ]].$$

Analogously, we can model a situation when the hardware fault is related only to some, but not all page(s) of the Flash memory, e.g.

$$[RAM \to [region1 \to 1, region2 \to 0], ExternalMem \to [page2 \to 0]],$$

i.e. here nominative data loses the name-value pair corresponding to the name *page1*.

It is natural and convenient for further reasoning to assume that in this case the value of the predicate in the precondition $\{ExternalMem.Flash.page1 = 1\}$ becomes undefined. So the precondition is a partial predicate on nominative data – it is defined, when there is a name-value pair in this nominative data, corresponding to the name *page1* (located in the value corresponding to the name *Flash* which is in the value corresponding to the name *ExternalMem*), and is undefined, otherwise. When this predicate is defined, its value is true, if the value corresponding to the name *page1* is 1, and is false otherwise. Definedness of the precondition predicate and its truth value, if any, has influence on the conclusion that can be done concerning the postcondition after execution of a program statement. In our case, the first program instruction refers

to a malfunctioning device (*ExternalMem.Flash.page1*). What actually happens in this case depends on the nature of the fault, the behavior of the hardware and software environment of the program, the semantics of the language in which it is written, etc. However, conservatively, we can assume that the value of the predicate in the post-condition {*RAM.region1 = 1*} and, in fact, the program behavior, is undefined in this case. On the other hand, if the statement did not refer to a malfunctioning device (e.g. if it dealt only with *RAM.region1*), then, assuming the fault in the Flash device had no effect on the program's execution environment, we would assume that the precondition and postcondition predicates and the program's behavior would be well-defined. Thus we assume that proving assertions in the extended Floyd–Hoare logic allows one to perform a form of software verification, while analysis of situations when the post-conditions and the program behavior are defined or undefined can give useful information about robustness of the verified software in the case of deviations of the program execution environment from its normal and assumed behavior.

In order to be able to apply the extended Floyd–Hoare to formal verification of software, generally, it is necessary to implement it in a software system like an interactive proof assistant and/or an automated theorem prover. We consider that a preferable approach is to firstly formalize its rules, semantics, and formalize the proof of its soundness in one of the existing interactive proof assistants, and then develop methods for automation of its application to software verification problems. To formalize our extended Floyd–Hoare logic we chose the Mizar system [16, 18] which is a proof assistant based on first-order logic and axiomatic Tarski-Grothendieck set theory. The reasons for this decision include:

- Mizar has a rich library (Mizar Mathematical Library, MML) of already formalized mathematical definitions and theorems. We assume that it has a good potential for application in the domain of formal verification of IoT systems and software since the behavior of the physical environment of IoT systems can be described by continuous-time dynamical system models using ordinary differential equations, differential inclusions, stochastic differential equations, etc., and reasoning about such models depends on facts from various branches of mathematics formalized in MML.
- Mizar and MML have a good support for defining of and direct reasoning about partial predicates and functions without reducing them to total function using option types (like in Isabelle/HOL [7]) or other similar techniques.

Formalization of the mentioned logic in Mizar can be done in three steps:

- Formalization of the notion of nominative data and of basic operations on them. We have done this in [17].
- Formalization of the notion of a program and of constructs (we call them *compositions*) which allow one to write programs (e.g. sequential execution, branching, loop, etc.) and which can be reasoned about using our extended Floyd–Hoare logic. The set of such programs (and predicates) and constructs (compositions) forms an algebraic structure which we introduced in [15] and called the *nominative Glushkov algorithmic algebra*, or simply, *the nominative algorithmic algebra*. This structure can be considered as a generalization of Glushkov algorithmic algebras [11] which

allows specification of programs which operate on hierarchically organized (nominative) data. It has two carrier sets – a set of programs which are, semantically, partial functions which map nominative data to nominative data (which are called *binominative functions*), and a set of partial predicates on nominative data. Its operations represent constructs which allow one to build complex programs and predicates from simpler ones.
– Formalization of the rules of the inference system [8] of the extended Floyd–Hoare logic and the proof of its soundness in Mizar.

The topic of this paper is the second item of this list, i.e. formalization of the carrier sets and operations of the nominative Glushkov algorithmic algebra in Mizar. We plan to consider the third item in the future works.

2 Mizar Formalization of Nominative Glushkov Algebras

Mathematical definitions of nominative data and the nominative Glushkov algorithmic algebra can be found in [15]. We formalized them in the form of a Mizar paper entitled NOMIN.MIZ. Formalization of nominative data and operations on them in this Mizar paper are described in [17], so we omit them here. Below we describe the formal definitions of the carrier sets and the most important operations of the Glushkov algorithmic algebra. Because of space limitations we omit some of the less important and/or straightforward definitions and the text of the formal proofs of the existence and uniqueness correctness conditions in the definitions. However, we note that they were written using various grammatical constructions supported by the Mizar language which increase the computational power of the Mizar checker (e.g. registrations and term expansions [19]) and were checked for correctness using the Mizar system.

Firstly, we fix the sets of names (`V`) and values (`A`) over which we consider the set of nominative data `ND(V,A)`, the formal definition of which is described in [17].

```
reserve V,A for set;
```

The carrier sets of the nominative Glushkov algorithmic algebra are

– the set of partial predicates on `ND(V,A)` which is denoted as `Pr(V,A)`;
– the set of programs – a partial functions from `ND(V,A)` to `ND(V,A)`, also called binominative functions, which is denoted as `FPrg(V,A)`.

We defined them in Mizar as follows:

```
definition let V,A;
  func Pr(V,A) -> set equals PFuncs(ND(V,A),BOOLEAN);
  func FPrg(V,A) -> set equals PFuncs(ND(V,A),ND(V,A));
end;
```

where `PFuncs(X,Y)` denotes the set of all partial functions from `X` to `Y` [20].

Besides, we introduced Mizar modes (types) *SCPartialNominativePredicate* and *SCBinominativeFunction*, corresponding to the elements of *Pr*(*V*,*A*) and *FPrg*(*V*,*A*):

```
definition let V,A;
mode SCPartialNominativePredicate of V,A is
     PartFunc of ND(V,A),BOOLEAN;
mode SCBinominativeFunction of V,A is
     PartFunc of ND(V,A),ND(V,A);
end;
```

where PartFunc of X,Y is a partial function from X to Y [20].

Then we formalized the operations (also called compositions) of the nominative Glushkov algorithmic algebra. The list of them is given in [15]. The most important among them are the sequential composition, the (variable) assignment, branching ("if" operator), and the "while" loop compositions on programs. We formalized them as *SCcomposition*, *SCassignment*, *SCIF*, and *SCwhile* functors in Mizar:

```
definition let A,B,C be set;
  func PFcompos(A,B,C) -> Function of
       [:PFuncs(A,B),PFuncs(B,C):],PFuncs(A,C) means
  for f being PartFunc of A,B
   for g being PartFunc of B,C holds it.(f,g) = g*f;
end;
```

where [:X,Y:] denotes the Cartesian product of sets X and Y [21].

```
definition let V,A;
  func SCcomposition(V,A) -> Function of
       [:FPrg(V,A),FPrg(V,A):],FPrg(V,A) equals
  PFcompos(ND(V,A),ND(V,A),ND(V,A));
end;
```

```
definition let V,A; let v be set;
  func SCassignment(V,A,v) ->
       Function of FPrg(V,A),FPrg(V,A) means
  for f being SCBinominativeFunction of V,A holds
   dom(it.f) = dom f &
   for d being TypeSCNominativeData of V,A holds
    d in dom(it.f) implies it.f.d =
       local_overlapping(V,A,d,f.d,v);
end;
```

where each object of the type TypeSCNominativeData of V,A is an element of *ND*(*V*,*A*) and local_overlapping is one of the operations defined on nominative data [17].

```
definition let V,A;
  func SCIF(V,A) -> Function of
    [:Pr(V,A),FPrg(V,A),FPrg(V,A):],FPrg(V,A) means
  for r being SCPartialNominativePredicate of V,A
  for p,q being SCBinominativeFunction of V,A holds
   dom(it.(r,p,q)) =
   {d where d is TypeSCNominativeData of V,A:
    d in dom r & r.d = TRUE & d in dom p
    or d in dom r & r.d = FALSE & d in dom q} &
   for d being TypeSCNominativeData of V,A holds
    (d in dom r & r.d = TRUE & d in dom p
       implies it.(r,p,q).d = p.d) &
    (d in dom r & r.d = FALSE & d in dom q
       implies it.(r,p,q).d = q.d);
end;
```

```
definition let V,A;
  func SCwhile(V,A) -> Function of
       [:Pr(V,A),FPrg(V,A):],FPrg(V,A) means
 for p being SCPartialNominativePredicate of V,A
 for f being SCBinominativeFunction of V,A holds
 dom(it.(p,f))={d where d is TypeSCNominativeData of V,A:
     ex n being Nat st
      (for i being Nat st i <= n-1 holds
       d in dom(p*iter(f,i)) & (p*iter(f,i)).d = TRUE)
      & d in dom(p*iter(f,n)) & (p*iter(f,n)).d = FALSE}
   & for d being TypeSCNominativeData of V,A st
     d in dom(it.(p,f)) ex n being Nat st
      (for i being Nat st i <= n-1 holds
       d in dom(p*iter(f,i)) & (p*iter(f,i)).d = TRUE)
      & d in dom(p*iter(f,n)) & (p*iter(f,n)).d = FALSE &
     it.(p,f).d = iter(f,n).d;
end;
```

where `iter(f,n)` denotes `n`-times iteration of the function `f` [22].

We formalized all other compositions of the nominative Glushkov algorithmic algebra using a similar approach.

3 Conclusion

We have formalized the nominative Glushkov algorithmic algebra in the Mizar proof assistant. This formalization is a necessary step towards formalization and application of the extended Floyd–Hoare logic proposed in [8] which supports partial pre- and postconditions and allows reasoning about programs operating on complex data structures, which can be useful for verification of IoT software. We plan to formalize this logic using the results obtained in this paper in further works.

References

1. International Telecommunication Union: Overview of the Internet of Things. ITU-T Recommendation Y.4000/Y.2060, June 2012. http://handle.itu.int/11.1002/1000/11559
2. Wiedijk, F. (ed.): The Seventeen Provers of the World. Lecture Notes in Artificial Intelligence, vol. 3600. Springer, Heidelberg (2006)
3. The CompCert project. http://compcert.inria.fr
4. CakeML: A verified implementation of ML. https://cakeml.org
5. The seL4 Microkernel. https://sel4.systems
6. CertiKOS: an extensible architecture for building certified concurrent OS kernels. In: OSDI 2016 Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016 (2016)
7. Nipkow, T., Wenzel, M., Paulson, L.C.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic. Springer, Heidelberg (2002)
8. Kryvolap, A., Nikitchenko, M., Schreiner, W.: Extending Floyd-Hoare logic for partial pre- and postconditions. Communications in Computer and Information Science, vol. 412, pp. 355–378. Springer (2013). http://dx.doi.org/10.1007/978-3-319-03998-5_18
9. Korniłowicz, A., Kryvolap, A., Nikitchenko, M., Ivanov, I.: An approach to formalization of an extension of Floyd-Hoare logic. In: Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer (ICTERI 2017), Kyiv, Ukraine, 15–18 May 2017. CEUR-WS.org (2017). http://ceur-ws.org/Vol-1844/10000504.pdf
10. Nikitchenko, N.S.: A composition nominative approach to program semantics, IT-TR 1998-020. Technical report, Technical University of Denmark (1998)
11. Glushkov, V.: Automata theory and formal transformations of microprograms. Cybernetics **5**, 3–10 (1965). (in Russian)
12. Floyd, R.: Assigning meanings to programs. In: Mathematical Aspects of Computer Science, vol. 19, pp. 19–32 (1967)
13. Hoare, C.: An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969). doi:10.1145/363235.363259
14. Nikitchenko, M., Shkilniak, S.: Algebras and logics of partial quasiary predicates. Algebra Discret. Math. **23**(2), 263–278 (2017)
15. Skobelev, V., Nikitchenko, M., Ivanov, I.: On algebraic properties of nominative data and functions. Commun. Comput. Inf. Sci. **469**, 117–138 (2014). doi:10.1007/978-3-319-13206-8_6
16. Grabowski, A., Korniłowicz, A., Naumowicz, A.: Four decades of Mizar. J. Autom. Reason. **55**(3), 191–198 (2015). doi:10.1007/s10817-015-9345-1
17. Korniłowicz, A., Kryvolap, A., Nikitchenko, M., Ivanov, I.: Formalization of the algebra of nominative data in Mizar. In: 2017 Federated Conference on Computer Science and Information Systems (2017, accepted)
18. Bancerek, G., Byliński, C., Grabowski, A., Korniłowicz, A., Matuszewski, R., Naumowicz, A., Pąk, K., Urban, J.: Mizar: state-of-the-art and beyond. In: Intelligent Computer Mathematics – International Conference, CICM 2015, Proceedings, Washington, DC, USA, pp. 261–279 (2015). http://dx.doi.org/10.1007/978-3-319-20615-8_17
19. Grabowski, A., Korniłowicz, A., Schwarzweller, C.: Equality in computer proof assistants. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2015 Federated Conference on Computer Science and Information Systems. Annals of Computer Science and Information Systems, vol. 5, pp. 45–54. IEEE (2015). http://dx.doi.org/10.15439/2015F229

20. Byliński, C.: Partial functions. Formaliz. Math. **1**(2), 357–367 (1990)
21. Byliński, C.: Some basic properties of sets. Formaliz. Math. **1**(1), 47–53 (1990)
22. Bancerek, G., Trybulec, A.: Miscellaneous facts about functions. Formaliz. Math. **5**(4), 485–492 (1996)

Service Oriented Systems and Cloud Computing

Military and Crisis Management Decision Support Tools for Situation Awareness Development Using Sensor Data Fusion

Mariusz Chmielewski(✉), Marcin Kukiełka, Damian Frąszczak, and Dawid Bugajewski

Cybernetics Faculty, Military University of Technology, gen. Kaliskiego 2, 00-908 Warsaw, Poland
{mariusz.chmielewski, marcin.kukielka, damian.fraszczak, dawid.bugajewski}@wat.edu.pl

Abstract. This paper presents the situation awareness development method and a tool elaborated for individual soldiers and low level commanders support. Presented results are outcomes taken from the process of mCOP plaftom design and development. The system itself is an innovative proof-of-concept and a testbed software, which utilizes sensors, wireless technologies and augmented reality for supporting ground troops during various combat operations in Network Enabled Capabilities environments [14]. Choosing commercially available mobile platforms is intentional as the system should offer security features while maintaining high availability and accessibility for every soldier. The system is supported by data integration services SOA environment hosting NFFI, TSO and JC3IEDM [14, 15] military and crisis management services. Presented solution offers adjustable and configurable layered architecture for integration services and command support web portal. mCOP solution offers automatic monitoring of individuals (location, movement, biomedical signals), C4ISR [15] system's data integration and operational picture development. Utilization of mobile technologies ensures portability, accessibility and simplifies wireless communication strategies (multiple channels) providing medium for distributing tactical situation awareness. Choosing Android platform delivers also major advantages connected with transparency of OS code, availability of professional, ruggedized devices. Therefore mCOP platform can be easily distributed and deployed on available commercial smartphones and tablets.

Keywords: Situation awareness · Operational picture · Mobile system · Augmented reality · Decision support

1 Introduction and Research Applicability

The paper summarizes problems and solutions encountered during implementation of DSS software including mobile and desktop devices. Gathered experience covers issues connected with data integration, data and information fusion, service oriented

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_17

architecture and state-of-the-art visualization techniques development. Implementation of these modules delivers combination of sensor data fusion methods and AR visualization to instantly support and help soldiers during operation. The platform utilize commercially available platforms, which increases the deployment strategies.

Presented paper is a proof-of-concept architecture and implementation of prototype tools which can be used for sharing operational and tactical information, fusing data on command level and developing situation awareness of tactical level commanders and soldiers. Utilization of mobile technologies enhances the capabilities of warfare by delivering means of achieving information superiority, important especially in asymmetric threats where the inertia of the operation is very slow.

2 Battlespace Data Integration Mechanisms and Used Standards

The presented system utilizes heterogeneous data sources with implementation of data fusion methods to improve situation awareness and COP (Common Operational Picture). tCOP hosts services which produces integrated data from following data sources: NFFI (NATO Friendly Force Information), TSO (Tactical Situation Object) and JC3IEDM (Joint Consultation Command & Control Information Exchange Data Model). Besides those mentioned before, tCOP user can easily create custom situation including both APP6A and TSO nomenclature to visualize the scenario. tCOP allows also to share it with other trainees to perform exercises improving ability to act in emergency situations. NFFI [1] [is an interoperability standard to exchange friendly force tracks between national Command and Control Information Systems (C2IS) within the ISAF coalition. This standard is composed by two crucial elements: a message definition and interface for information exchange on friendly forces. TSO [1] standard defines requirements for information structure to represent situation for crisis management and is inspired by the experience of the military interoperability. It is used to transfer record of the current view of particular observer to the other one. It is mainly used for information exchange between emergency response institutions to increase situation awareness by utilized transferred data to console view or dispatcher. The TSO is crucial to coordinate and integrate quick reaction forces (QRF) by providing a minimum level of interoperability during emergency operations. The simplest way to increase the interoperability level between institutions is to use the same data model and data source. Such approach results in providing a timely and comprehensive COP. TSO defines a message structure consisting of four parts: Context, Event, Resources and Missions. Moreover, it describes both static and dynamic elements of crisis operations. Data integration layer of TSO standard in tCOP implementation is shown on Fig. 1. This feature improves situation awareness by dividing gathered data into pieces so that user can be given with optimal, not overwhelming amount of information. An example of TSO event can be: ACC/ACCTRF/HGHWAY/MOTTUN where ACC means accident, ACCTRF means an accident of the type traffic accident, HGHWAY means a traffic accident which occurred on a highway and finally MOTTUN means that an accident occurred inside a tunnel [2]. The example message written

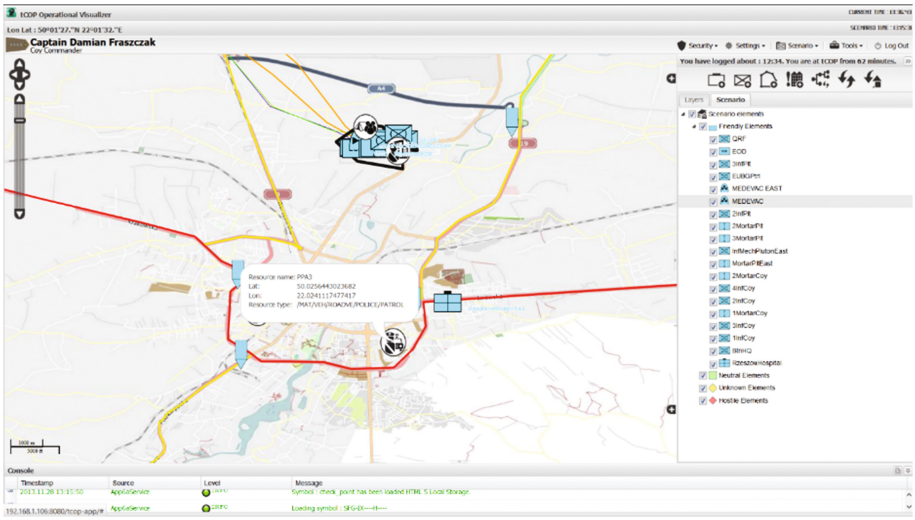


Fig. 1. Data integration of TSO and NFFI services in tCOP web client

in XML language is shown on Fig. 2 [2]: tCOP services implements several standards of data exchange.

tCOP is used to acquire and integrate the data, and generate COP product from multiple data sources to increase situation awareness both for military domain and crisis management. Each external data source is mapped to tCOP scenario structure by integration algorithms to ensure interoperability between services. Each scenario

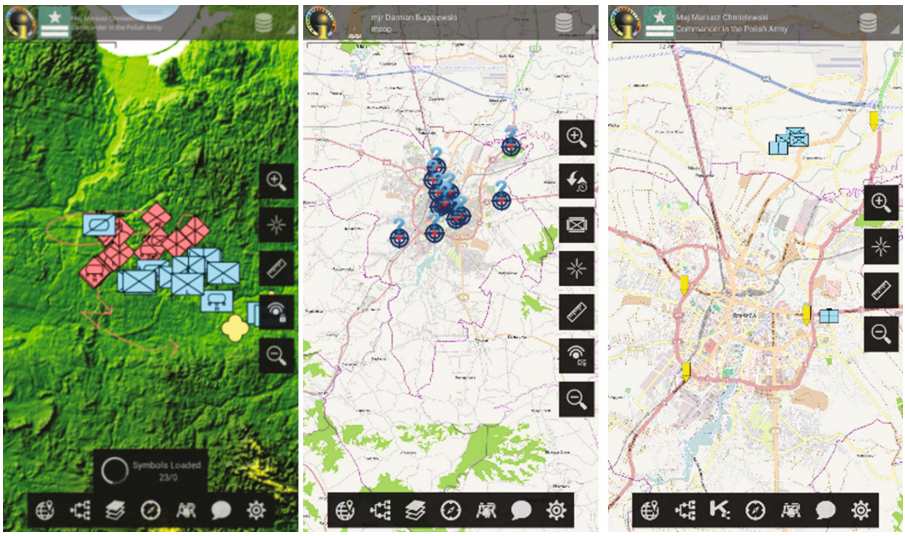


Fig. 2. (Left) Battlefield COP products, NFFI and JC3IEDM fused situation (left) estimates of critical infrastructure threat level (center), CIMIC operation fusing TSO and NFFI data

created by tCOP Scenario Creator can by translated to both NFFI and TSO standards and transferred to other clients. tCOP extends TSO and NFFI specification providing REST services which use JSON (JavaScript Object Notation). JSON data format is much simpler than XML language providing shorter representation of the transferred objects what is crucial to save limited mobile devices network transfer. tCOP extends TSO and NFFI specification providing REST services which use JSON (JavaScript Object Notation). JSON data format is much simpler than XML language providing shorter representation of the transferred objects what is crucial to save limited mobile devices network transfer.

3 tCOP Information Fusion from Heterogeneous Data Sources

The iCOP integration mechanisms utilize several integration approaches and mechanisms performing on three levels of fusion – data association, state estimation and decision fusion. Each of these layers should be implemented to achieve optimal, fully automatic effect of gathering and processing appropriate information. mCOP applies data association and state estimation algorithms leaving decisions for decision maker – soldiers wearing DSS outfit. The aim of data association is to determine a set of observations which comes from the same target in time [8]. It is necessary to keep in mind that observation generated from each sensor is received in fusion node in discrete time, whilst the sensor is not obliged to deliver data in significant time period. Moreover, some of observed information can be noisy [10]. In DSS solution data association algorithms are used to group sensed data in packages corresponding to observed object which serves as a base for state estimation methods. To provide a clear insight in the problem one should assume two objects coming from the same direction with colliding courses. The core of data association problem is to determine which of gathered data came from certain object. mCOP implements three data association techniques – nearest-neighbor algorithm, K-means algorithm and probabilistic data association including credibility factor. Obtained results are passed through majority voting mechanism. Such approach compensate pros and cons of used methods reducing disadvantages coming from separate usage. State estimation mechanisms are used to discover the values of state vector basing on given observations. From mathematical side it means that it is needed to find a set of appropriate parameters with given redundant observations.

To enrich DSS solution mCOP implements particle filter – one of the most flexible state estimation technique which improves situational awareness. It is divided in two phases: prediction phase and update phase. While being in prediction phase each of the particles are modified accordingly with existing model compensating sums of random noise. In update phase particle wages are calculated basing on recent sensor's observation, while the particles with lower wages are removed [9].

mCOP uses state estimation algorithms mainly in order to determine the state of the target especially while moving. Estimated state of the target improves COP and delivers additional information supporting decision maker. One of the most

representative example is assessing the level of fatigue of hostile object basing inter alia on movement speed, terrain parameters and observation time.

As the data fusion method is highly connected with various data input sources, mCOP utilizes several standardized information services. One of mostly used in COP environments are NFFI (NATO Friendly Forces Information) and TSO (Tactical Situation Object). In mCOP both these sources can be configured in order to receive information provided from local or remote instances (Fig. 3).

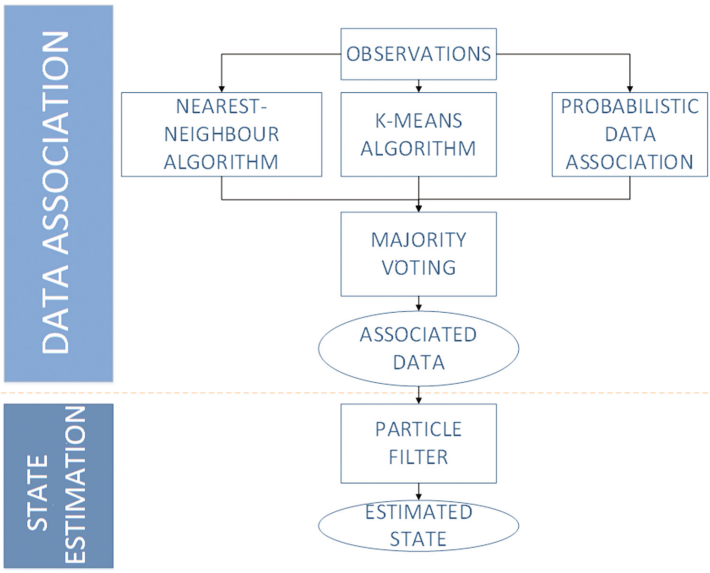


Fig. 3. Data fusion algorithm implementation in DSS solution fusing operational picture products in mCOP software

4 COP Environment Architecture

tCOP system provides COP utilizing data fusion algorithms and integration connectors. This tool enable allied (NATO) soldiers equipped with handheld devices to use fusion algorithms to present situation picture which leads to increase their perception of the battle space environment within a volume of time and space which is known as a SA (Situation Awareness). Soldiers equipped with mCOP application and having wireless tactical networks access can be given visualization of the battle space. Such visualization expresses current situation including both military and civilian objects, information concerning both in AOR (Area of Responsibility) and in AOI (Area of Interest). Additionally, mobile client is delivered with a demo visualization of the command center utilizing web browser as a thin client COP provider. This application is responsible for rendering COP data and providing GIS (Geographic Information System) support for reflecting current or planned operational situation expressed in tactical symbology: TSO and APP6A. tCOP services have been developed in J2EE

environment to provide scalable, reliable and secure application. Business logic has been implemented according to SOA approach utilizing EJB 3.1 (Enterprise Java Beans) (Fig. 4).

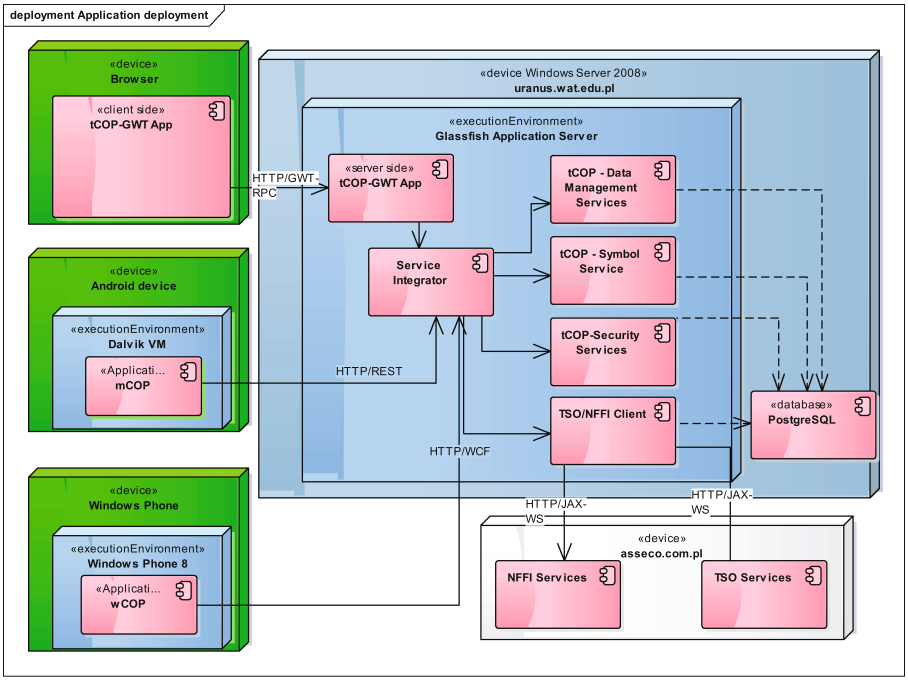


Fig. 4. Developed application deployment model [3]

Multilayer divided architecture allocate different responsibilities to separate layers improving flexibility and maintenance factors. Each component provides interfaces for both remote JNDI and web services invocations. Web services are used by mobile client to communicate with server to increase situation awareness of particular soldier. Moreover mobile client can use a server as a proxy for a communication with other similar, JC3IEDM compatible command support systems – e.g. JAŚMIN. These web services are implemented in two currently common approaches: REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) protocols. Both services are implemented as a Business Delegate object's over HTTP (Hypertext Transfer Protocol) requests. These components are responsible for process request in particular format and transfer responsibility for logic execution to EJB component. REST services utilizes JAX-RS library to provide communication base both for JSON (JavaScript Object Notation) and XML (Extensible Markup Language) communication format. SOAP services are encapsulated using JAX-WS (Java API for XML Web Services). Additionally, tCOP implements a web based application according to RIA (Rich Internet Application) which uses asynchronous methods invocation by AJAX (Asynchronous JavaScript and XML) encapsulated into GWT-RPC protocol and

GWT-RPC servlets responsible for processing this messages and transmitting them to EJB components. The persistence layer have been implemented according to DAO (Data Access Object) approach utilizing JPA 2.1 (Java Persistence API) which defines a specification for persisting, accessing and managing the data between Java objects and the relational database as an ORM (Object-Relational Mapping). EclipseLink has been chosen as a JPA implementation provider.

Access to each exposed web service is secured by authentication and authorization services which filter HTTP responses depending on existing HTTP headers or request parameters describing server session. The security layer is implemented according to SSO (Single Sign On) recommendations to simplify distributed and multiplatform authentication. This approach simplifies trainee access to C4ISR system and shorten time needed for both registration and authentication. According to web services standards REST and SOAP are transferred over HTTP requests with plain text.

5 Supporting Soldier's Situation Awareness

Dismounted soldier systems often limits to display current tactical situation in GIS environment. mCOP implements functions that could assist soldier on the battlefield, i.e. March trail recording, orienting in the field, monitoring life-functions or reporting to higher levels of command. Nowadays mobile devices are not only able to perform demanding calculations but also provide an array of sensors. Most commonly used are accelerometer, magnetometer and gyroscope. Fused data from these sensors allow to implement a compass and further fusion with GPS receiver and camera allows to display tactical data in form of Augmented Reality view.

The mCOP system widely described in [3] was designed to utilize data from many sources to empower soldier's situation awareness. The core of utility functions of mCOP were encapsulated within two views that are shown on Fig. 5.

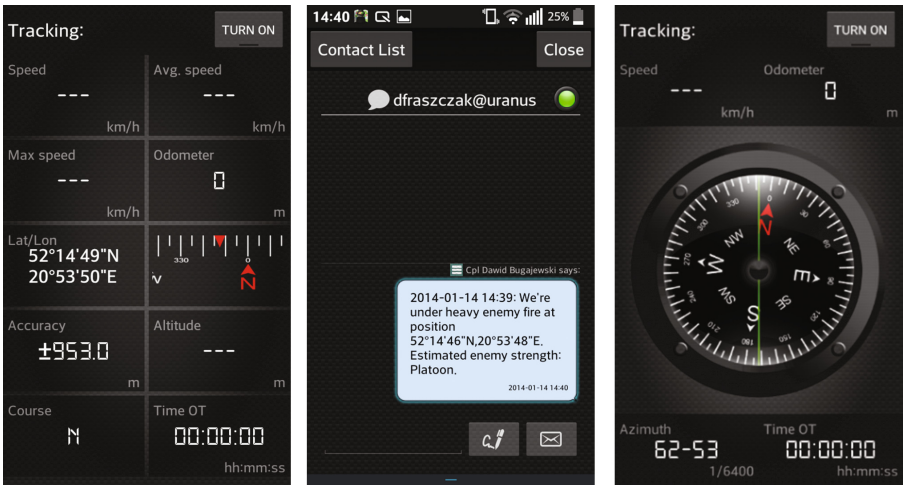


Fig. 5. Utility views: (left) movement monitoring during operation, (center) tactical communication messenger with report template support, (right) compass utility view

These screens present fused data from sensors to display current information about speed, distance, position and azimuth. Furthermore after turning tracking on the data about position is gathered, collected into packs and sent to the backend when the connection is available. In order to simplify communication with command center via XMPP protocol a message template language was introduced. These templates deliver rapid report composition using current situation and location. mCOP specifies an array of attributes filled automatically on report creation: name, function, geoposition and time.

6 Construction of Augmented Reality View

Building an augmented reality view required utilization of the sensor, battlespace and GPS data combined with camera display. Construction of augmented reality view utilizes location of battlespace elements, location and inertial sensors. Developed picture is further Displaying an augmented reality view requires definition of two crucial parameters (Fig. 6):

Fig. 6. AR based situation awareness view merging sensor data, battlespace information and threat level estimates (1 – deflection, 2 – location, 3 – elevation, 4 – azimuth)

Vertical field of vision and horizontal field of vision. These parameters are the height and width of 2D projection of rectangle defined on sphere around the soldiers head which camera displays on the screen

$$size = \max\left\{0.1, 1.0 - \frac{d}{d_{\max}}\right\} \cdot size_{\max} \tag{1.1}$$

where $size$ – displayed unit size, $size_{max}$ – maximal unit size, d – distance to displayed unit, d_{max} – maximal distance to displayed unit.

These parameters are crucial to displaying augmented reality properly in terms of locating the tactical objects in correct location on the screen. To make the augmented reality view easier to interpret the size of unit image was determined by its distance from the soldier. The size is calculated according to Eq. (1.1) The height on which the unit is displayed is not determined by its z-coordinate of location but by its rank. The units of lowest rank are displayed on the horizon line where the units of higher rank in defined intervals over the horizon line.

7 Construction of Situation Awareness View Using AR

To immerse even further soldier senses in mCOP system, an augmented reality view has been developed. The VR view was based on our Augmented Reality view and as prototype was designed as 2D plane displayed in 3D space. Such approach allows to rapidly prototype the view and asses its capabilities (Fig. 7).

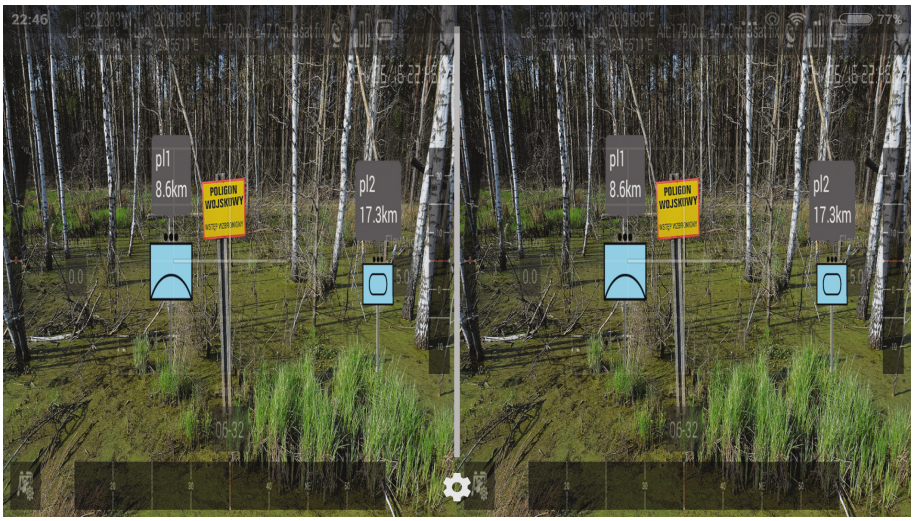


Fig. 7. Augmented reality view generated by mCOP tool and fusing available information of operational elements and tactical geopoints with appropriate threat level estimates - no correction and adjustment for VR projection glasses

From the technical point of view both the camera and the legacy android augmented reality view where rendered as OpenGL ES2.0 surface textures and displayed on rectangle. These textures where mixed in shader to achieve the performance needed in real time view. It has to be noticed that while creating a VR view performance is really important factor as any fps drops or inconsistency can result in dizziness or feeling sick. The OpenGL view was rendered using Google VR SDK which allows to render regular OpenGL scene to Virtual Reality one. This SDK manages splitting the

view for both eyes and applying eventual lenses distort correction for the Google Cardboard. The future improvement perspective for this view is to move the whole augmented reality overlay from legacy android view into OpenGL scene that would allow to achieve correct depth visualization for the units making it true 3D view (Fig. 8).

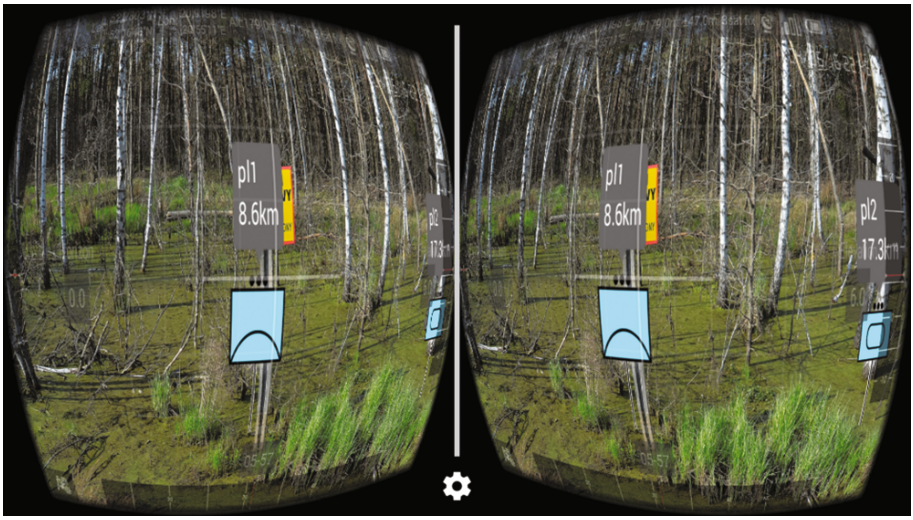


Fig. 8. Augmented reality view generated by mCOP tool lens distortion correction projected in 3D environment supporting VR glasses

8 Conclusions and Future Development

Presented family of COP tools provides the information infrastructure for military and civilian crisis operation support. These tools implement state-of-the-art mechanisms and techniques to increase the efficiency of combat operations on any stage of mission execution including tactical and operational level consuming and producing standardized data sources NFFI, JC3IEDM and TSO dispatch services. The environment have been validated in several deployment configurations utilizing ad-hoc tactical networks, server environments and variety of android devices (also ruggedized). The iCOP platform is able to process and refine operational scenario elements fusing the data from many heterogeneous data sources. Application of AR to support situation awareness of TSU member delivers new means of information rendering accuracy and increases decision making efficiency during combat. mCOP VR technology in conjunction with appropriate head gear provides means of situation awareness projection supplemented by threat level estimates for every battlespace element. The iCOP tools are currently being adopted for ruggedized closed platforms as DSS demonstrator technologies for Polish Armed Forces and crisis management institutions.

Acknowledgements. This work was partially supported by Grant RMN-948, RMN-815, RMN-817, research project DOBR/0023/R/ID3/2013/03 and supported by the National Center For Research and Development.

References

1. Chmielewski, M., Gałka, A.: Semantic battlespace data mapping using tactical symbology. In: Advances in Intelligent Information and Database Systems. Studies in Computational Intelligence, vol. 283, pp. 157–168 (2010)
2. Hall, D.L., Llinas, J.: Handbook of Multisensor Data Fusion. CRC Press, Boca Raton (2001)
3. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. IEEE Trans. Signal Process. **50**, 736–746 (2002)
4. Shalom, Y., Tse, E.: Tracking in a cluttered environment with probabilistic data association. Automatica **11**, 451–460 (1975)
5. Blasch, E., Bosse, E., Lambert, D.A.: High-Level Information Fusion Management and System Design. Artech House, Norwood (2012)
6. NATO: NATO Friendly Force Interoperability Standard, STANAG 5527
7. C. WS/ISDEM: Definition of the OASIS Tactical Situation Object (2006)
8. Chmielewski, M., Kulas, W., Kukiełka, M., Frąszczak, D., Bugajewski, D., Kędzior, J., Rainko, D., Stąpor, I.P.: Development of Operational Picture in DSS Using Distributed SOA Based Environment, Tactical Networks and Handhelds. Library of Informatics of University Level Schools, Wrocław (2014)
9. Chmielewski, M.: Ontology applications for achieving situation awareness in military decision support systems. In: ICCCI 2009. LNCS, vol. 5796, pp. 528–539 (2009). doi:10.1007/978-3-642-04441-0_46
10. Chmielewski M., et al.: Development of operational picture in DSS using distributed SOA based environment, tactical networks and handhelds. In: Information Systems Architecture and Technology: Selected Aspects of Communication and Computational Systems, Wroclaw (2014). ISBN/ISSN 978-83-7493-856-3
11. Chmielewski M., Kukielka M.: Applications of RFID technology in dismounted soldier solution systems – study of mCOP system capabilities. In: CSCC 2016, MATEC Web of Conferences, vol. 7 (2016). Article No. 04014. doi:10.1051/matecconf/20167604014
12. Endsley, M.R., Garland, D.J.: Situation Awareness, Analysis and Measurement. CRC Press, Mahwah (2000)
13. NC3A (NATO Consultation, Command and Control Agency): NATO Common Operational Picture (NCOP)-NC3A (2006)
14. UK MoD (UK Ministry of Defence): Network Enabled Capability, Joint Service Publication 777 (2005). http://barrington.cranfield.ac.uk/resources/papers/JSP777. Accessed 1 Apr 2017
15. US DoD (US Department of Defense): Department of Defense Dictionary of Military and Associated Terms, Joint Publication 1-02, 17 March 2009 (2009). Accessed 1 Apr 2017

A Practical Approach to Tiling Zuker's RNA Folding Using the Transitive Closure of Loop Dependence Graphs

Marek Palkowski(✉) and Wlodzimierz Bielecki

Faculty of Computer Science and Information Systems, West Pomeranian University of Technology in Szczecin, Zolnierska 49, 71210 Szczecin, Poland
{mpalkowski, wbielecki}@wi.zut.edu.pl

Abstract. RNA secondary structure prediction is a computationally-intensive task that lies at the core of search applications in bioinformatics. In this paper, we consider Zuker's RNA folding algorithm, which is challenging to optimize because it is resource intensive and has a large number of non-uniform dependences. We describe the application of a previously published approach, proposed by us, to automatic tiling Zuker's RNA Folding loop nest using the exact polyhedral representation of dependences exposed for this nest. First, rectangular tiles are formed within the iteration space of Zuker's loop nest. Then tiles are corrected to honor all dependences, exposed for the original loop nest, by means of applying the exact transitive closure of a dependence graph. We implemented our approach as a part of the source-to-source TRACO compiler. The experimental results present the significant speed-up factor of tiled code on a single core of a modern processor. Related work and future algorithm improvements are discussed.

Keywords: RNA secondary structure prediction · High performance computing · Zuker · Polyhedral model · Loop tiling · Transitive closure

1 Introduction

The acceleration of dynamic programming recurrences in computational biology is still a challenging task for developers and researchers. On modern architectures, memory latency and bandwidth have become a major limiting factor in achieving good performance. Fortunately, many of time-consuming bioinformatics programs, such as energy minimization algorithms for RNA folding, involve mathematical computation over affine control loops whose iteration space can be represented by polyhedral models [15]. Those models are a powerful theoretical framework that can represent and analyze regular loop programs with static dependences. Loop transformations such as tiling allow us to expose far more locality than *ad hoc* methods.

In this paper, we focus on the automatic code locality improvement of RNA folding realized with Zuker's algorithm [21]. Zuker's dynamic programming algorithm predicts the structure using more detailed and accurate energy models [8] and entails a large number of non-uniform loop dependences.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_18

We apply loop tiling, or loop blocking, to Zuker's loop nest. Tiling for improving locality groups loop nest statement instances in the loop nest iteration space into smaller blocks (tiles) allowing reuse when the block fits in local memory.

To our best knowledge, well-known tiling techniques are based on linear or affine transformations [4, 6, 7, 20]. In paper [2], we presented a novel algorithm to tile affine arbitrary nested loops which is based on the transitive closure of program dependence graphs. First, rectangular tiles are formed, and then they are corrected to establish tiling validity by means of the transitive closure of loop nest dependence graphs. The approach is able to tile non-fully permutable loops with non-uniform data dependences which are exposed for energy minimization algorithms.

In this paper, we adopt this technique to tile the code of a simplified version of Zuker's algorithm where the energy contribution of multiple loops is set to zero and dangling single bases have no effect. Such an algorithm is implemented in the UNAFold package [13].

The rest of the paper is organized as follows. Section 2 explores related work. Section 3 presents Zuker's algorithm. Tiling is explained in Sect. 4. Section 5 discusses results of experiments and demonstrates that the tiled (optimized) code of Zuker's algorithm is dramatically faster than original one for modern processors. Section 6 concludes the paper.

2 Related Work

Recently, significant effort has been spent in accelerating Zuker's algorithm in a number of ways on multi-core processors, graphics accelerators, and FPGAs [8, 12–15]. It frequently focuses on parallel, rather than single-core, performance, or includes hardware implementations.

GTfold [14] is a well-known optimized multi-core implementation of RNA secondary structure prediction algorithms. It optimizes memory layout of the arrays to improve spatial locality. However, GTfold does not perform tiling to improve temporal locality.

The state-of-the-art source-to-source Pluto compiler [4] is able to tile RNA folding loop nests. It forms and applies affine transformations to generate tiled code. However Pluto fails to generate tiles of the maximal dimensions because the tile dimensionality is limited to the number of linearly independent solutions to the space/time partition constraints. For example, Pluto produces only 2-d tiles for the Nussinov dependence patterns [15] and fails to tile general linear recurrence equations [2].

Mullapudi and Bondhugula presented dynamic tiling for Zuker's's optimal RNA secondary structure prediction [15]. 3-d iterative tiling for dynamic scheduling is calculated by means of reduction chains. Operations along each chain can be reordered to eliminate cycles in an inter-tile dependence graph. But this technique is not able to generate static tiled code.

Wonnacott et al. introduced 3-d tiling of "mostly-tileable" loop nests of RNA secondary-structure prediction codes in paper [19]. However the authors presented only a script for generating tiled Nussinov's loop nest. An open question is whether that

technique is able to tile other RNA folding codes and what is the performance of such codes [9].

In our previous work, we have accelerated Nussinov's RNA folding loop nest [16] by means of the exact transitive closure of loop dependence graphs. We were able to analyze the recurrence using the polyhedral framework to improve cache usage in arbitrary nested loops with non-uniform dependences [2] but Zuker's loop nests were never considered by us.

3 Zuker's Algorithm

Zuker's algorithm is executed in two steps. First, it calculates the minimal free energy of the input RNA sequence on recurrence relations as shown in the formulas below. Then, it performs a trace-back to recover the secondary structure with the base pairs. The first step consumes almost all of the total execution time. Thus, optimization of computing energy matrices is crucial to improve code performance.

Zuker defines two energy matrices, *W(i, j)* and *V(i, j)*, defined over the domain $D = \{1 \le i \le N;\ i \le j \le N\}$, where N is the length of a sequence. *W(i, j)* is the total free energy of subsequence i and j, *V(i, j)* is defined as the total free energy of subsequence $i \bullet j$ if i and j pairs, otherwise $V(i,j) = \infty$.

There exist several possibilities to compute matrix *V(i, j)* [9]. Base pair $i \bullet j$ can be part of a hairpin loop (*eH*), or can form part of a stack (*eS*) on base pair $i + 1 \bullet j - 1$, or can be part of a bulge/interior loop with the *eL* energy function, or can be part of a multiple loop (*W*). This leads to the following equations

$$V(i,j) = \begin{cases} eH(i,j) \\ V(i+1, j-1) + eS(i,j) \\ \min\limits_{\substack{i \le i' \le j' \le j \\ i'-i+j'-j \le 2}} \{V(i', j') + eL(i, j, i', j')\} \\ \min\limits_{i<k<j-1} \{W(i+1, k) + W(k+1, j-1)\} \end{cases} \tag{1}$$

The recursion formula for *W(i, j)* is as follows

$$W(i,j) = \begin{cases} W(i+1, j) \\ W(i, j-1) \\ V(i,j) \\ \min\limits_{i<k<j} \{W(i, k) + W(k+1, j)\} \end{cases} \tag{2}$$

The memory complexity of the defined matrices is $O(n^2)$. The time complexity of a direct implementation of this algorithm is $O(n^4)$ because we need $O(n^4)$ operations to compute the bulge or interior loop energy. However, large internal loops are uncommon in nature, so implementations of Zuker sacrifice accuracy for speed by limiting the size of internal loops (third element of Equation) to at most 30, reducing the overall time complexity to $O(n^3)$, albeit with a large constant factor [8]. Dynamic programming for Zuker's RNA folding is guaranteed to give the mathematically optimal structure.

Any lack of prediction accuracy is more the scoring system's problem than the algorithm's problem [5].

The computation domain and dependences for Zuker's recurrence cell *(i, j)* are shown in Fig. 1 [8]. Long-range (non-local) dependences (fourth elements of Eqs. (1) and (2)) of the cell (*i, j*) are depicted in gray boxes. Therefore, the Zuker data dependences result in a non-uniform structure. Dependences for third element of Eq. 1 are in the triangle, whose area is limited to several dozens or hundreds of cells in nature. The other equations present short-range (local) dependences.

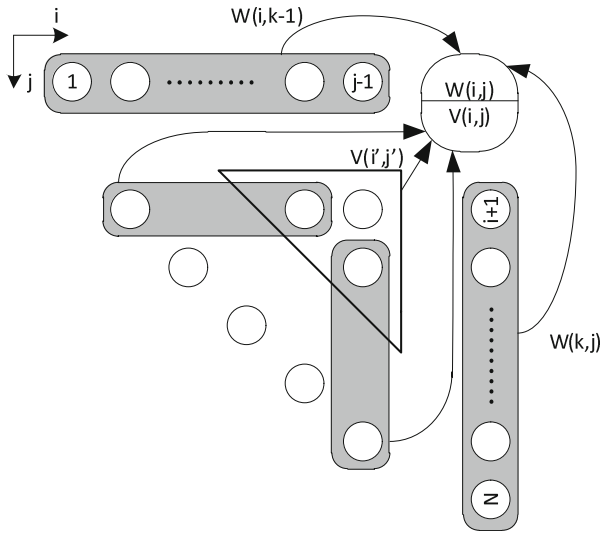


Fig. 1. Loop dependences for the cell (i, j) of Zuker's recurrence.

4 Tiling Zuker's Loop Nests

4.1 Background

Dependence analysis produces execution-order constraints between statements/instructions and is required to guarantee the validity of loop nest transformations. Two statement instances *I* and *J* are *dependent* if both access the same memory location and if at least one access is a write.

If the difference of iteration vector *J* and *I* is constant for dependent statement instances *T(J)* and *S(I)*, we call the dependence *uniform*, otherwise the dependence is *non-uniform*.

We use an exact representation of loop-carried dependences and consequently an exact dependence analysis which detects a dependence if and only if it actually exists. The dependence analysis proposed by Pugh and Wonnacott [17] was chosen by us, where dependences are represented with relations.

A dependence relation is a tuple relation of the form [*input list*]→ [*output list*]: *formula*, where *input list* and *output list* are the lists of variables and/or expressions used to describe input and output tuples and *formula* describes the constraints imposed upon *input list* and *output list* and it is a Presburger formula built of constraints represented with algebraic expressions and using logical and existential operators [17].

Standard operations on relations and sets are used, such as intersection ($\cap$), union ($\cup$), difference ($-$), domain (dom R), range (ran R), relation application ($S' = R(S) : e' \in S'$ iff exists e s.t. $e \rightarrow e' \in R,\ e \in S$). The detailed description of these operations is presented in [10, 18].

The positive transitive closure of a given relation R, R^+, is defined as follows [10]:

$$R^+ = \{e \rightarrow e' : e \rightarrow e' \in R \wedge \exists e'' \, s.t. e \rightarrow e'' \in R \wedge e'' \rightarrow e' \in R^+\} \tag{3}$$

It describes which vertices e' in a dependence graph (represented by relation R) are connected directly or transitively with vertex e.

Transitive closure, R*, is defined as $R^* = R^+ \cup I$, where I is the identity relation. It describes the same connections in a dependence graph (represented by R) that R^+ does plus connections of each vertex with itself.

In sequential loop nests, the iteration i, represented with vector $(i_1, i_2, \ldots, i_k, \ldots, i_n)^T$, executes before iteration j, represented with vector $(j_1, j_2, \ldots, j_k, \ldots, j_n)^T$ if i is *lexicographically less* than j, denoted as $i \prec j$, i.e., $i_1 < j_1 \wedge \exists\, k \leq 1 : i_k = j_k \wedge i_t = j_t$, for $t < k$.

Figures 2 (a) and (b) present dependence relations and positive transitive closure, respectively. Complete transitive dependences are shown only for the one iteration in the third row to avoid cluttering the diagram.

4.2 Tiling Zuker's Loop Nest

In this subsection, we explain how to apply the tiling technique [2], previously published by us, to generate tiled Zuker's RNA folding loop nest using the transitive closure of the dependence graph.

We carried out a dependence analysis by means of the Petit tool [11] to get 31 dependence relations for Zuker's loop nest. It is worth noting that 21 relations represent non-uniform dependences.

To obtain the exact[1] transitive closure of the union of all extracted relations, represented the whole dependence graph, we applied the iterative method presented in paper [1]. That approach uses basis dependence distance vectors in the modified Floyd-Warshall algorithm.

To generate valid target tiled code, first we form set, *TILE*, including statement instances belonging to a rectangular tile (with respect to the tile identifier). Next, we form sets, *TILE_LT* and *TILE_GT*, that are the unions of all the tiles whose identifiers are lexicographically less and greater than that of set *TILE*, respectively.

[1] without non-existing dependence relations.

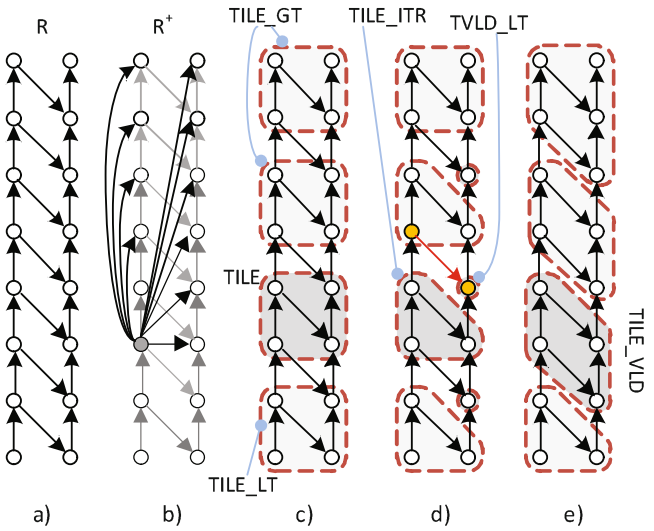


Fig. 2. An example iteration space and its tilings (a) dependence relations, (b) positive transitive closure and transitive dependences for one cell, (c) original rectangular tiles, *TILE*; tiles, *TILE_LT* and *TILE_GT*, whose identifiers are lexicographically less and greater than the identifier of *TILE*; (d) tiles *TILE_ITR* and *TVLD_LT*; (e) valid target tiles, *TILE_VLD*.

Below, we present the mathematical descriptions of those sets.

*TILE(**II**, **B**)* includes statement instances belonging to a tile (with respect to ***II***), as follows

$$TILE(\mathbf{II}, \mathbf{B}) = \{[\boldsymbol{I}] \mid \boldsymbol{B} * \boldsymbol{II} + \boldsymbol{LB} \leq \boldsymbol{I} \leq \min(\boldsymbol{B} * (\boldsymbol{II} + \mathbf{1}) + \boldsymbol{LB} - \mathbf{1}, \boldsymbol{UB}) \wedge \boldsymbol{II} \geq 0\},$$

where vectors ***LB*** and ***UB*** include the lower and upper loop index bounds of the original loop nest, respectively; diagonal matrix ***B*** defines the size of a rectangular original tile; elements of vector ***I*** represent the original loop nest iterations contained in the tile whose identifier is ***II***; ***1*** (or ***0***) is the vector whose all elements have value 1 (or 0)[2], *TILE_LT* is the union of all the tiles whose identifiers are lexicographically less than that of *TILE(**II**, **B**)*:

$$TILE\_LT = \{[\boldsymbol{I}] \mid \exists \boldsymbol{II}' \text{ s.t. } \boldsymbol{II}' \prec \boldsymbol{II} \wedge \boldsymbol{II} \geq 0 \wedge \boldsymbol{B} * \boldsymbol{II} + \boldsymbol{LB} \leq \boldsymbol{UB} \wedge \boldsymbol{II}' \geq 0 \wedge \boldsymbol{B} * \boldsymbol{II}' + \boldsymbol{LB} \leq \boldsymbol{UB} \wedge \boldsymbol{I} \in TILE(\boldsymbol{II}', \boldsymbol{B})\},$$

TILE_GT is the union of all the tiles whose identifiers are lexicographically greater than that of *TILE(**II**, **B**)*:

$$TILE\_LT = \{[\boldsymbol{I}] \mid \exists \boldsymbol{II}' \text{ s.t. } \boldsymbol{II}' \succ \boldsymbol{II} \wedge \boldsymbol{II} \geq 0 \wedge \boldsymbol{B} * \boldsymbol{II} + \boldsymbol{LB} \leq \boldsymbol{UB} \wedge \boldsymbol{II}' \geq 0 \wedge \boldsymbol{B} * \boldsymbol{II}' + \boldsymbol{LB} \leq \boldsymbol{UB} \wedge \boldsymbol{I} \in TILE(\boldsymbol{II}', \boldsymbol{B})\},$$

[2] The notation $x \geq (\leq) y$ where x, y are two vectors in Z^n corresponds to the component-wise inequality, that is, $x \geq (\leq) y \Leftrightarrow x_i \geq (\leq) y_i$, $i = 1, 2, \ldots, n$.

These sets are illustrated in Fig. 2(c).

Next, using the exact form of R+, we calculate set

$$TILE\_ITR \;=\; TILE - R^{+}(TILE\_GT),$$

which does not include any invalid dependence target, i.e., it does not include any dependence target whose source is within set *TILE_GT*.

The following set

$$TVLD\_LT \;=\; (R^{+}(TILE\_ITR) \cap TILE\_LT) - R^{+}(TILE\_GT)$$

includes all the iterations that (i) belong to the tiles whose identifiers are lexicographically less than that of set *TILE_ITR*, (ii) are the targets of the dependences whose sources are contained in set *TILE_ITR*, and (iii) are not any target of a dependence whose source belong to set *TILE_GT*.

Sets *TILE_ITR* and *TVLD_LT* are illustrated in Fig. 2(d).

Target tiles are defined by the following set

$$TILE\_VLD \;=\; TILE\_ITR \cup TVLD\_LT,$$

and are illustrated in Fig. 2(e).

TILE_VLD_EXT is built by means of inserting (i) into the first positions of the tuple of set *TILE_VLD* elements of vector $\boldsymbol{II}$: $ii_1, ii_2, \ldots, ii_d$; (ii) into the constraints of set *TILE_VLD* the constraints defining tile identifiers $\boldsymbol{II} \geq 0 \wedge \boldsymbol{B} * \boldsymbol{II} + \boldsymbol{LB} \leq \boldsymbol{UB}$.

Target code is generated by means of applying any code generator allowing for scanning elements of set *TILE_VLD_EXT* in the lexicographic order, for example, isl AST [18]. The details of the applied technique and its proof are presented in our paper [2].

5 Experimental Study

The technique to generate tiled code, presented in this paper, was implemented by us in the publicly available TRACO compiler. Using this compiler, we generated tiled Zuker's loop nest for different values of *N, V*, and *W* (see Table 1). To carry out experiments, we have used a machine with a processor Intel Xeon E5-2699 v3 (3.6 Ghz, 576 KB L1 Cache for code and data separately, 4.5 MB L2 Cache and 45 MB L3 Cache) and 128 GB RAM. All programs were compiled by means of the Intel C++ Compiler (*icc* 15.0.2) with the −O3 flag of optimization. By carried out experiments, we found that the best size of original rectangular tiles is $16 \times 16 \times 16 \times 16$.

We used randomly generated RNA strands from the alphabet {*G* (guanine), *A* (adenine), *U* (uracil), *C* (cytosine)}. This decision was made on the basis that the algorithm performance does not change based on the strings themselves, but by their size. Hence, there is no difference, performance wise, on actual sequences versus randomly generated sequences.

Table 1. Execution times of tiled Zuker's loop nest (in seconds)

N	V, W size (MB)	Original	TRACO	Pluto+
500	7.5	8.26	4.31	4.46
600	11	16.64	8.33	9.08
700	15	30.62	14.81	16.39
800	19.5	50.97	25.04	27.77
900	25	80.64	39.01	43.24
1000	30.5	122.57	59.22	65.41
1100	37	176.37	85.36	95.66
1200	43	250.35	119.16	134.02
1300	51.5	341.70	163.33	182.45
1400	60	458.67	216.24	241.89
1500	69	608.07	282.74	318.32
1600	78	786.11	364.01	408.94
1700	88	1000.61	464.57	518.77
1800	99	1229.58	570.44	648.48
1900	110	1523.81	720.04	803.02
2000	122	1878.31	874.83	981.11

The results in Table 1 shows single-thread execution times and demonstrates that tiled codes are significantly faster than original sequential ones for various lengths of a sequence N. The second column presents the memory size of energy matrices V and W (in Megabytes).

The last column of the table presents execution time of tiled code generated by means of the state-of-the-art Pluto+ compiler [3]. The tool is not able to tile the third loop nest automatically because of cycles existence in the dependence graph. Ana-

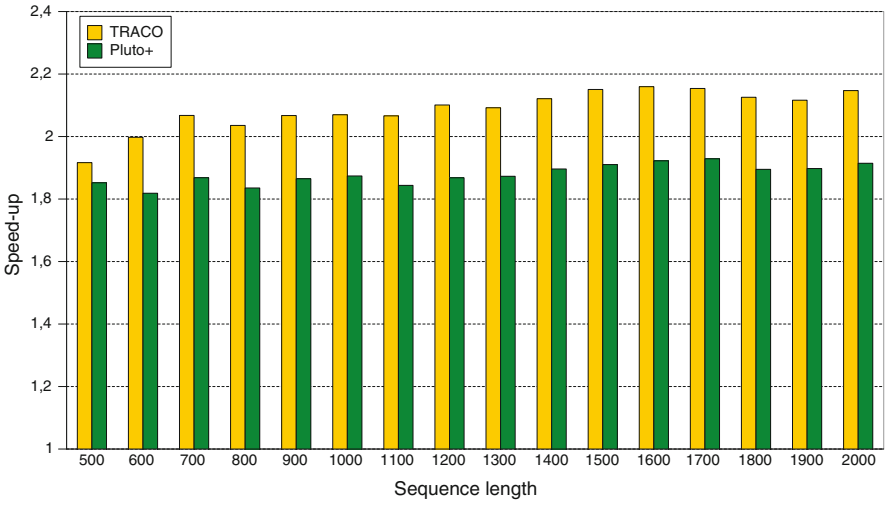


Fig. 3. Speed-up of the tiled and single-thread Zuker's RNA folding implementations.

lyzing the data in columns 4 and 5, we may conclude that tiled code generated with the technique presented in this paper is from 3\% to 12\% faster than that generated with Pluto+. Speed-ups, ratios of original and tiled code time executions, are depicted in Fig. 3. The proposed approach leads to better reuse of Zuker's arrays for the all considered lengths of a sequence *N*.

The source codes and tiled ones generated by means of the TRACO and Pluto+ compilers are available at the following repository https://sourceforge.net/projects/traco.

6 Conclusion

In this paper, we presented the usage of the transitive closure of loop nest dependence graphs to generate tiled code for Zuker's RNA folding algorithm. The presented technique was implemented in the publicly available TRACO compiler. Tiled Zuker's loop nests, automatically generated with TRACO, demonstrate improved code locality that results in outperforming original Zuker's loop nest in average by a factor of 2.15 on a single core of modern processors.

In the longer-term, we plan to apply the presented approach to other challenging dynamic programming codes and in particular to more recent algorithms for RNA folding. We also plan to parallelize generated serial tiled code and examine its performance on multi-core systems.

Acknowledgments. Thanks to the Miclab Team (`miclab.pl`) from the Technical University of Czestochowa (Poland) that provided access to high performance machines for the experimental study presented in this paper.

References

1. Bielecki, W., Klimek, T., Palkowski, M., Beletska, A.: An iterative algorithm of computing the transitive closure of a union of parameterized affine integer tuple relations. In: COCOA 2010. LNCS, vol. 6508/2010, pp. 104–113 (2010)
2. Bielecki, W., Palkowski, M.: Tiling of arbitrarily nested loops by means of the transitive closure of dependence graphs. Int. J. Appl. Math. Comput. Sci. (AMCS) **26**(4), 919–939 (2016)
3. Bondhugula, U., Acharya, A., Cohen, A.: The Pluto+ algorithm: a practical approach for parallelization and locality optimization of affine loop nests. ACM Trans. Program. Lang. Syst. **38**(3), 12:1–12:32 (2016)
4. Bondhugula, U., et al.: A practical automatic polyhedral parallelizer and locality optimizer. SIGPLAN Not. **43**(6), 101–113 (2008)
5. Eddy, S.: How do RNA folding algorithms work? Nat. Biotechnol. **22**(11), 1457–1458 (2004)
6. Griebl, M.: Automatic parallelization of loop programs for distributed memory architectures (2004)

7. Irigoin, F., Triolet, R.: Supernode partitioning. In: Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 1988, pp. 319–329. ACM, New York (1988)
8. Jacob, A.C., Buhler, J.D., Chamberlain, R.D.: Rapid RNA folding: analysis and acceleration of the Zuker recurrence. In: 2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, pp. 87–94 (2010)
9. Jiang, T., Zhang, M., Xu, Y. (eds.): Current Topics in Computational Molecular Biology. MIT Press, Cambridge (2002)
10. Kelly, W., Maslov, V., Pugh, W., Rosser, E., Shpeisman, T., Wonnacott, D.: The omega library interface guide. Technical report. College Park, MD, USA (1995)
11. Kelly, W., Maslov, V., Pugh, W., Rosser, E., Shpeisman, T., Wonnacott, D.: New User Interface for Petit and Other Extensions (1996)
12. Liu, L., Wang, M., Jiang, J., Li, R., Yang, G.: Efficient nonserial polyadic dynamic programming on the cell processor. In: IPDPS Workshops, pp. 460–471. IEEE, Alaska (2011)
13. Markham, N.R., Zuker, M.: UNAFold, pp. 3–31. Humana Press, Totowa (2008)
14. Mathuriya, A., Bader, D.A., Heitsch, C.E., Harvey, S.C.: Gtfold: a scalable multi-core code for RNA secondary structure prediction. In: Proceedings of the 2009 ACM Symposium on Applied Computing, SAC 2009, pp. 981–988. ACM, New York (2009)
15. Mullapudi, R.T., Bondhugula, U.: Tiling for dynamic scheduling. In: Rajopadhye, S., Verdoolaege, S. (eds.) Proceedings of the 4th International Workshop on Polyhedral Compilation Techniques, Vienna, Austria (2014)
16. Palkowski, M., Bielecki, W.: Parallel tiled Nussinov RNA folding loop nest generated using both dependence graph transitive closure and loop skewing. BMC Bioinform. **18**(1), 290 (2017)
17. Pugh, W., Wonnacott, D.: An exact method for analysis of value-based array data dependences. In: Sixth Annual Workshop on Programming Languages and Compilers for Parallel Computing. Springer (1993)
18. Verdoolaege, S.: Integer set library - manual. Technical report (2011). www.kotnet.org/∼skimo//isl/manual.pdf
19. Wonnacott, D., Jin, T., Lake, A.: Automatic tiling of "mostly-tileable" loop nests. In: 5th International Workshop on Polyhedral Compilation Techniques, IMPACT 2015, Amsterdam, The Netherlands (2015)
20. Xue, J.: Loop Tiling for Parallelism. Kluwer Academic Publishers, Norwell (2000)
21. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**(1), 133–148 (1981)

On Loss Process in a Queueing System Operating Under Single Vacation Policy

Wojciech M. Kempa(✉)

Faculty of Applied Mathematics, Institute of Mathematics,
Silesian University of Technology, ul. Kaszubska 2A, 44-100 Gliwice, Poland
wojciech.kempa@polsl.pl

Abstract. A finite-buffer queueing model with Poisson arrivals and generally distributed processing times is analyzed. Every time when the service station becomes idle a single vacation period is being initialized, during which the processing is suspended. A system of integral equations for the probability distribution of the length of the first loss series is built, using the idea of embedded Markov chain and the total probability law. The solution of the is obtained in a compact form by using the linear algebraic approach. The corresponding result for next series of losses is hence derived. Numerical utility of analytical formulae is presented in a computational example.

Keywords: Embedded Markov chain · Finite-buffer queue · Integral equation · Loss process · Single vacation policy

1 Introduction

Applications of finite-buffer queueing models in modelling and performance evaluation of telecommunication and computer networks, in manufacturing lines, transport and logistic issues are evident. In the analysis of the appropriate queueing model some important stochastic characteristics, like virtual waiting time, queue-size distribution, time to buffer overflow and the probability distribution of lengths of consecutively lost jobs, describing the operation of the system, are very helpful. The knowledge of the behavior of these characteristics seems to be important also in the non-stationary state of the system, not only for the equilibrium. Transient analysis of the system behavior is essential if we are interested e.g. in the investigation of system's evolution just after its opening or reorganizing the control process. The phenomenon of packet losses is a typical one in packet-type networks with finite buffer capacities. Because during the buffer overflow period all the arriving jobs are rejected without processing, the knowledge of the probabilistic behavior of successive buffer overflow periods' durations is essentially important. Besides, the statistical structure of the loss process is very important, since it makes a difference whether the arriving jobs are lost in long series or not.

Analytical results fort probability distributions of buffer overflow durations in the finite- and infinite-buffer M/G/1-type models without limitation in access to the channel can be found e.g. in [1–4]. In [12] the system with general independent input flow and single server vacations is considered and treated via genetic algorithms. The compact-form representations for the distributions of the time to buffer overflow are

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_19

derived in [5, 10], where the GI/M/1/N-type model and the M/G/1/N-type queue with setup/closedown mechanisms are considered, respectively. In [6–9] the transient analysis of queueing systems with single vacation policy or some other control mechanisms implemented in the service process can be found.

In the article we consider the M/G/1/N-type finite-buffer queueing model operating under single vacation policy. Using the analytical approach based on the idea of embedded Markov chain, integral equations and linear algebra, we obtain the explicit formula for the probability mass function of the length of the first lost series of jobs, conditioned by the initial buffer state. Hence we obtain similar result for next such series lengths.

The article is organized as follows. In the next section we give the precise mathematical description of the considered queueing model. In Sect. 3 we deal with the probability distribution of the first lost series' length. The main analytical result is given in Sect. 4. Section 5 is devoted to the case of next successive series of losses. Section 6 present examples of numerical results and the last Sect. 7 contains short conclusions.

2 Model Description

In the paper we deal with the M/G/1/N-type queuing system in which jobs occur according to a Poisson process with intensity $\boldsymbol{\lambda}$ and are being processed individually with a cumulative distribution function $\boldsymbol{F}(\cdot)$, according to the FIFO service discipline. The capacity of the system equals $\boldsymbol{N}$ jobs, i.e. we have $\boldsymbol{N}-1$ places in the buffer queue and one place "in processing". The system may start the operation with a number $\boldsymbol{n}$ of packets accumulated in the buffer, where $0 \leq \boldsymbol{n} \leq \boldsymbol{N}$. Every time when the server becomes idle, the single server vacation with a general-type cumulative distribution function $\boldsymbol{V}(\cdot)$ is being initialized. During the vacation period the processing of jobs is blocked. At the completion epoch of the vacation period, if there is at least one job accumulated in the buffer, the service station restarts processing immediately. In the case of empty buffer, the machine remains active and waits for the first job occurrence.

3 Equations for Length of First Lost Series

We start with introducing the necessary nomenclature. Let us define

$$L_n(k) \underline{\underline{\text{def}}} \mathbf{P}\{\text{length of first lost series} = k | n\}, \tag{1}$$

where $k \geq 1$ and $0 \leq n \leq N$.

Investigate, firstly, the case of the buffer being empty at the starting epoch. Let us observe that the following equation is then true:

$$\begin{aligned} L_0(k) = & \int_{y=0}^{\infty} dV(y) \left[\sum_{j=1}^{N-1} \frac{(\lambda y)^j}{j!} e^{-\lambda y} L_j(k) \right. \\ & + \int_{x=0}^{y} \frac{\lambda^N}{(N-1)!} x^{N-1} e^{-\lambda x} dx \int_{u=0}^{\infty} \left(\frac{(\lambda(u+y-x))^k}{k!} e^{-\lambda(u+y-x)} + L_{N-1}(k) e^{-\lambda(u+y-x)} \right) dF(u) \Bigg] \\ & + L_1(k) \int_0^{\infty} -\lambda x V(x) dx. \end{aligned} \tag{2}$$

Indeed, the first summand under the integral on the right side of (2) relates to the situation in which the buffer does not become saturated before the completion epoch of the vacation period, while the second one describes the case of buffer overflow occurring during the vacation. The last summand on the right side of (2) relates to the case in which the vacation period ends before the first arrival moment.

Now, let us deal with the case of the system that contains a number of jobs before the opening. Since departure epochs are Markovian moments in the evolution of the considered system, then, applying the continuous version of the formula of total probability with respect to the first departure moment u after the opening of the system, we obtain

$$L_n(k) = \int_0^\infty \left[\sum_{j=0}^{N-n} \frac{(\lambda u)^j}{j!} e^{-\lambda u} L_{n+j-1}(k) + \frac{(\lambda u)^{N-n+k}}{(N-n+k)!} e^{-\lambda u} \right] dF(u), \tag{3}$$

where $1 \leq n \leq N$.

Introduce the following notations:

$$p_j(\lambda, x) = p_j(x) \underline{\underline{\text{def}}} \frac{(\lambda x)^j}{j!} e^{-\lambda x}, \tag{4}$$

$$a_j \underline{\underline{\text{def}}} \int_0^\infty \frac{(\lambda x)^j}{j!} e^{-\lambda x} dF(x) = \int_0^\infty p_j(x) dF(x), \tag{5}$$

$$\bar{a}_j \underline{\underline{\text{def}}} \int_0^\infty \frac{(\lambda x)^j}{j!} e^{-\lambda x} dV(x) = \int_0^\infty p_j(x) dV(x) \tag{6}$$

and

$$\begin{aligned} b(k) \underline{\underline{\text{def}}} \frac{\lambda^{N+k}}{(N-1)!k!} \int_{y=0}^\infty e^{-\lambda y} dV(y) \int_{u=0}^\infty e^{-\lambda u} dF(u) \int_{x=0}^y x^{N-1} (u+y-x)^k dx \\ = \lambda \int_{y=0}^\infty dV(y) \int_{u=0}^\infty dF(u) \int_{x=0}^y p_{N-1}(x) p_k(u+y-x) dx. \end{aligned} \tag{7}$$

Now, the equations of the system (2) and (3) can be reformulated to the following ones:

$$L_0(k) = \sum_{j=1}^{N-1} \bar{a}_j L_j(k) + \bar{a}_0 L_1(k) + b(k) + b(0) L_{N-1}(k) = \sum_{j=1}^{N-1} c_j L_j(k) + b(k) \tag{8}$$

where

$$c_j \underline{\underline{\text{def}}} \bar{a}_j + \bar{a}_0 \delta_{j,1} + b(0) \delta_{j,N-1}, \tag{9}$$

and, for $1 \le n \le N$,

$$L_n(k) = \sum_{j=0}^{N-n} a_j L_{n+j-1}(k) + a_{N-n+k}. \tag{10}$$

In (9) the symbol $\delta_{i,j}$ stands for the Kronecker delta function.

Let us apply in (8)–(10) the following substitution:

$$\bar{L}_{N-n}(k) \underline{\underline{\text{def}}} L_n(k), \tag{11}$$

where $0 \le n \le N$.

We obtain from (10)

$$\sum_{j=-1}^{n-1} a_{j+1} \bar{L}_{n-j}(k) - \bar{L}_n(k) = -a_{n+k}, \tag{12}$$

where $0 \le n \le N-1$.

Similarly, from (8) we get

$$\bar{L}_N(k) = \sum_{j=1}^{N-1} c_j \bar{L}_{N-j}(k) + b(k) = \sum_{j=1}^{N-1} c_{N-j} \bar{L}_j(k) + b(k). \tag{13}$$

4 Compact-Form Solution

The following algebraic result is proved in [11]:

Lemma. *Let* $(a_k), k \ge 0, a_0 \ne 0$, *and* $(\tau_k), k \ge 1$, *be sequences. Each solution of the infinite-sized system of linear equations of the form*

$$\sum_{j=-1}^{n-1} a_{j+1} x_{n-j} - x_n = \tau_n, \tag{14}$$

where $n \ge 1$, *can be written as*

$$x_n = MR_n + \sum_{j=1}^{n} R_{n-j} \tau_j, \tag{15}$$

where M *is independent on* n *and the sequence* (R_k) *is defined recursively by the formulae*

$$R_0 = 0, R_1 = \frac{1}{a_0}, R_{j+1} = R_1\left(R_j - \sum_{i=0}^{j} a_{i+1}R_{j-i}\right), \tag{16}$$

for $j \geq 1$.

Observe that the system (12) has a form similar to (14) with unknown functions $\bar{L}_n(k)$ also depending on k, and with $\tau_n = \tau_n(k) = -a_{n+k}$. Hence, in general, $M = M(k)$, and utilizing (15), we can write now

$$\bar{L}_n(\mathrm{k}) = M(k)R_n + \sum_{j=1}^{n} R_{n-j}\tau_j(k) = M(k)R_n - \sum_{j=1}^{n} R_{n-j}a_{j+k}, \tag{17}$$

where $n \geq 1$.

Evidently, we need the formulae for $M(k)$ and $\bar{L}_0(\mathrm{k})$, due to the fact that the formula (17) holds only for $n \geq 1$.

Substituting, firstly, $n = 1$ in (12), we get

$$\bar{L}_0(\mathrm{k}) = a_0\bar{L}_1(\mathrm{k}) + a_k. \tag{18}$$

To find the representation for $M(k)$, let us apply (17) in (13). We obtain

$$M(k)R_N - \sum_{j=1}^{N} R_{N-j}a_{j+k} = b(k) + \sum_{j=1}^{N-1} c_{N-j}\left[M(k)R_j - \sum_{i=1}^{j} R_{j-i}a_{i+k}\right] \tag{19}$$

and hence

$$M(k) = \frac{\sum_{j=1}^{N} R_{N-j}a_{j+k} - \sum_{j=1}^{N-1} c_{N-j}\sum_{i=1}^{j} R_{j-i}a_{i+k} + b(k)}{R_N - \sum_{j=1}^{N-1} c_{N-j}R_j}. \tag{20}$$

Taking into consideration (11), (17) and (20), we can formulate the following main result:

Theorem. *The probability mass function $L_n(k)$ of the length of the first lost series of jobs, on condition that the buffer initially contains n jobs, can be represented in the form*

$$\begin{aligned} L_n(k) = {} & \frac{\sum_{j=1}^{N} R_{N-j}a_{j+k} - \sum_{j=1}^{N-1} c_{N-j}\sum_{i=1}^{j} R_{j-i}a_{i+k} + b(k)}{R_N - \sum_{j=1}^{N-1} c_{N-j}R_j} R_{N-n} \\ & - \sum_{j=1}^{N-n} R_{N-n-j}a_{j+k}, \end{aligned} \tag{21}$$

where $0 \leq n \leq N-1$, *and*

$$L_N(k) = a_0 L_{N-1}(k) + a_k, \tag{22}$$

where the formulae for $a_j, b(k), c_j$ *and* R_j *are given in (5), (7), (9) and (16), respectively.*

5 The Case of Next Lost Series

Let us denote by $L_n^{(r)}(k)$ the probability that the length of the r th lost series equals k, on condition that the system starts the operation with n jobs accumulated in the buffer, where $r \geq 2$. Since after the first lost series, the number of jobs present in the buffer always equals $N - 1$ (due to the individual service organization), then the probability distribution of next lost series does not depend on the initial system state and we obtain

$$L_n^{(r)}(k) = L_{N-1}(k), \tag{23}$$

where $k \geq 1, 0 \leq n \leq N$.

6 Numerical Example

As an example let us consider the model with exponential vacation duration and 2-Erlang service times. Investigate the dependence of the probability mass function of the first lost series' length on system parameters for $N = 5$. In Fig. 1 the case of the buffer containing initially $n = 4$ jobs is visualized, where the intensity of Poisson arrivals equals $\lambda = 1.0$, while means of service times and vacation durations are 2.0 and 0.5, respectively. The situation of the buffer being empty initially is presented in Fig. 2. Figures 1 and 2 show that, in the case of non-exponential service time, the probability distribution of the length of the series of consecutively lost jobs essentially depends on the initial buffer state.

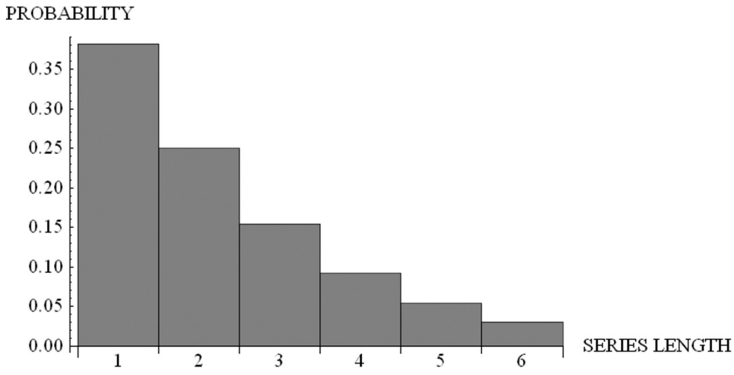


Fig. 1. Distribution of the first lost series' length for mean service time 2.0 and mean vacation time 0.5 and for $n = 4$

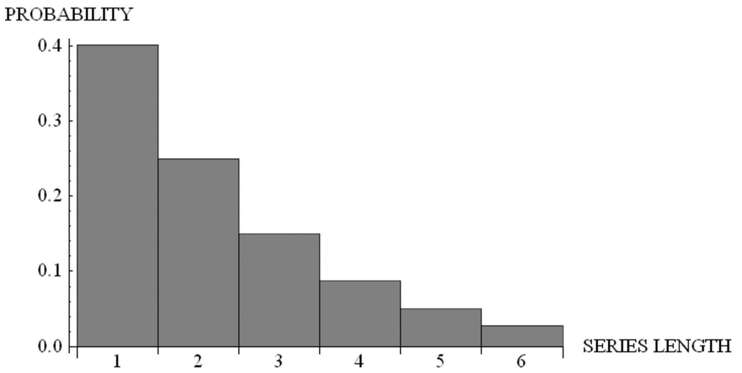


Fig. 2. Distribution of the first lost series' length for mean service time 2.0 and mean vacation time 0.5 and for $n = 0$

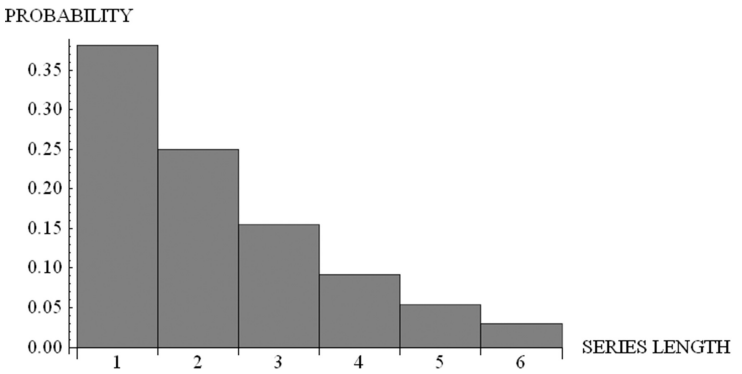


Fig. 3. Distribution of the first lost series' length for mean service time 2.0 and mean vacation time 2.0 and for $n = 4$

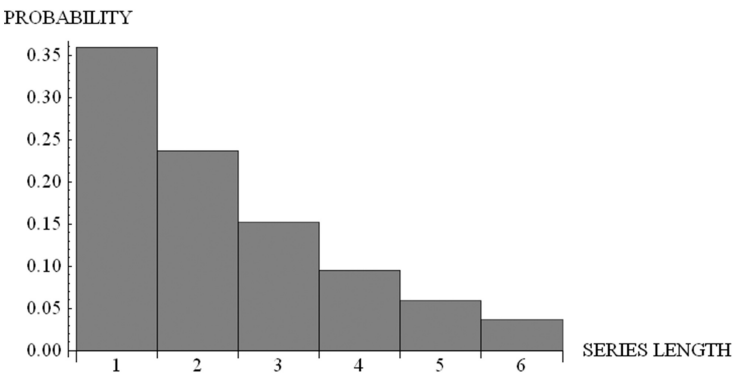


Fig. 4. Distribution of the first lost series' length for mean service time 2.0 and mean vacation time 2.0 and for $n = 0$

In Figs. 3 and 4 similar visualizations are given for the case of long vacation period with mean 2.0.

7 Conclusions

In the paper a finite-capacity queueing model with Poisson arrivals and generally distributed processing times is investigated. Every time when the service station becomes idle a single vacation period is being started, during which the processing is suspended. Using the analytical approach based on the idea of embedded Markov chain, integral equations and linear algebra, a system of equations for the probability distribution of the length of the first loss series is built and solved. The corresponding result for next series of losses is hence obtained. Numerical utility of analytical formulae is shown in computational examples. The considered queueing model has potential application in the analysis and performance evaluation of computer and telecommunication networks, and manufacturing lines.

References

1. de Boer, P.T., Nicola, V.F., van Ommeren, J.C.W.: The remaining service time upon reaching a high level in M/G/1 queues. Queueing Syst. **39**, 55–78 (2001)
2. Chae, K.C., Kim, K., Kim, N.K.: Remarks on the remaining service time upon reaching a target level in the M/G/1 queue. Oper. Res. Lett. **35**, 308–310 (2007)
3. Chydziński, A.: On the remaining service time upon reaching a target level in M/G/1 queues. Queueing Syst. **47**, 71–80 (2004)
4. Fakinos, D.: The expected remaining service time in a single server queue. Oper. Res. **30**, 1014–1018 (1982)
5. Kempa, W.M.: On the distribution of the time to buffer overflow in a queueing system with a general-type input stream. In: 35th International Conference on Telecommunication and Signal Processing (TSP 2012), pp. 207–211, Piscataway, Prague, Czech Republic, 3–4 July 2012
6. Kempa, W.M.: On transient queue-size distribution in the batch-arrivals system with a single vacation policy. Kybernetika **50**, 126–141 (2014)
7. Kempa, W.M.: On queueing delay in WSN with energy saving mechanism based on queued wake up. In: Mustra, M., et al. (eds.) Proceedings of 21st International Conference on Systems, Signals and Image Processing (IWSSIP 2014), pp. 187–190, Zagreb, Dubrovnik, Croatia, 12–15 May 2014
8. Kempa, W.M.: Time-dependent analysis of transmission process in a wireless sensor network with energy saving mechanism based on threshold waking up. In: IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2015), pp. 26–30, Piscataway, Stockholm, Sweden, 28 June–1 July 2015
9. Kempa, W.M.: Transient workload distribution in the M/G/1 finite-buffer queue with single and multiple vacations. Ann. Oper. Res. **239**, 381–400 (2016)

10. Kempa, W.M., Paprocka, I.: Time to buffer overflow in a finite-capacity queueing model with setup and closedown times. In: Świątek, J., Wilimowska, Z., Borzemski, L., Grzech, A. (eds.) Proceedings of 37th International Conference on Information Systems Architecture and Technology (ISAT 2016), Part 3. Advances in Intelligent Systems and Computing, vol. 523, pp. 215–224. Springer, Cham (2017)
11. Korolyuk, V.S.: Boundary-value Problems for Compound Poisson Processes. Naukova Dumka, Kiev (1975)
12. Woźniak, M., Kempa, W.M., Gabryel, M., Nowicki, R.: A finite-buffer queue with a single vacation policy: an analytical study with evolutionary positioning. Int. J. Appl. Math. Comput. Sci. **24**, 887–900 (2014)

Nash Equilibrium of Capacity Allocation Game for Autonomic Multi-domain Software Defined Networks

Dariusz Gąsior(✉)

Faculty of Computer Science and Management,
Wroclaw University of Science and Technology, Wroclaw, Poland
dariusz.gasior@pwr.edu.pl

Abstract. The software defined networking concept has attracted much attention recently. Decoupling of the data and control planes is believed to enable efficient network control and high resource utilisation. Especially, it is noteworthy in the context of the new applications which require not only quality of service constraints to be met but also the high quality of experience perceived by users. In this paper, we consider the capacity allocation problem in multi-domain software defined networks (SDN). We assume that each domain is managed by independent SDN controller. Thus, we formulate the considered problem as a Capacity Allocation Game and we find its Nash Equilibrium. Such an approach allows us to implement the elaborated algorithm in the automatic manner consistent with autonomic networking paradigm. The initial simulation experiments, which have been performed for small networks, indicate that the proposed method is very promising one.

Keywords: SDN · Game theory · Quality of service · Quality of experience · Utility

1 Introduction

For the contemporary applications, it is substantial to guarantee appropriate values of transmission parameters. It makes Quality of Service (QoS) crucial and vital for modern networks. The QoS requirements concern delay, jitter, loss probability, and transmission rate. Ensuring adequate values of these parameters depends on the network resources allocation algorithms.

There are two main concepts of QoS network architectures, namely integrated services, and differentiated services [21]. The first idea consists in ensuring QoS requirements for each individual flow. It makes this approach hardly scalable and impractical to implement in existing large-scale systems. The latter approach involves aggregation of the flows in traffic classes. However, while the resources are reserved for the particular class, flows have no individual QoS guarantees. As for now, none of these concepts are fully satisfactory.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_20

While fulfilling the QoS requirements makes the application useful, the allocation of the additional resources allows improving the quality perceived by the user. We call it Quality of Experience (QoE). There are many approaches to model the end-user QoE [2,20]. One of the considered ideas is to describe user QoE in relation to the QoS parameters using the utility function [5,12,13].

The software defined networking seems to be the promising network architecture. The basis of this concept lays in decoupling control plane from data plane [17,19]. While the packet forwarding tasks are executed in simplified network devices, all the necessary calculations are made by central SDN controller. It is also assumed that SDN architecture enables efficient per-flow behaviors. A comprehensive survey of current approaches to QoS in software defined networks may be found in [11]. Nevertheless, most of the works concern the single domain networks. Some results for multi-domain SDNs may be found e.g. in [10,18,22]. However, the coordination mechanisms between domains are required by the solutions presented in all aforementioned works.

In this paper, we propose to apply autonomic networking paradigm which enables self-managing [8] to solve the capacity allocation problem in the multi-domain networks. Meanwhile, the necessity of simultaneous negatiations between SDN controllers is eliminated. We assume that each SDN controller wants to maximize the QoE perceived by users whose transmission flows traverse corresponding domain while QoS requirements for all transmissions are met. We assume that the admission control mechanism exists (it may be implemented in SDN controllers) and it ensures that QoS requirements may be met for all accepted transmission demands. Moreover, we assume that the transmission routes are given in advance. We also limit the QoS parameters under consideration to the transmission rate, while it is believed to be the most crucial one [23]. The fulfilling other QoS requirements may be reached with other network mechanisms (e.g. appropriate routing algorithms, scheduling algorithms on nodes, etc.) We formulate the considered problem in terms of the game theory and we introduce the capacity allocation algorithm which finds a pure Nash Equilibrium of the proposed game. Finally, we present simulation results of the introduced solution method.

2 Mathematical Model and Problem Formulation

The multi-domain software defined network may be treated as a set of D interconnected subnetworks (domains). It is assumed that each domain is managed independently by a different SDN controller. Each subnetwork consists of nodes and links. Links represent connections between nodes inside domain as well as between domains. There are L links in the whole network. Each link l is characterized by its capacity C_l. This parameter reflects the maximal amount of data which may be sent between nodes in the unit time. The data exchanged between a pair of nodes is called a transmission or a flow. It is assumed that there are R transmissions in the whole network. The sequence of links used for a transmission is a route. It is assumed that only one route is available for each

transmission. The variable a_{rl} indicates if l link is used for rth transmission. The capacity allocation x_{rl} is made on each link l traversed by the rth flow. The capacity allocations are calculated in each domain independently by its SDN controller. The rth transmission's rate $\overline{x}_r$ results from the capacity allocations made on all links along its route. This rate must fulfill the QoS requirements expressed in terms of the minimum acceptable value $x_{r,\min}$. On the other hand, the links' capacities are the limited network resources and the total allocations for all transmissions must not exceed them. We also assume that there is a utility function $f(\overline{x}_r; w_r)$ associated with every transmission r which reflects , i.e. satisfaction level perceived when the transmission rate is $\overline{x}_r$ and the rth flow priority is w_r. The aim of each SDN controller is to maximize the total quality of experience (total utility) from all transmissions traversing corresponding domain (more precisely, traversing any link of this domain). The summary of notation is given in Table 1. The example topology of such a multi-domain SDN is given in Fig. 1 while the architecture of such a system is depicted in Fig. 2.

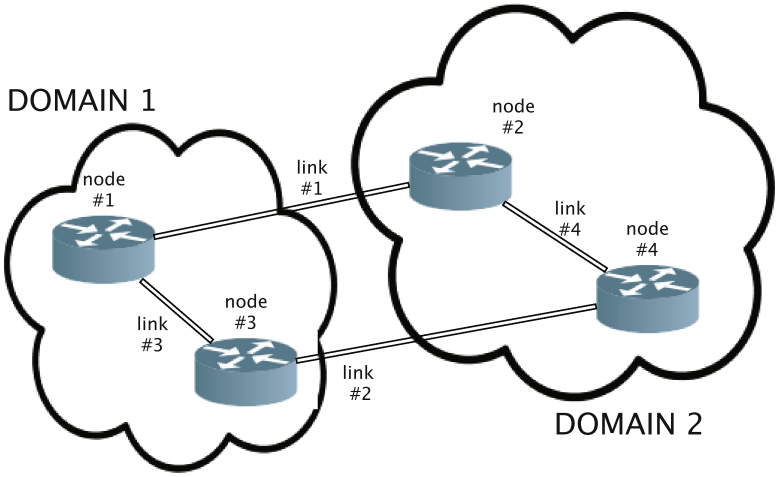


Fig. 1. The example topology of multi-domain SDN network.

This problem may be formulated as a noncooperative game [16]. The SDN controllers are players (there are D players - one for each domain since each domain is managed by one SDN controller). The feasible capacity allocation for transmissions traversing particular domain $\mathbf{x}_d$ is a strategy. The allocation matrix $\mathbf{x}$ is a strategy profile. The objectives $Q_d(\mathbf{x}_d, \mathbf{x}_{-d})$ are the players' payoffs. We refer to this game as the Capacity Allocation Game (CAG).

Formally, each player d (SDN controller) solves the following optimization problem:

Given: $R, L, a_{rl}, e_{ld}^{(link)}, e_{rd}^{(req)}, C_l, w_r, \beta_r, \alpha$

Find:

$$\mathbf{x}_d^* = \arg\max_{\mathbf{x}_d} Q_d(\mathbf{x}_d, \mathbf{x}_{-d}) \tag{1}$$

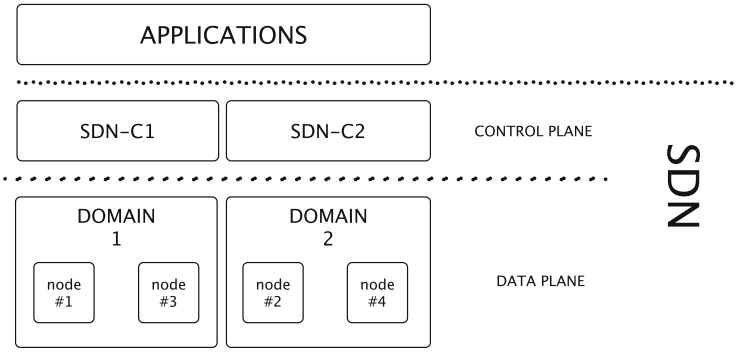


Fig. 2. The example architecture of multi-domain SDN network.

such that:

$$\forall_{l:e_{ld}^{(link)}=1} \quad \sum_{r=1}^{R} a_{rl}x_{rl} \leq C_l$$

$$\forall_{r=1,2,\ldots,R} \quad x_{rl} \geq x_{r,\min}$$

It is assumed that: $\forall_{l=1,2,\ldots,L} \sum_{r=1}^{R} a_{rl}x_{r,\min} \leq C_l$, i.e. there is an independent mechanism of the admission control. From the game theory perspective, the solution of such a problem is a game equilibrium.

3 Capacity Allocation Algorithm

First, we define the following auxiliary problem, which we refer to as In-domain Capacity Allocation Problem:
Given: $R, L, a_{rl}, e_{ld}^{(link)}, e_{rd}^{(req)}, C_l, w_r, \beta_r, \alpha, x_{r,\max}$
Find:

$$\mathbf{x}_d^* = \arg\max_{\mathbf{x}_d} \sum_{r=1}^{R} e_{rd}^{(req)} \beta_r f(\min_{l:e_{ld}^{(link)}=1 \,\wedge\, a_{rl}=1} x_{rl}; w_r) \tag{2}$$

such that:

$$\forall_{l:e_{ld}^{(link)}=1} \sum_{r=1}^{R} a_{rl}x_{rl} \leq \tilde{C}_l$$

$$\forall_{r=1,2,\ldots,R} \quad x_{r,\max} \geq x_{rl} \geq x_{r,\min}$$

$$\forall_{r=1,2,\ldots,R} \forall_{l_1,l_2 \in \{l:e_{ld}^{(link)}=1\}} \quad a_{rl_2}x_{rl_2} = a_{rl_1}x_{rl_1}$$

where $x_{r,\max}$ are given upper bounds of transmission rates.

Table 1. Notation

L	Number of links
R	Number of flows (transmissions)
D	Number of domains (subnetworks)
C_l	lth link's capacity ($l = 1, 2, \ldots, L$)
$\mathbf{a} = [a_{rl}]_{r=1,2,\ldots,R;l=1,2,\ldots,L}$	Routing matrix: $a_{rl} = 1$, rth flow traverse lth link $a_{rl} = 0$, otherwise
$e_{ld}^{(link)}$	Variable indicating if lth link belongs to the dth domain
$e_{rd}^{(req)}$	Variable indicating if rth transmission is performed by any link in dth domain $e_{rd}^{(req)} = \begin{cases} 1 & \text{gdy } \sum_{l=1}^{L} e_{ld}^{(link)} a_{rl} \geq 1 \\ 0 & \text{otherwise} \end{cases}$
w_r	rth flow priority parameter
$x_{r,\min}$	rth flow minimal transmission rate (QoS parameter)
$x_{rl} \geq 0$	lth link's capacity allocation for rth flow
$\hat{\mathbf{x}}_l$	lth link's capacity allocation vector: $\hat{\mathbf{x}}_l = [x_{rl}]_{r=1,2,\ldots,R}$
$\overline{x}_r \geq 0$	rth flow transmission rate: $\overline{x}_r = \min_{l:a_{rl}=1} x_{rl}$
$\mathbf{x}_d$	The allocation matrix for dth domain, $\mathbf{x}_d = [\hat{\mathbf{x}}_l]_{l:e_{ld}^{(link)}=1}$, $\mathbf{x} = [\mathbf{x}_d]_{d=1,2,\ldots,D}$
$\mathbf{x}_{-d}$	The allocation matrix $\mathbf{x}$ without dth component, it is assumed: $\mathbf{x} = [\mathbf{x}_d, \mathbf{x}_{-d}]$
$f(\overline{x}_r; w_r) = w_r \varphi(\overline{x}_r)$	rth flow utility function $\varphi(\overline{x}_r) = \begin{cases} \frac{\overline{x}_r^{(1-\alpha)}}{(1-\alpha)} & \alpha \geq 0 \ \wedge \ \alpha \neq 1 \\ \ln \overline{x}_r & \alpha = 1 \end{cases}$
$\beta_r \geq 0$	Given coefficient for rth flow e.g. $\beta_r = 1$ lub $\beta_r = (\sum_{l=1}^{L} a_{rl})^{-1}$
$Q_d(\mathbf{x}_d, \mathbf{x}_{-d})$	The objective of the dth domain (payoff): $Q_d(\mathbf{x}_d, \mathbf{x}_{-d}) = \sum_{r=1}^{R} e_{rd}^{(req)} \beta_r f(\overline{x}_r; w_r)$ $= \sum_{r=1}^{R} e_{rd}^{(req)} \beta_r w_r \varphi(\min_{l:a_{rl}=1} x_{rl})$

This is a convex optimization problem. It may be solved with methods given e.g. in [3].

One may note that for the given domain d, if we assume that allocations for all other domains have been determined and we put $x_{r,\max} = \min_{l:a_{rl}=1 \wedge e_{ld}^{(link)}=0} x_{rl}$, the solution obtained in (2) is the best response in the considered Capacity Allocation Game. Thus, we can use it to formulate allocation Algorithm 1 which provable stops at the Nash Equilibrium [15].

Algorithm 1. Multi-domain Utility-based Rate Allocation Algorithm (MURA)

1: Set $stopcondition \leftarrow False$ and $\forall_{r\in\{1,2,\ldots,R\}}$ $x_{\max,r} \leftarrow \infty$.
2: **while** not *stopcondition* **do**
3: $\forall_{d\in\{1,2,\ldots,D\}}$ Find optimal allocation $\mathbf{x}_d = [\mathbf{x}_l]_{l:e_{ld}^{(link)}}$ for dth domain for current values of parameters: $x_{\max,r}, r = 1, 2, \ldots, R$, i.e. solve problem (2) independently for each domain.
4: $\forall_{l\in\{1,2,\ldots,L\}}\forall_{r\in\{1,2,\ldots,R\}}$ Update: $x_{rl} \leftarrow \min_{j:a_{rj}=1} x_{rj}$
5: **if** $\forall_{r\in\{1,2,\ldots,R\}:\sum_{d=1}^{D} e_{rd}^{(req)}>1}$ $x_{r,\max} = \min_{j:a_{rj}=1} x_{rj}$ **then**
6: $stopcondition \leftarrow True$
7: **end if**
8: $\forall_{r\in\{1,2,\ldots,R\}:\sum_{d=1}^{D} e_{rd}^{(req)}>1}$ Update: $x_{\max,r} \leftarrow \min_{j:a_{rj}=1} x_{rj}$.
9: **end while**
10: Return $\mathbf{x}$.

The following properties of the Algorithm 1 occur.

Theorem 1. *If one set any other initial value of* $x_{r,\max}$ *in a step 1 such that* $x_{r,\max} \geq \min_{l:a_{rl}=1} C_l$ *then the Algorithm 1 finds a pure Nash Equilibrium.*

It means, one may replace ∞ in a step 1 with any finite non-negative value and Algorithm 1 still finds pure Nash Equilibrium. The Theorem 1 results from the following facts:

1. The condition in the step 5 guarantees that allocation for rth transmission are equal in every link l.
2. Consider any transmission flow which traverses more than one domain. Increasing its capacity allocation cannot increase its rate, so the objective function cannot increase.
3. In the last iteration of the Algorithm 1 the optimal solution of problem (2) is found for each domain. So, the objective of any domain cannot be increased by changing any transmission rate (even for flows which traverse only one domain).

In consequence, each strategy found in the final iteration of the Algorithm 1 is the best response to all other strategies. Thus, the obtained strategy profile is a Nash Equilibrium.

Remark 1. If each domain consists of one node and there is no QoS requirements (i.e. $\forall_r x_{r,\min} = 0$) then Algorithm 1 is equivalent to the Iterated Allocation Algorithm presented in [8].

Therefore, Algorithm 1 may also find Pareto-optimal equilibrium in some special cases.

Further remarks are limited only to cases when $\alpha \in (0, 1)$. Let us introduce concepts of Social Welfare, Price of Anarchy (PoA) and Price of Stability (PoS) according to [14]. First, let us define the following Social Welfare function:

$$SW(\mathbf{x}) \triangleq \sum_{r=1}^{R} f(\overline{x}_r; w_r)$$

Now, we can define the Price of Anarchy (PoA) as:

$$PoA = \frac{SW^*}{\min_{\mathbf{x} \in SNE} SW(\mathbf{x})}$$

where: SNE - set of all pure Nash Equilibria, $SW^* = \max_{\mathbf{x} \in \hat{D}_{\mathbf{x}}} SW(\mathbf{x})$, $\hat{D}_{\mathbf{x}}$ - set of all feasible strategy profiles.
Finally, we can also define Price of Stability (PoS) as:

$$PoS = \frac{SW^*}{\max_{\mathbf{x} \in SNE} SW(\mathbf{x})}$$

The PoA measures the maximal deterioration of the social welfare when each player makes decision independently (non-cooperative game) in comparison to the situation when all players act for common good (maximal social welfare). On the other hand, PoS measures the minimal deterioration of the social welfare.

Remark 2. In some instances of the Capacity Allocation Game, Price of Anarchy (PoA) is unbounded.

This remark results from the Theorem 1. It is enough to consider the problem with no QoS requirements. The solution obtained with the Algorithm 1 when $x_{r,\max}$ is set to 0 in the step 1 constitutes the Nash Equilibrium. The value of social welfare ($SW(\mathbf{x})$) for such strategy profile is 0. So, PoA tends to infinity.

Remark 3. In some instances of the Capacity Allocation Game, Price of Stability (PoS) is equal to 1.

To justify this remark, it is enough to consider a network without any transmission traversing more than one domain and to assume that $\forall_r \beta_r = 1$. Then, each optimizing problem (1) is independent and all solutions of these problems constitute strategy profile for which $SW(\mathbf{x})$ is maximal.

4 Simulation

Some preliminary simulation experiments have been performed. The main objective was to initially evaluate the elaborated capacity allocation algorithm. The algorithm was implemented using Python 2.7 programming language and the experiments were conducted using the computer with Intel Core M 1.1 processor and 8 GB RAM.

The simulations was run for $\beta_r = 1$ and the following problem parameters:

- network's parameters:
  - $N \in \{7, 8, 9, 10\}$ - number of networks,
  - $\rho = 0.4$ - links' density: $L = \frac{1}{2}\rho N(N-1)$,
  - $D \in \{2, 3, 4, 5\}$ - number of domains, (at least one link was assigned to each domain),
  - $C_l \sim U(1000, 10000)$ - capacity of lth link [Mbps].
- transmissions' parameters:
  - $R \sim U(\frac{1}{2}\underline{\theta}N(N-1); \frac{1}{2}\overline{\theta}N(N-1))$ - number of transmissions, $\underline{\theta} = 0.3, \overline{\theta} = 0.7$,
  - for each transmission r the source and origin nodes are randomly chosen (i, j),

- a_{rl} - calculated using shortest path algorithm (from origin node to destination node),
- $w_r \sim U(1, 10)$
- $\alpha = 0.5$

The uniform distribution on the interval $[c, d]$ is denoted by $U(c, d)$. The network topologies were generated with the method described in [4]. The simulation was run three times for each set of parameter values (N, D). The considered network instances correspond to the small real-life backbone networks [1].

The quality criterion is defined as follows: $\gamma = \frac{SW(\mathbf{x}^{MURA})}{\max_{\mathbf{x} \in \hat{D}_{\mathbf{x}}} SW(\mathbf{x})}$, where $\mathbf{x}^{MURA}$ is the solution found by the proposed algorithm. One may notice that $PoS \leq \frac{1}{\gamma} \leq PoA$. The optimal value of the social welfare function $SW(\mathbf{x})$ was found with primal-dual projected gradient method.

The statistical results of the simulation experiments are given in Table 2. The representative chart of the average value of quality criterion for a different number of network nodes and different number of network domains is given in Fig. 3.

Table 2. Simulation results

Statistics for γ	MURA
Average	0.83
Median	0.83
Variance	0.01
Min	0.56
Max	1.0

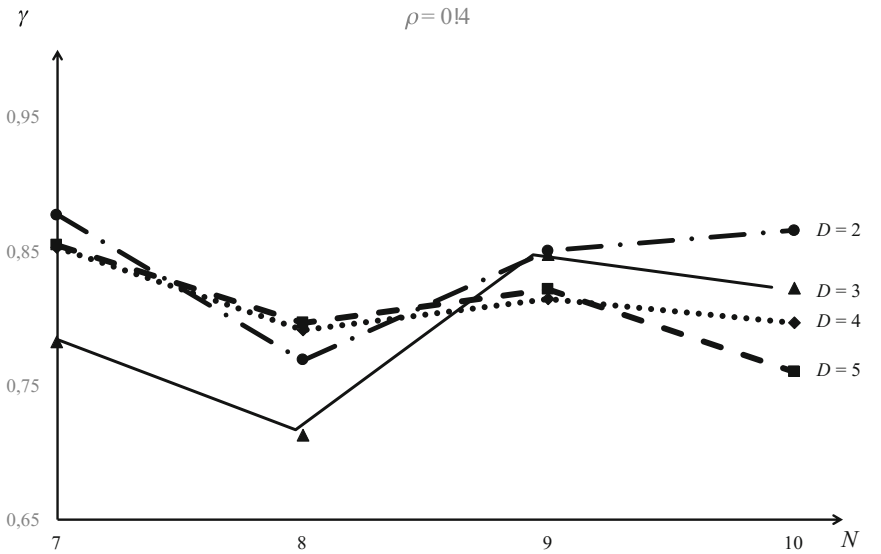


Fig. 3. The average value of quality criterion depending on number of nodes for different number of network domains.

One may notice that the Algorithm 1 gives the average solution which deviates no more than 20% from the socially optimal (i.e. optimal in terms of the social welfare). In the worst case, this deviation is not greater than 50%, while in best cases one may obtain a socially optimal solution (i.e. $\gamma = 1$). Nevertheless, the results (including those presented in Fig. 3) indicates that there is no strict dependency between solution quality and problem parameters (such as a number of network nodes or a number of network domains).

5 Final Remarks

In this paper, the capacity allocation problem with the quality of experience and quality of service requirements for autonomic multi-domain software defined networks was considered. The problem was formulated in the terms of the game theory and the algorithm which finds Nash Equilibrium was presented. Introduced solution method allows SDN controllers to act independently, so no communication and no negotiations are required. The initial experiments indicate that the proposed approach is very promising one. However, the experiments for larger networks are needed, so the more practical conclusions can be made. The future works may also include the application of the presented approach to the non-IP software defined networks like content-aware SDN (CASDN) as an extension of the ideas presented in [9] and to the virtual software defined networks as an extension of the ideas presented in [6,7].

References

1. Reference networks. http://www.av.it.pt/anp/on/refnet2.html. Accessed 19 July 2017
2. Alreshoodi, M., Woods, J.: Survey on QoE QoS correlation models for multimedia services. Int. J. Distrib. Parallel Syst. **4**(3), 53 (2013)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Bu, T., Towsley, D.: On distinguishing between internet power law topology generators. In: Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, INFOCOM 2002, vol. 2, pp. 638–647. IEEE (2002)
5. Gasior, D.: QoS rate allocation in computer networks under uncertainty. Kybernetes **37**(5), 693–712 (2008)
6. Gasior, D.: Capacity allocation in multilevel virtual networks under uncertainty. In: 2012 XVth International Telecommunications Network Strategy and Planning Symposium (NETWORKS), pp. 1–6. IEEE (2012)
7. Gasior, D.: Game-theoretical approach to capacity allocation in self-managed virtual networks. In: Proceedings of 36th International Conference on Information Systems Architecture and Technology–ISAT 2015–Part II Information Systems Architecture and Technology, pp. 155–164. Springer, Heidelberg (2016)
8. Gasior, D., Drwal, M.: Pareto-optimal nash equilibrium in capacity allocation game for self-managed networks. Comput. Netw. **57**(14), 2817–2832 (2013)

9. Gasior, D., Drwal, M.: Caching and capacity allocation game in self-managed content provider networks. In: 2014 16th International Telecommunications Network Strategy and Planning Symposium (Networks), pp. 1–7. IEEE (2014)
10. Helebrandt, P.: Architecture for core networks utilizing software defined networking. Inf. Sci. Technol. **8**(2), 56 (2016)
11. Karakus, M., Durresi, A.: Quality of Service (QoS) in software defined networking (SDN): a survey. J. Netw. Comput. Appl. **80**, 200–218 (2016)
12. Kelly, F.P., Maulloo, A.K., Tan, D.K.: Rate control for communication networks: shadow prices, proportional fairness and stability. J. Oper. Res. Soc. **49**(3), 237–252 (1998)
13. Khan, M.A., Toseef, U.: User utility function as Quality of Experience (QoE). Proc. ICN **11**, 99–104 (2011)
14. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. Comput. Sci. Rev. **3**(2), 65–69 (2009)
15. Nash, J.: Non-cooperative games. Ann. Math. **54**(2), 286–295 (1951)
16. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
17. Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutorials **16**(3), 1617–1634 (2014)
18. Phemius, K., Bouet, M., Leguay, J.: Disco: distributed multi-domain SDN controllers. In: 2014 IEEE Network Operations and Management Symposium (NOMS), pp. 1–4. IEEE (2014)
19. Shin, M.K., Nam, K.H., Kim, H.J.: Software-defined networking (SDN): a reference architecture and open APIs. In: 2012 International Conference on ICT Convergence (ICTC), pp. 360–361. IEEE (2012)
20. Stankiewicz, R., Jajszczyk, A.: A survey of QoE assurance in converged networks. Comput. Netw. **55**(7), 1459–1473 (2011)
21. Tanenbaum, A.S., Wetherall, D.J.: Computer Networks. Prentice Hall, Upper Saddle River (2013)
22. Vilalta, R., Mayoral, A., Muñoz, R., Casellas, R., Martínez, R.: Hierarchical SDN orchestration for multi-technology multi-domain networks with hierarchical ABNO. In: 2015 European Conference on Optical Communication (ECOC), pp. 1–3. IEEE (2015)
23. Wydrowski, B., Zukerman, M.: Qos in best-effort networks. IEEE Commun. Mag. **40**(12), 44–49 (2002)

Queueing Delay in a Finite-Buffer Model with Failures and Bernoulli Feedback

Wojciech M. Kempa(✉)

Faculty of Applied Mathematics, Institute of Mathematics,
Silesian University of Technology, ul. Kaszubska 2A, 44-100 Gliwice, Poland
wojciech.kempa@polsl.pl

Abstract. In the paper a finite-capacity queueing model with server breakdowns is considered. Jobs arrive according to a Poisson process and are being served during exponentially distributed processing time under the FIFO service discipline. During the operation of the system successive exponential failure-free times are followed by generally distributed repair periods. After the processing a job may rejoin the queue (feedback) with probability q or leave the system with probability $1 - q$. Applying the analytical approach based on the idea of Markov chain and the formula of total probability, a system of integral equations for the transient queueing delay distribution is built, conditioned by the initial buffer state. Algebraic method based on the idea of Korolyuk's potential is used to obtain the solution of the corresponding system written for Laplace transforms in a closed form. Numerical example is attached as well.

Keywords: Bernoulli feedback · Finite-Buffer queue · Integral equations · Queueing delay · Transient state

1 Introduction

Wide applications of finite-buffer queueing models in modelling and performance evaluation of telecommunication and computer networks, in automated manufacturing lines, transport and logistic problems are obvious. In the analysis of the appropriate queueing model some important stochastic characteristics, like queueing delay, queue-size distribution or time to buffer overflow, describing the operation of the system, are very helpful. As it seems, particularly important is the knowledge of the behavior of these characteristics in the non-stationary state of the system (i.e. at a fixed time epoch t), not only in the stable state. The time-dependent investigation of the system behavior is essential if we are interested e.g. in the analysis of system's evolution just after the initial moment or after introducing a new control mechanism (see e.g. [3]).

In practice, queueing systems with different-type limitation in service process are often used. In this class the model with server failures and Bernoulli feedback is an important one (see [2, 15]). A single-server model with Poisson arrivals and unreliable service station is considered e.g. in [1] and generalized for the multi-channel case in [14]. In [12] an infinite-buffer M/M/1-type queueing system with server subject to breakdowns is investigated. Some models with server failures are also considered e.g.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_21

in [5, 6, 13]. In [4] a queueing model with unreliable server and negative jobs (disasters), the arrivals of which cause losing of some or all customers present in the system at the arrival epoch, is analyzed.

In the paper we study a single-server queueing model with finite buffer and server breakdowns, in which successive inter-arrival, service and failure-free times are independent exponential random variables, while repair periods have general-type distribution. Moreover, a Bernoulli feedback mechanism is implemented, in which the job after the service may rejoin the queue with a fixed probability. As it seems, such a mechanism can be successfully used in modelling of packet retransmission. Using the approach based on the memoryless property of exponential distribution, the formula of total probability, integral equations and linear algebra, we obtain the formula for Laplace transform of the virtual delay $d(t)$ distribution at time t.

Transient analysis of the virtual delay in general-type batch-arrival queueing model can be found e.g. in [7] (see also [8] for the actual waiting time) is discussed. In [9] (see also [10]) the case of the virtual waiting time in a finite-buffer queue with single and multiple vacation policy is analyzed.

2 Model Description and Methodology

2.1 Queueing System

We deal with a single-channel queueing system in which the stream of the arriving jobs (customers, calls, packets, etc.) is governed by a Poisson process with rate $\boldsymbol{\lambda}$. The processing is offered according to a FIFO service discipline and successive service times are independent and exponentially distributed with mean $\boldsymbol{\mu}^{-1}$. The service station is unreliable, i.e. successive failure-free times which are exponentially distributed with mean $\boldsymbol{\gamma}^{-1}$, are followed by generally distributed repair periods with cumulative distribution function (CDF in short) $\boldsymbol{F}(\cdot)$.

The capacity of the system equals $\boldsymbol{N}$, i.e. we have a buffer with $\boldsymbol{N}-1$ places and one place "in processing". A typical Tail Drop procedure is used in the case of buffer overflow: the incoming job that finds the system in state $\boldsymbol{N}$ cannot be buffered and is lost without service. The buffer may contain a number of jobs accumulated before the start of the system. It is assumed that the service station is in order at the starting time t = 0 and can break down only being busy with processing of a job.

A mechanism of Bernoulli feedback is implemented in the considered system. Each job, independently on the others, can rejoin the queue after the service with probability $\boldsymbol{q} \in [\mathbf{0}, \mathbf{1}]$. Similarly, after the processing, a job leaves the system definitely with probability $\mathbf{1}-\boldsymbol{q}$.

Let us denote by $d(t)$ the queueing delay at time t, i.e. the waiting time of a "hypothetical" job arriving the system exactly at time t. The main goal of the theoretical analysis is to find a compact-form and numerically tractable formula for the tail of CDF of $d(t)$, in dependence on the initial level of buffer saturation, i.e. for the following stochastic characteristic:

$$D_n(t,x) \underline{\underline{\text{def}}} \mathbf{P}\{d(t) > x | X(0) = n\}, \tag{1}$$

where $\boldsymbol{t} > 0, x > 0, n \in \{\mathbf{0}, \mathbf{1}, \ldots, \boldsymbol{N}\}$ and $\boldsymbol{X}(\mathbf{0})$ stands for the number of jobs present in the system (buffered) at the starting time $\boldsymbol{t} = \mathbf{0}$. The formula will be derived in terms of Laplace transforms (LTs in short).

2.2 Analytical Approach

In the analysis we use the analytical approach that can be presented as a certain step-by-step procedure. We start by defining a set of random events describing the behavior of queueing delay distribution in the case of the system being empty at the opening. Considering also the situation of the buffer containing a number of jobs before the starting moment, we next build a system of integral equations for $D_n(t, x)$, where $n = 0, 1, \ldots, N$.

In the construction of the system we use the continuous version of the total probability law and a renewal-theory approach, determining the Markov moments in the evolution of the system. The original system of equation we change into the corresponding form written for appropriate LTs and write it in a specific form. Finally, we obtain the solution of the latter system by using the idea of potential of a random walk. Namely, the solution is written in a closed form utilizing an additional sequence defined by coefficients of the system.

3 Equations for Queueing Delay Distribution

3.1 Integral Equations for $D_n(t,x)$ at Fixed Time t

Let us start with considering the case of the system being empty at the starting epoch $t = 0$. If we denote by y the first arrival moment after this time, we may write the following obvious integral equation:

$$D_0(t,x) = \lambda \int_0^t e^{-\lambda y} D_1(t-y,x)dy. \tag{2}$$

Assume now that at least one job is accumulated in the buffer before the opening of the system, i.e. $n \in \{1, 2, \ldots, N\}$.

Note that, due to exponential probability distributions of inter-arrival, service and failure-free times, the moments of occurrences of arrivals, service completions (departures) and machine (service station) failures are Markov moments in the evolution of the system. To utilize this fact, let us define the following random events:

(1) A_1 – the first arrival occurs before the moment t and precedes the first departure epoch and the first failure time;
(2) A_2 – the first departure occurs before the moment t and precedes the first arrival epoch and the first failure time;

(3) A_3 – the first machine failure (breakdown) occurs before the moment t and precedes the first arrival and departure epochs;
(4) A_4 – before time t nor neither an arrival nor departure nor failure moment occurs.

Evidently, the following equation is true:

$$D_n(t,x) = \sum_{j=1}^{4} \mathbf{P}\big\{d(t) > x, A_j | X(0) = n\big\}. \tag{3}$$

For the buffer being not saturated at the opening, i.e. $n \in \{1, 2, \ldots, N-1\}$, it is easy to check that, just from the definitions of random events A_j, the following relationships hold:

$$\mathbf{P}\{d(t) > x, A_1 | X(0) = n\} = \lambda \int_0^t e^{-(\lambda+\mu+\gamma)y} D_{n+1}(t-y,x)dy; \tag{4}$$

$$\mathbf{P}\{d(t) > x, A_2 | X(0) = n\} = \mu \int_0^t e^{-(\lambda+\mu+\gamma)y} [(1-q)D_{n-1}(t-y,x) + qD_n(t-y,x)]dy; \tag{5}$$

$$\begin{aligned}
&\mathbf{P}\{d(t) > x, A_3 | X(0) = n\} \\
&= \gamma \int_{y=0}^{t} e^{-(\lambda+\mu+\gamma)y} dy \Bigg\{ \int_{z=0}^{t-y} \Bigg[\sum_{j=0}^{N-n-1} \frac{(\lambda z)^j}{j!} e^{-\lambda z} D_{n+j}(t-y-z,x) \\
&\quad + D_N(t-y-z,x) \sum_{j=N-n}^{\infty} \frac{(\lambda z)^j}{j!} e^{-\lambda z} \Bigg] dF(z). \\
&\quad + \int_{z=t-y}^{\infty} \sum_{j=0}^{N-n-1} \frac{[\lambda(t-y)]^j}{j!} e^{-\lambda(t-y)-\mu(x-y-z+t)} \sum_{k=0}^{n+j-1} \frac{[\mu(x-y-z+t)]^k}{k!} dF(z) \Bigg\}
\end{aligned} \tag{6}$$

and, finally,

$$\mathbf{P}\{d(t) > x, A_4 | X(0) = n\} = e^{-(\lambda+\mu+\gamma)t-\mu x} \sum_{j=0}^{n-1} \frac{(\mu x)^j}{j!}. \tag{7}$$

Let us comment Eqs. (4)–(7) briefly. Indeed, in (4) the first arrival occurs before time t, and before the first departure and failure moment, exactly at time denoted by y. Hence, the evolution of the system "restarts" in Markov moment y with $n+1$ jobs present. In (5), at the Markov moment y corresponding to the first departure epoch, the system "renews" with $n-1$ jobs present with probability $1-q$ (no feedback) and with n jobs with probability q (feedback). The representation (6) describes the case in which a breakdown precedes the first arrival and the first departure. The first summand on the right side of (6) relates to the situation in which a repair is also completed before

time t and during the repair period the buffer does not become saturated. In the second summand on the right side of (6) the system "restarts" the operation after the repair period with the maximal number of N jobs present. In the last summand the repair period finishes after time t. The formula (8) is evident: since before time t neither arrival nor departure nor breakdown occurs, the probability that the queueing delay at this time exceeds x equals to the probability that at most $n-1$ of accumulated jobs will be served during time period of length x.

For the system that starts the operation with the buffer being saturated ($n = N$), analogically, the following equations can be written:

$$\mathbf{P}\{d(t) > x, A_1 | X(0) = N\} = \lambda \int_0^t e^{-(\lambda+\mu+\gamma)y} D_N(t-y,x)dy; \tag{8}$$

$$\mathbf{P}\{d(t) > x, A_2 | X(0) = N\} = \mu \int_0^t e^{-(\lambda+\mu+\gamma)y}[(1-q)D_{N-1}(t-y,x) + qD_N(t-y,x)]dy; \tag{9}$$

$$\mathbf{P}\{d(t) > x, A_3 | X(0) = N\} = \gamma \int_{y=0}^t e^{-(\lambda+\mu+\gamma)y} dy \int_{z=0}^{t-y} D_N(t-y-z,x)dF(z) \tag{10}$$

and

$$\mathbf{P}\{d(t) > x, A_4 | X(0) = N\} = 0. \tag{11}$$

3.2 Corresponding System for Laplace Transforms

In this subsection we construct a system of equations, corresponding to (2), (4)–(7) and (8)–(11), written for LTs of conditional distributions $\boldsymbol{D_n(t,x)}$.

Define

$$d_n(s,x) \underline{\underline{\text{def}}} \int_0^\infty e^{-st} D_n(t,x)dt, \tag{12}$$

where $\Re(s) > 0$ and $n \in \{0, 1, 2, \ldots, N\}$, and introduce, additionally, the following functions:

$$f_k(\lambda, s) \underline{\underline{\text{def}}} \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-(\lambda+s)t} dF(t), \tag{14}$$

$$H(s,x) \underline{\underline{\text{def}}} e^{-\mu x} \int_{y=0}^\infty e^{\mu y} dF(y) \sum_{j=0}^{N-n-1} \int_{z=0}^y \frac{(\lambda z)^j}{j!} e^{-(\lambda+s+\mu)z} \sum_{k=0}^{n+j-1} \frac{[\mu(x-y+z)]^k}{k!} dz. \tag{15}$$

Due to the fact that the following representations are true:

$$\gamma \int_{t=0}^{\infty} e^{-st}dt \int_{y=0}^{t} e^{-(\lambda+\mu+\gamma)y}dy \int_{z=0}^{t-y} \frac{(\lambda z)^k}{k!} e^{-\lambda z} D_j(t-y-z,x)dF(z) = \gamma(\lambda+\mu+\gamma+s)^{-1} f_k(\lambda,s)d_j(s,x), \tag{16}$$

$$\gamma \int_{t=0}^{\infty} e^{-st}dt \int_{y=0}^{t} e^{-(\lambda+\mu+\gamma)y}dy \int_{z=0}^{t-y} D_N(t-y-z,x)dF(z) = \gamma(\lambda+\mu+\gamma+s)^{-1} f_0(0,s)d_N(s,x) \tag{17}$$

and

$$\gamma \int_{t=0}^{\infty} e^{-st}dt \int_{y=0}^{t} e^{-(\lambda+\mu+\gamma)y}dy \int_{z=t-y}^{\infty} \sum_{j=0}^{N-n-1} \frac{[\lambda(t-y)]^j}{j!} e^{-\lambda(t-y)-\mu(x-y-z+t)} \sum_{k=0}^{n+j-1} \frac{[\mu(x-y-z+t)]^k}{k!} dF(z) = \gamma(\lambda+\mu+\gamma+s)^{-1} H(s,x), \tag{18}$$

then we obtain the following system of equations written for LTs $d_n(s,x), n \in \{0,1,\ldots,N\}$:

$$d_0(s,x) = \lambda(\lambda+s)^{-1} d_1(s,x); \tag{19}$$

$$d_n(s,x) = \frac{1}{\lambda+\mu+\gamma+s}\Big[\lambda d_{n+1}(s,x) + \mu((1-q)d_{n-1}(s,x) + q d_n(s,x)) + \gamma\left(\sum_{k=0}^{N-n-1} d_{n+k}(s,x)\alpha_k(s) + d_N(s,x)\sum_{k=N-n}^{\infty} f_k(\lambda,s) + H(s,x)\right) + e^{-\mu x}\sum_{j=0}^{n-1}\frac{(\mu x)^j}{j!}\Big] \tag{20}$$

and

$$d_N(s,x)[\mu(1-q)+s+\gamma(1-f_0(0,s))+s] = \mu(1-q)d_{N-1}(s,x). \tag{21}$$

4 Main Result

Let us notice that introducing the following functions and substitution:

$$a_0(s) \underline{\underline{\text{def}}} \frac{\mu(1-q)}{\lambda+\mu(1-q)+\gamma+s}; a_{k+1}(s) \underline{\underline{\text{def}}} \frac{1}{\lambda+\mu(1-q)+\gamma+s}\left[\lambda\delta_{k,1}+\gamma\alpha_k(s)\right]; \tag{22}$$

$$\chi_k(s,x) \underline{\underline{\text{def}}} -\frac{1}{\lambda+\mu(1-q)+\gamma+s}\left[\gamma\left(\eta_0(s,x)\sum_{i=k}^{\infty}\alpha_i(s)+H(s,x)\right)+e^{-\mu x}\sum_{j=0}^{N-k-1}\frac{(\mu x)^j}{j!}\right] \quad (23)$$
$$\underline{\underline{\text{def}}}\ \Gamma_k(s)\eta_0(s,x)+\Delta_k(s,x),$$

where $\delta_{i,j}$ is Kronecker delta function,

$$\Gamma_k(s) \underline{\underline{\text{def}}} -\frac{\gamma}{\lambda+\mu(1-q)+\gamma+s}\sum_{i=k}^{\infty}\alpha_i(s), \quad (24)$$

$$\Delta_k(s,x) \underline{\underline{\text{def}}} -\frac{1}{\lambda+\mu(1-q)+\gamma+s}\left[\gamma H(s,x)+e^{-\mu x}\sum_{j=0}^{N-k-1}\frac{(\mu x)^j}{j!}\right], \quad (25)$$

and

$$\eta_{\mathrm{n}}(s,x) \underline{\underline{\text{def}}}\, d_{N-n}(s,x), \quad (26)$$

The equations of the system (19)–(21) can be rewritten in the following form:

$$\eta_0(s,x)[\mu(1-q)+s+\gamma(1-f_0(0,s))+s]=\mu(1-q)\eta_1(s,x), \quad (27)$$

$$\sum_{\boldsymbol{k=-1}}^{\boldsymbol{n}}\boldsymbol{\alpha_{k+1}(s)\eta_{n-k}(s,x)}-\boldsymbol{\eta_n(s,x)}=\boldsymbol{\chi_k(s,x)}, \quad (28)$$

where $\mathbf{1}\le\mathbf{n}\le\mathbf{N}-\mathbf{1}$, **and**

$$\eta_N(s,x)=\lambda(\lambda+s)^{-1}\eta_{N-1}(s,x). \quad (29)$$

In [11] the following system of equations is considered:

$$\sum_{k=-1}^{n}\alpha_{k+1}y_{n-k}-y_n=\chi_k, \quad (30)$$

where $n \ge 1$ and $(\alpha_k), a_0\neq 0, (\chi_k)$ are two known sequences. It is proved in [11] that each solution of (30) can be given in the form ($n \ge 1$):

$$x_n=MR_n+\sum_{k=1}^{n}R_{n-k}\chi_k, \quad (31)$$

in which M is a constant and the sequence (R_n) is connected with the sequence of coefficients (α_k) by the following equation:

$$\sum_{k=0}^{\infty}R_kz^k=\frac{z}{z-A(z)}, \quad (32)$$

where $A(z) \underline{\underline{\text{def}}} \sum_{k=0}^{\infty}\alpha_k z^k, |z|<1.$

Evidently, the system (28) has the same form as (30) but with functions (a_k) and (χ_k) depending on s, and s and x, respectively. In consequence (see (31)) M can be in general dependent on s and x, so we obtain for $n \geq 1$

$$\eta_n(s,x) = M(s,x)R_n(s) + \sum_{k=1}^{n} R_{n-k}(s)\chi_k(s,x), \tag{33}$$

where $\chi_k(s,x)$ was defined in (23). Obviously, we need the formulae for $M(s, x)$ and $\eta_0(s, x)$, since (33) is valid for $n \geq 1$ only.

Taking $n = 1$ in (33), we get

$$\eta_1(s,x) = M(s,x)R_1(s). \tag{34}$$

Next, defining

$$G(s) \underline{\underline{\text{def}}} [\mu(1-q)R_1(s)]^{-1}[\mu(1-q) + \gamma(1 - f_0(0,s)) + s], \tag{35}$$

and referring to (27) and (34), we obtain

$$M(s,x) = G(s)\eta_0(s,x). \tag{36}$$

Substituting now $n = N$ into (33) and using (23), we have

$$\eta_N(s,x) = G(s)R_N(s)\eta_0(s,x) + \sum_{k=1}^{N} R_{N-k}(s)[\Gamma_k(s)\eta_0(s,x) + \Delta_k(s,x)]. \tag{37}$$

From the other side, applying (33) in (29), we get

$$\eta_N(s,x) = \lambda(\lambda+s)^{-1}\left[G(s)\eta_0(s,x)R_{N-1}(s) + \sum_{k=1}^{N-1} R_{N-1-k}(s)(\Gamma_k(s)\eta_0(s,x) + \Delta_k(s,x))\right]. \tag{38}$$

Comparing the right sides of (37) and (38), we eliminate $\eta_0(s, x)$ as follows:

$$\eta_0(s,x) = \left[\sum_{k=1}^{N-1}\left(\lambda(\lambda+s)^{-1}R_{N-1-k}(s) - R_{N-k}(s)\right)\Delta_k(s,x)\right] \Big/ \left[G(s)\left(R_N(s) - \lambda(\lambda+s)^{-1}R_{N-1-k}(s)\right) - \sum_{k=1}^{N-1}\left(\lambda(\lambda+s)^{-1}R_{N-1-k}(s) - R_{N-k}(s)\right)\Gamma_k(s,x)\right]. \tag{39}$$

Now, from the representations (26), (33), (36) and (39) we obtain the following main result:

Theorem. The Laplace transform $d_n(s, x)$ of the tail of CDF of queueing delay in the considered queueing system with failures and Bernoulli feedback is given by the following formula:

$$d_n(s,x) = \left[G(s)R_{N-n}(s) + \sum_{k=1}^{N-n} R_{N-n-k}(s)\Gamma_k(s)\right]\eta_0(s,x) + \sum_{k=1}^{N} R_{N-k}(s)\Delta_k(s,x), \tag{40}$$

where $Re(s) > 0, 0 \le n \le N$, *and the formulae for* $\Gamma_k(s,x), \Delta_k(s), G(s)$ *and* $\eta_0(s,x)$ *are found in (24), (25), (35) and (39), respectively. Moreover (compare (32))*

$$\sum_{k=0}^{\infty} R_k(s)z^k = \frac{z}{z - A(s,z)}, \tag{41}$$

where $A(s,z) = \sum_{k=0}^{\infty} \alpha_k(s)z^k, |z| < 1$, *and* $\alpha_k(s)$ *is defined in (22).*

References

1. Avi-Itzhak, B., Naor, P.: Some queueing problems with the server station subject to breakdown. Oper. Res. **11**, 303–320 (1963)
2. van den Berg, J.L., Boxma, O.J., Groenendijk, W.P.: Sojourn times in the M/G/1 queue with deterministic feedback. Commun. Stat. Stochast. Models **5**(1), 115–129 (1989)
3. Bertsimas, D.J., Nakazato, D.: Transient and busy period analysis of the GI/G/1 queue: the method of stages. Queueing Syst. **10**(3), 153–184 (1992)
4. Boxma, O.J., Perry, D., Stadje, W.: Clearing models for M/G/1 queues. Queueing Syst. **38**, 287–306 (2001)
5. Gray, W.J., Wang, P.P., Scott, M.K.: A vacation queueing model with server breakdowns. Math. Model. **24**, 391–400 (2000)
6. Ke, J.C.: An M/G/1 queue under hysteretic vacation policy with an early startup and unreliable server. Math. Methods Oper. Res. **63**, 357–369 (2006)
7. Kempa, W.M.: The virtual waiting time for the batch arrival queueing systems. Stochast. Anal. Appl. **22**(3), 1235–1255 (2004)
8. Kempa, W.M.: Some results for the actual waiting time in batch arrival queueing systems. Stochast. Models **26**(3), 335–356 (2010)
9. Kempa, W.M.: Transient workload distribution in the M/G/1 finite-buffer queue with single and multiple vacations. Ann. Oper. Res. **239**(2), 381–400 (2016)
10. Kempa, W.M.: A comprehensive study on the queue-size distribution in a finite-buffer system with a general independent input flow. Perform. Eval. **107**, 1–15 (2017)
11. Korolyuk, V.S.: Boundary-Value Problems for Compound Poisson Processes. Naukova Dumka, Kiev (1975)
12. Lam, Y., Zhang, Y.L., Liu, Q.: A geometric process model for M/M/1 queueing system with a repairable service station. Eur. J. Oper. Res. **168**, 100–121 (2006)

13. Madan, K.C.: A M/G/1 type queue with time-homogeneous breakdowns and deterministic repair times. Soochow J. Math. **29**(1), 103–110 (2003)
14. Neuts, M.F., Lucantoni, D.M.: A Markovian queue with N servers subject to breakdowns and repairs. Manag. Sci. **25**, 849–861 (1979)
15. Rege, K.: On the M/G/1 queue with Bernoulli feedback. Oper. Res. Lett. **14**(3), 163–170 (1993)

Tuning Energy Effort and Execution Time of Application Software

Thomas Rauber[1(✉)] and Gudula Rünger[2]

[1] University of Bayreuth, Bayreuth, Germany
rauber@uni-bayreuth.de
[2] Chemnitz University of Technology, Chemnitz, Germany
ruenger@informatik.tu-chemnitz.de

Abstract. Software products are usually required to meet some static or dynamic properties. Well-known examples of dynamic properties are the program execution time and the related goal of software performance optimization. Because of the increasing importance of ecological and environmental issues, also the energy consumption of software products is a dynamic property of increasing importance. Modern computer systems already provide features, such as multicores and voltage-frequency scaling, to support the reduction of the energy consumption of software. However, a low program execution time and a good energy efficiency might be conflicting goals and it may be difficult so simultaneously reduce the program execution time and the energy consumption. In this article, the relation between energy effort and execution time of software is investigated and a software tuning method for task-based programs is proposed, which appraises different program versions and different task structures concerning their execution time and energy consumption with the objective to pick the most favorable solution.

Keywords: Software tuning · Energy effort · Execution time · Task-based programs

1 Introduction

Energy-aware computing and the efficient use of computing resources are now accepted to be as important for software products as performance aspects or issues such as reliability and security. Energy efficiency has already been a critical concern in digital circuit design for the last two decades, which has caused several power-aware system features, including multicore-on-a-chip processors, core-independent functional units, dynamic voltage and frequency scaling (DVFS), or clock gating. These techniques are mainly aimed at a reduction of the energy consumption of the processors. Other techniques also address other compute devices. To fully exploit the hardware capabilities for reducing the energy consumption of software running on that hardware, there have to be programming

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_22

methods that are aware of the energy consumed. Energy models and measures are needed to plan and evaluate the time and energy behavior of the software on specific hardware. This article considers this problem of evaluating the behavior of program execution for the case of task-based programs.

The design of application software code can influence the energy consumption of program execution. This can be done by either decreasing the execution time, which may result in a smaller energy consumption, or by scaling the frequency to decrease the power consumption, which may however increase the execution time. Since a low execution time is still an important goal, for instance when simulating large applications or when a predefined response time is required, energy-awareness and performance awareness have to be treated together resulting in a multidimensional design space for the software. Typical approaches are to minimize the execution time while guaranteeing an upper bound for the energy consumption or to minimize the energy consumption while guaranteeing an upper bound for the execution time. This can be done by solving optimization problems in a planning and tuning phase before software execution as proposed in this article.

The power consumption and the execution time of an application belong to the dynamic properties, which can only be measured and evaluated exactly at runtime. For the planning and tuning phase before execution, appropriate prediction methods for execution time and energy consumption are required which mimic the performance behavior by some metrics. Well-known metrics for the measurements of performance are floating point operations per second (flop/s) or the time needed to solve a specific problem (time-to-solution). Concerning the power or the energy, the metric for power efficiency measured in performance per Watt (flop/s/w) and the amount of energy needed to solve a specific problem (energy-to-solution) are used. The energy-delay product (EDP) combines the time-to-solution and the energy-to-solution and, thus, constitutes an appropriate metrics to evaluate the execution time and energy consumption together. The metrics time-to-solution and the energy-to-solution are considered in this article.

In this article, we propose a software planning and tuning process which selects a program with a minimum execution time with or without energy constraint or vice versa. The underlying programming model considers programs built up of independent sequential tasks and to be executed on a multicore or other shared memory hardware. The planning and tuning is done before execution time and its goal is to pick a program version from a multitude of semantically equivalent program versions in this task model. The program versions are created by different assignments of the tasks to the cores or processors and by selecting different operational frequencies for those cores or processors. The assignment process is guided by cost information for the single sequential tasks, which can be execution time (measured or predicted) or any other suitable metrics. The resulting overall execution time can be adapted to improve the parallel execution time or to meet any other optimization goal. As a running example, we consider a frequency scaling adaptation process.

The contribution of the article is to propose a software tuning methodology that supports the software developer in the tuning process before program execution. The advantage is that an expensive redesign and reimplementation of software can be avoided, since a violation of energy or performance constraints can be detected and clarified in the early product development steps and not only in a posteriori measurement and analyses phase. These design steps are included in a learning phase which is typically iterating over the assignment and adaption steps and may include a practical test tun. The final result is a program structure which is ready for efficient production runs.

The rest of the article is structured as follows: Sect. 2 summarizes the task-based programming approach considered. Section 3 presents the energy model for software. Section 4 addresses the resulting optimization problem as well as the task scheduling for multicore systems. Section 5 presents the software architecture of the planning and tuning tool and discusses the workflow for the software planning. Section 6 discusses related work and Sect. 7 concludes the article.

2 Task-Based Software Construction

Task-based programming is a popular parallel programming model supported by multiple operating systems, programming environments or languages. Examples are OpenMP [1], the TPL library for .NET [2], Cilk [3] and Charm++ [4]. A task is a basic unit of the program comprising a distinct amount of work to be executed. In a task-based programming model, an entire application program is decomposed into these smaller units and the entirety of the executions to be done is represented by a set $\mathcal{T}$ of tasks. Tasks may depend on each other due to some result data produced by one task and needed by another tasks or vice versa. But tasks can also be entirely independent of each other which gives rise to a parallel execution on different execution units of a multicore or of another parallel hardware platform. Such a set of independent tasks can originate from one application but also from different applications.

Task parallelism exploits the task structure for a faster execution and is sometimes also called control parallelism, in contrast to data parallelism, or even functional parallelism, since function calls are often used as tasks. Some programming environments exploiting task parallelism support parallel execution on different cores or processors by different threads which are pinned to cores and processors. Since a thread assignment to execution units is often a one-to-one mapping, we use thread and execution unit in an interchangeable way. An important advantage of tasks parallelism is load balancing leading to an overall small execution time by keeping all execution units (or threads assigned to them) busy while guaranteeing small idle times. The quality of the load balance achieved depends on the scheduler and the scheduling algorithms assigning the task to threads or execution unit. Schedulers are either part of the operating systems or are included in a higher level of the software stack, e.g. a programming library such as OpenMP [1], or tools proposed for task parallelism, such as Tlib [5].

In the following, we consider a set $\mathcal{T} = \{t_1, \ldots, t_n\}$ of tasks which are to be scheduled and assigned to a set of execution units p_i, $i = 0, \ldots, p_{max} - 1$. It is assumed that the tasks are independent and that their input data are available on any execution unit p_i at any time due to a shared memory architecture, e.g. a multicore. Thus, the tasks $t \in \mathcal{T}$ can be executed in any order and a scheduling algorithm has no constraints concerning a task to be ready for execution so that a task can be assigned to any execution unit p_i at any time.

To perform a good assignment of tasks, a schedule needs information about the cost for each tasks. Costs can represent the execution time needed on an execution unit. As execution units, we use multicore processors with DVFS. A processor with DVFS can run on a lower frequency with the effect that the processor is slower and that the execution time for a tasks is higher, but might consume less energy.

In the following, we consider DVFS systems with p_{max} cores and operational frequencies f ranging between a minimum frequency f_{min} and a maximum frequency f_{max}. The frequency can only be changed in discrete steps. Frequency scaling for DVFS processors can be expressed by a dimensionless scaling factor $s \geq 1$ which describes a smaller frequency $f \leq f_{max}$ relative to the maximum possible frequency f_{max} as $f = f_{max}/s$. The execution time of task $t \in \mathcal{T}$ on a processor using frequency f corresponding to scaling factor s is denoted as $T_t(s)$. The execution time of the entire subset of task assigned to a thread or execution unit p processing with a frequency with scaling factor s is then: $\sum_{\text{t assigned to p}} T_t(s)$.

3 Energy Model

In addition to the execution time, we will also consider the energy consumption of a program execution. The energy consumed for the execution of an application program results from the execution time $T(s)$[sec] of the application and the power $P(s)$[Watt] consumed during that time. To model power and energy consumption, we use a well-accepted energy model that has already been applied to embedded systems [6], to heterogeneous computing systems [7], or to shared-memory architectures [8]. This section summarizes the energy model according to [6].

The power consumption of a processor consists of the *dynamic* power consumption P_{dyn}, which is related to the switching activity and the supply voltage, and the static power consumption P_{static}, which captures the leakage power consumption as well as the power consumption of peripheral devices, such as the I/O subsystem [9]. The dynamic power consumption is approximated by $P_{dyn} = \alpha \cdot C_L \cdot V^2 \cdot f$ where α is the switching probability, C_L is the load capacitance, V is the supply voltage, and f is the operational frequency. The frequency f depends linearly on the supply voltage V, i.e., it is $V = \beta \cdot f$ with some constant β. Reducing the frequency by a scaling factor s, i.e., using a modified frequency value $\tilde{f} = s^{-1} \cdot f$ with $s \geq 1$, leads to a decrease of the dynamic power consumption, which results in the following equation for the dynamic power consumption:

$\tilde{P}_{dyn}(s) = \alpha \cdot C_L \cdot V^2 \cdot \tilde{f} = s^{-3} \cdot P_{dyn}(1)$ [Watt], where $P_{dyn}(1)$ corresponds to the dynamic power consumption of an un-scaled system with $s = 1$.

The leakage power consists of several components, including sub-threshold leakage, reverse-biased-junction leakage, gate-induced-drain leakage, gate-oxide leakage, gate-current leakage, and punch-through leakage [10]. An approximation has been proposed by [11], who model the static power consumption due to leakage power as $P_{static} = V \cdot N \cdot k_{\text{design}} \cdot I_{\text{leak}}$ where V is the supply voltage, N is the number of transistors, k_{design} is a design dependent parameter, and I_{leak} is a technology-dependent parameter. Other authors have also proposed to make the simplified assumption that P_{static} is independent of the voltage or frequency scaling [6]. This is justified by the close match between the data sheet curves of real DVFS processors and the analytical curves obtained by using this assumption [12].

The power consumption of a task $t \in \mathcal{T}$ is then $P_t(s) = P_{dyn}(s) + P_{static}$. Using the scaling factor s as parameter, the energy consumption of a task $t \in \mathcal{T}$ on one processing unit is $E_t(s) = P_t(s) \cdot T_t(s) = (P_{dyn}(s) + P_{static}) \cdot T_t(s)$ [Joule]. The energy consumption for the execution of several tasks on the same processor p is then computed by adding the energy needed to compute all the tasks, i.e., $\sum_{\text{t assigned to p}} E_t(s)$.

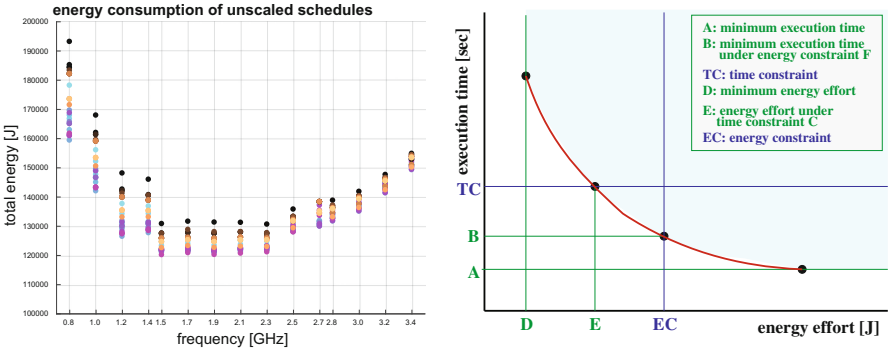


Fig. 1. Left: Distribution of the energy consumption resulting from different task assignments using different operational frequencies. **Right:** Illustration of the resulting execution time achieved for a specific energy effort.

4 Tuning Energy Effort and Execution Time

In this section, we consider the energy consumption and execution time of task-based application programs built up from a set of tasks.

4.1 Optimization Problems for Time and Energy

The parallel execution time of a parallel program $T_{par}(s)$ is usually defined as the time elapsed between the starting time when the first processor starts

its execution and the finishing time when the last active processor finishes its execution. The parameter s indicates that the execution time depends on the speed of the processors and, in general, the execution time is *smaller* for higher frequencies (meaning smaller values of s). When assumed that all processors start at the same time, the parallel execution time is the maximum of all execution times of the processors involved, which might mean that some processors have some idle times without useful work.

The parallel execution time is accompanied by an energy consumption $E_{par}(s)$ of these processors which comprises the energy of the active computation times and also the idle times in which only static power is consumed. When using a higher frequency (meaning smaller scaling factor s) the energy usually *increases*. Measurements have shown that there is non-linear dependence between frequency and energy consumed.

For parallel computing, a small execution time is usually a goal, which is nowadays accompanied by the aim to have a low energy consumption as well. Given that the behavior of execution time and energy consumption with respect to the scaling factor is quite the contrary, these are conflicting goals, which gives rise to several optimization problems concerning time or energy. When considering a parallel program in the task-based model with a set $\mathcal{T}$ of tasks executed on p_{max} processors, there is the additional degree of freedom of choosing a specific assignment of tasks to processors, which influences the parallel execution time due to possible idle times of different length and, thus, different energy consumption values may result. To capture the conflicting goals, we formulate the following different optimization problems:

Find an assignment $\mathcal{A} : \mathcal{T} \rightarrow \{p_0, \ldots, p_{max} - 1\}$ and a frequency s such that:

- (T-OPT): $T_{par}(s) \rightarrow$ minimum
- (T-OPT-C): $T_{par}(s) \rightarrow$ minimum under the constraint $E_{par}(s) \leq EC$
- (E-OPT): $E_{par}(s) \rightarrow$ minimum
- (E-OPT-C): $E_{par}(s) \rightarrow$ minimum under the constraint $T_{par}(s) \leq TC$

The determination of an optimal assignment $\mathcal{A}$ of tasks to processors is an NP-complete problem, which is typically solved by heuristics, such as scheduling algorithms.

Figure 1(left) shows the energy consumption resulting for different schedules for the same task-based program and using different frequencies (but the same for all processors). The schedules have been computed for the 26 SPEC CPU2006 benchmarks on eight processors using timings for an Intel Haswell processor over different frequencies. Each dot represents a different schedule. The smallest energy consumption is obtained for schedules between 1.5 GHz and 2.3 GHz. It can be seen that the results are quite different, yet they follow a u-shape like overall behavior.

Figure 1(right) illustrates the correlation between energy effort and resulting execution time. Investing more energy typically results in a reduction of the execution time. The red line shows the border in the energy-time phase space for all possible schedules and frequency adaptations. Point A represents the smallest

Algorithm 1. Scheduling algorithm with frequency adaptation.

```
1: procedure ScheduleTasks (TaskSet 𝒯)
2:     for (all frequencies available for DVFS) do
3:         s is the scaling factor for the selected frequency f
4:         Sort tasks {t_1, ..., t_n} in queue Q such that T_{t_1}(s) ≥ T_{t_2}(s) ≥ ... ≥ T_{t_n}(s);
5:         while (un-assigned tasks remain in Q) do
6:             if (worker thread W becomes idle) then
7:                 Remove next task from beginning of Q and assign it to W
8:             end if
9:         end while
10:        Compute makespan of resulting task assignment;
11:        Compute resulting overall energy consumption;
12:        Reduce energy consumption by frequency adaptation of individual processors;
13:    end for
14:    Compare and select best solution;
15: end procedure
```

execution time possible, point D the smallest energy consumption possible. Point B represents the smallest possible execution time for a given maximum energy budget EC. Point E represents the smallest possible energy consumption for a given maximum execution time TC not to be exceeded.

4.2 Task Assignment

Scheduling algorithms for collections of independent tasks are often based on list scheduling, see [13] for an extensive overview. In this work, we use such a list-based scheduling algorithm for the experiments, see Algorithm 1. In contrast to other scheduling algorithms, Algorithm 1 takes DVFS into consideration and investigates task timings for different frequencies, represented by their scaling factor. The tasks are sorted in decreasing order of their execution time for the frequency selected and are then assigned in a greedy way to the next processor that becomes idle. The resulting maximum execution time of all processors determines the so-called makespan of the schedule computed. The resulting energy consumption is the sum of the energy consumptions of the individual processors. After the computation of the schedule, which uses the same frequency for all processors, the algorithm investigates whether an adaptation of the frequencies of the individual processors may lead to a reduction of the overall energy consumption. If this is the case, the frequencies are adapted accordingly. Only those adaptations are considered that do not lead to an increase of the overall makespan.

4.3 Experimental Evaluation

The scheduling algorithm from the previous subsection can be applied to assign tasks to processors. For the tasks, different operational frequencies can be used.

Figure 2 shows the result of such an assignment of the 26 SPEC CPU2006 benchmarks, see [14], to eight processors using timings for an Intel Core i7-4770 with Haswell architecture. The schedules are generated for 15 different operational frequencies between 0.8 and 3.4 GHz, as they are provided by the Haswell processor. Each schedule can again be performed with different frequencies, yielding 225 possible combinations of energy effort and execution time. In Fig. 2(left), each dot represents a different schedule and the position of the dot denotes the energy effort required and the makespan obtained, using execution time and energy consumption values for tasks measured on the Intel Core i7 processor. The energy measurement technique is described in detail in [12]. For the schedules used for the data in Fig. 2(left), all processors are running at the same frequency and the corresponding frequencies are denoted in the diagram. Additionally, the frequencies of the processors can be adapted after the schedules have been computed, to hide possible idle times, which can lead to a further reduction of the energy effort. This is shown in Fig. 2(right), where the brown dots represent schedules with frequency adaptation for the individual processors. It can be seen that the frequency adaptation can lead to a significant decrease of the resulting energy consumption without affecting the overall makespan, see [15] for more information about the experiments. Only the most relevant area of Fig. 2(left) is captured in Fig. 2(right). The resulting lower left border line corresponds to the red border line in the illustration in Fig. 1(right). The experiments show that for the same set of tasks and the same architecture, a multitude of different solutions result from which the most favorable can be selected depending on the optimization problem.

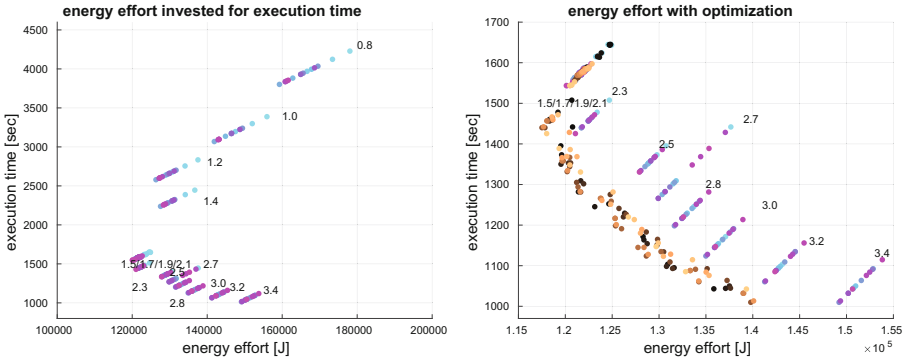


Fig. 2. Energy effort and resulting execution times: Schedules without (left) and with (right) frequency scaling for individual processors for SPEC benchmarks with Haswell timings.

5 Software Architecture for Optimization Environment

Figure 3 shows the coarse structure of the workflow for the software planning and tuning towards an efficient execution and a reduction of the overall energy consumption. The input program may be available in several task-based versions that may differ in the arrangement and granularity of the tasks. All these task versions perform the same computations and lead to the same output, but may differ in performance and energy consumption. A suitable program version is selected in the first step of the workflow. For this program version, a scheduling algorithm is selected, see [16], and is then used to compute a schedule for the given set of tasks. The assignment function $\mathcal{A}$ is computed and evaluated according to its resulting makespan and energy consumption. Based on this schedule and the DVFS capabilities of the given hardware platform, the frequencies of the individual processors can be adapted by a adaptation method to reduce the overall energy consumption. Different adaptation methods can be used, depending on the optimization goals described in Sect. 4.1. For example, the energy consumption can be reduced by adapting the frequencies of the different processors without increasing the overall makespan, but the adaption can also be done such that the makespan is allowed to increase by a certain amount, if this leads to a further reduction of the energy consumption. After the adaptation, the resulting schedule and DVFS setting is evaluated with respect to its actual performance by a theoretical analysis or by performance measurements. If the result is satisfactory, the tasks will be executed with the selected setting. Otherwise, the user can iterate over the decision process by either selecting a different adaptation method, a different scheduling algorithm, or a different task-based program version. These steps of the workflow can be considered as the learning phase of the program tuning. The final program version is then ready for efficient production runs of the software.

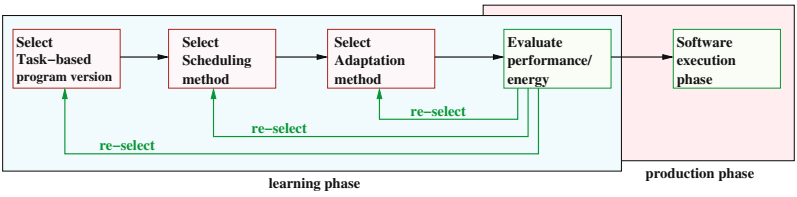


Fig. 3. Main structure of the workflow for time and energy tuning of task-based programs. The learning phase iterates over task decomposition, scheduling method, and adaptation until the final result is ready for the production phase of the software execution.

Figure 4 gives an overview of the corresponding software architecture implementing the planning and tuning methodology and the workflow from Fig. 3. The task-based program is analyzed by a task scheduler, which can use different scheduling algorithms to produce a schedule for the given hardware environment

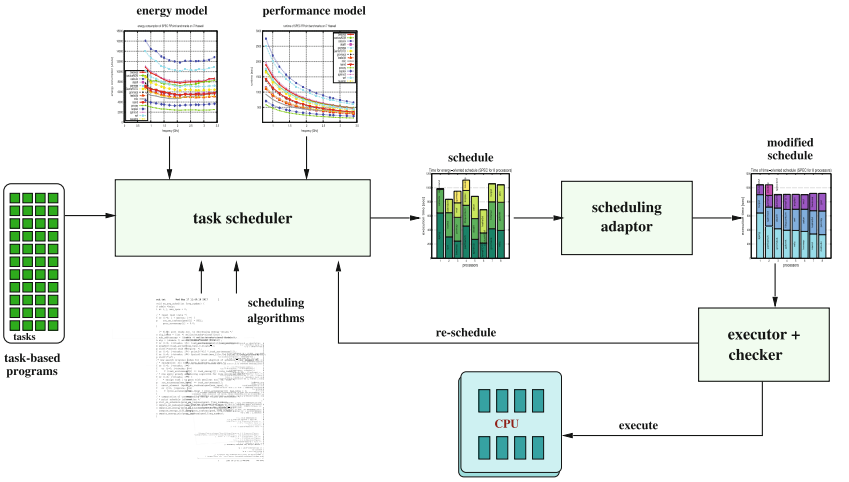


Fig. 4. Software architecture for the planning and tuning tool of energy-efficient software.

and number of execution units. The scheduling algorithm applied can be time-oriented, such as the one given in Algorithm 1, but can also be energy-oriented or mixed, using the energy-delay product. The task scheduler also uses information about the hardware of the execution platform, including information about the number of cores to be employed and the DVFS capabilities, as well as a performance model and an energy model for the specific behavior of the hardware. The performance and energy model allow the prediction of the execution time and resulting energy consumption of the tasks, see [16] for more information about the energy model. As an alternative, the performance and energy information could also be provided by measurements of the single tasks in isolation on the hardware platform used. This is especially suited for cases in which exact information is required and where the different tasks of the programs do not change their execution behavior between different program runs. The schedule produced by the task scheduler can be improved by a scheduling adaption according to a given optimization goal, such as a reduction of the energy consumption by a frequency adaptation of the individual processors, as described in Sect. 4.1. The result of the adaption is a potentially modified schedule that meets the optimization goal. The scheduling result is then checked, and it may be sent back to the task scheduler, if the result is not satisfying. If the schedule is accepted, the tasks are assigned according to the schedule to the hardware resources and the entire task program is executed according to the tuning decisions.

6 Related Work

Task-based approaches have been considered at several levels and with different programming models in mind. Language extensions have been proposed

to express the execution of tasks, including Fortran M, Opus, Braid, Orca, and Spar/Java. The HPCS language proposals, Sun's Fortress, IBM's X10, and Cray's Chapel, also contain some support for the specification of tasks. Moreover, skeleton-based approaches like P3L, LLC, Lithium, and SBASCO, as well as library-based and coordination-based approaches have been proposed. The usage of tasks for cloud computing services has been considered in [17].

Power-management features are integrated in computer systems of different size and class [18,19]. An important feature is the DVFS technique that trades off performance for power consumption by lowering the operating voltage and frequency if this is possible [6]. The approach to determine the voltage scaling factor that minimizes the total CPU energy consumption by taking both the dynamic power and the leakage power into consideration has been discussed in [6,9,20] for sequential tasks. Power models for multicore designs are developed in [19].

An important aspect of the DVFS technique is the integration into task scheduling algorithms, which traditionally have been developed with the goal to minimize the overall execution time. Integrating energy-saving goals increases the resulting complexity of the scheduling algorithms and many fundamental issues need to be addressed. Fundamental aspects of speed scaling and power-down scheduling are described in [21]. Theoretical aspects of state-of-the-art techniques of dynamic voltage scaling problems in the context of tasks with time constraints are considered in [22]. The integration of task scheduling and energy overhead considerations to minimize the energy consumption of real-time dependent tasks on systems with multiple execution resources is considered in [23]. An introduction to power measuring and profiling techniques is given in [24]. Both hardware-based power measurement and software-based power profiling and the corresponding power models are considered.

7 Conclusions

The goal to improve the overall energy efficiency of program executions needs a detailed knowledge of the application, the energy needed for its execution, and the energy saving techniques of the hardware available. We have proposed a software planning and tuning methodology which exploits a task-based software design method and hardware-oriented power and energy models resulting in a software coding of flexible energy-aware programs. Experiments have shown that the multitude of different choices can be reduced and how the methodology facilitates the selection of the best program version in the context of the programmer's optimization goal. Thus, the same task-based program can result in different parallel execution codes which compute the same algorithmic result, but may minimize the execution time with or without energy constraint, may minimize the energy with or without time constraint, or may minimize both time and energy based on a fused energy metric and a corresponding scheduling and tuning.

Acknowledgement. This work was performed within the Federal Cluster of Excellence EXC 1075 "MERGE Technologies for Multifunctional Lightweight Structures" and supported by the German Research Foundation (DFG). This work is also supported by the German Ministry of Science and Education (BMBF) under project number 01IH16012A/B. Financial support is gratefully acknowledged.

References

1. OpenMP Architecture Review Board: OpenMP Application Program Interface, Version 4.5. (2015)
2. Leijen, D., Schulte, W., Burckhardt, S.: The design of a task parallel library. In: Proceedings of the 24th ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications, OOPSLA 2009, pp. 227–242. ACM, New York (2009)
3. Blumofe, R., Joerg, C., Kuszmaul, B., Leiserson, C., Randall, K., Zhou, Y.: Cilk: an efficient multithreaded runtime system. J. Parallel Distrib. Comput. **37**(1), 55–69 (1996)
4. Kale, L., Bohm, E., Mendes, C., Wilmarth, T., Zheng, G.: Programming petascale applications with Charm++ and AMPI. In: Bader, D. (ed.) Petascale Computing: Algorithms and Applications, pp. 421–441. Chapman & Hall/CRC Press, Boca Raton (2008)
5. Rauber, T., Rünger, G.: Tlib - a library to support programming with hierarchical multi-processor tasks. J. Parallel Distrib. Comput. **65**(3), 347–360 (2005)
6. Zhuo, J., Chakrabarti, C.: Energy-efficient dynamic task scheduling algorithms for DVS systems. ACM Trans. Embed. Comput. Syst. **7**(2), 1–25 (2008)
7. Lee, Y., Zomaya, A.: minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, pp. 92–99. IEEE Computer Society (2009)
8. Korthikanti, V., Agha, G.: Towards optimizing energy costs of algorithms for shared memory architectures. In: Proceedings of the 22nd ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2010, pp. 157–165. ACM, New York (2010)
9. Jejurikar, R., Pereira, C., Gupta, R.: Leakage aware dynamic voltage scaling for real-time embedded systems. In: Proceedings of the 41st Annual Design Automation Conference, DAC 2004, pp. 275–280. ACM (2004)
10. Kaxiras, S., Martonosi, M.: Computer Architecture Techniques for Power-Efficiency. Morgan & Claypool Publishers, Seattle (2008)
11. Butts, J., Sohi, G.: A static power model for architects. In: Proceedings of the 33rd International Symposium on Microarchitecture (MICRO-33) (2000)
12. Rauber, T., Rünger, G., Schwind, M., Xu, H., Melzner, S.: Energy measurement, modeling, and prediction for processors with frequency scaling. J. Supercomput. **70**, 1451–1476 (2014)
13. Leung, J., Kelly, L., Anderson, J.: Handbook of Scheduling: Algorithms, Models, and Performance Analysis. CRC Press, Inc., Boca Raton (2004)
14. Henning, J.: SPEC CPU2006 benchmark descriptions. SIGARCH Comput. Archit. News **34**(4), 1–17 (2006)
15. Rauber, T., Rünger, G.: Comparison of time and energy oriented scheduling for task-based programs. In: Proceedings of 12th International Conference on Parallel Processing and Applied Mathematics. LNCS. Springer (2017)

16. Rauber, T., Rünger, G.: Modeling and analyzing the energy consumption of fork-join-based task parallel programs. Concurrency Comput. Pract. Exp. **27**(1), 211–236 (2015)
17. Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T., Epema, D.: Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Trans. Parallel Distrib. Syst. **22**(6), 931–945 (2011)
18. Saxe, E.: Power-efficient software. Commun. ACM **53**(2), 44–48 (2010)
19. Esmaeilzadeh, H., Blem, E., Amant, R., Sankaralingam, K., Burger, D.: Power challenges may end the multicore era. Commun. ACM **56**(2), 93–102 (2013)
20. Irani, S., Shukla, S., Gupta, R.: Algorithms for power savings. ACM Trans. Algorithms **3**(4), 41 (2007)
21. Chrobak, M.: Algorithmic aspects of energy-efficient computing. In: Ahmad, I., Ranka, S. (eds.) Handbook of Energy-Aware and Green Computing, pp. 311–329. CRC Press, London (2012)
22. Kim, T.: Power saving by task-level dynamic voltage scaling: a theoretical aspect. In: Ahmad, I., Ranka, S. (eds.) Handbook of Energy-Aware and Green Computing, pp. 361–383. CRC Press, London (2012)
23. Zhang, Y., Hu, X., Chen, D.: Energy minimization for multiprocessor systems executing real-time tasks. In: Ahmad, I., Ranka, S. (eds.) Handbook of Energy-Aware and Green Computing. CRC Press, London (2012)
24. Chen, H., Shi, W.: Power measurement and profiling. In: Ahmad, I., Ranka, S. (eds.) Handbook of Energy-Aware and Green Computing, pp. 649–674. CRC Press, London (2012)

An Approach to Semantics for UML Activities

Dariusz Gall(✉) and Anita Walkowiak

Wroclaw University of Science and Technology, Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland
{dariusz.gall,anita.walkowiak}@pwr.edu.pl

Abstract. The precise semantics for UML Activities is must-have in automated applications of them. The paper proposes an approach to a definition of a semantics of a set of UML Activities in the form of LTS (*Labelled Transition System*). The set of activities is transformed to a graph of possible execution traces, using OCL (*Object Constraint Language*) operations. So far, it covers basic elements of an activity, i.e. sequential control and object flows. Moreover, an Activity Semantics Metamodel is introduced. It specifies concepts, informally described in the UML, used for Activity semantics definition.

Keywords: UML Activity · OCL · LTS · Formal semantics · Reachability graph

1 Introduction

An activity is well-known behavior representation commonly called a "control and data flow" model [8, 12, 15]. There are intensively used for software requirements specification [3, p. 92, p. 477, 9, 10, 17, p. 105]. However, the main obstacle in using activity is lack of precise semantics. The definition of UML presented in the OMG standard [8] is semi-formal, i.e. the semantics of elements is expressed in natural language and has variation points. Additional challenges are the extensibility of the UML and (deliberate or accidental) under-specifications.

The lack of formal semantics brings ambiguity problems [5, 8, 11, 13, 16], especially in case of automation of system development process – i.e. transformation of PIM and PSM models and code generation in MDA approach; and design of tools supporting the process [3]. Furthermore, the possibility of UML model-checking is limited to syntax verification [1].

There are many works on the UML Activities semantics, e.g., [2, 3, 5, 12–16, 4]. These approaches specify the UML Activities semantics in some scope, omitting certain aspects of the standard and consider only specific usages of the UML Activities [2, 3, 5, 8] and/or are just outdated [4], since refer to previous versions of the UML [8]. Some of these works provide translational semantics, by expressing the UML Activities in other formalisms, e.g., Petri-Nets, Abstract State Machines. Nevertheless, it is challenging to formulate the Activity semantics in these formalism, especially when more UML Activities constructions are considered [14, 16]. Moreover, the authors investigate an execution of a single Activity and do not discuss an execution of a set of Activities.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_23

The main goal of the paper is to give an approach to a precise interpretation of a set of activities, consistent with the UML standard. The interpretation is expressed in terms of all possible execution traces that may be derived from the set. We propose a transformation of a set of activities into the Labelled Transition System [6], which describes the modeled behavior in the form of a graph of possible execution traces.

The proposed semantics will support automation of system development process. It will check compliance with end-users expectations, e.g. by model simulations and model execution. Next, it will support model checking, especially by using existing tools, like Construction and Analysis of Distributed Processes tool (CADP). Moreover, it is a base for test-scenarios generation. And last but not least, it is necessary to build and validate model transformations between PIM and PSM models, and code generation.

In Sect. 2, we discuss the syntax and semantics of the UML Activities presented in the UML specification [8]. We define the LTS-based semantics of the UML Activities in Sect. 3. It gives definitions that are necessary to introduce the LTS for the UML Activities, and finally we introduce the algorithm to construct a graph of possible execution traces. The paper is concluded in Sect. 4.

2 Syntax and Semantics of Activities

The activity defined by UML specification [8] is a graph. It is represented by the Activity metaclass which consists of ActivityNodes connected by ActivityEdges, stored in the *node* and the *edge* metaattributes of the Activity. The ActivityEdge connects two ActivityNodes via the *source* and the *target* metaattributes, respectively. Subclasses of the ActivityNode are the ExecutableNode, the ObjectNode, and the ControlNode. There are two types of the ActivityEdge: the ControlFlow and the ObjectFlow.

An Activity is a kind of a Behavior. The Behavior describes a set of possible executions, or more precisely it is "*a specification of events that may occur dynamically over time*" [13, p. 284]. The UML specification defines an Event as "*a set of possible occurrences*", whereas an occurrence is "*something that happens that has some consequence with regard to the system*" [13, p. 12]. An invocation of a Behavior creates its instance, known as a behavior execution. Each behavior execution is related to a specific execution trace, which is "*actual sequence of event occurrences due to the invocation [of the Behavior], consistent with the specification of the Behavior*" [13, p. 284].

An Activity appoints a partial order of behavior steps that may be executed and data flows corresponding to the steps. A behavior step lasts non-zero time, is expressed by ExecutableNode and represents, for example, an arithmetic computation, a call to an operation, or manipulation of object contents. The partial order of behavior steps execution is specified in terms of tokens flow rules inspired by Petri Nets formalism [14] – "*The effect of one ActivityNode on another is specified by the flow of tokens over the ActivityEdges between ActivityNodes*" [13, p. 372], whereas it is assumed that the flow, if it occurs, is immediate. There are two kind of tokens: a control and an object token. A control token flow specifies an effect of execution one ActivityNode on

another – set an order of the execution, and flows over ControlFlow edges. An object token is a container for a value that flows over ObjectFlow edges.

A behavior step is enabled to be executed when tokens are offered to it on incoming edges and specified conditions are met. When an execution of a behavioral step is started, "*tokens are accepted from some or all of its incoming* ActivityEdges *and a token is placed on the node*" [13, p. 373]. When a behavioral step completes an execution, "*a token is removed from the node and tokens are offered to some or all of its outgoing* ActivityEdges" [13, p. 373]. Thereby, we may distinguish following types of events occurring during the Activity execution:

- tokens acceptation by a node – e.g. a start of an ExecutableNode execution,
- tokens offering by a node – e.g. a finish of an ExecutableNode execution.

At the given point of an activity's execution, event types are disjoint for an activity node, which means that only one of them can be enabled.

An execution trace of an Activity is an actual sequence of tokens acceptances and/or offerings occurrences due to the invocation, consistent with the specification of the Activity. Set of all possible execution traces of the Activity is the semantic of the Activity.

In our approach we consider the semantics of a set of Activities with one indicated initial activity. The initial activity refers directly or indirectly to activities in the set by calling other activities (CallBehaviorAction), signal sending (SendSignalAction), etc.

3 Semantics of Activities

We introduce the metamodel representing execution traces of a set of activities (hereinafter referred to as "ASmetamodel"), (Fig. 1). We define the ASmetamodel in the context of UML Activity specification [8], thereby some of classes introduced to the ASmetamodel refer to UML Activity metaclasses, for example to the ActivityNode, the ActivityEdge. The ASmetaclasses are marked by «ASmetaclass» keyword.

The set of activities (instances of UML Activity metaclass) are represented by a set of Act activities. When an Act is invoked a new instance of this is created. The instance is represented by the Execution metaclass.

An execution trace of an Act activity is defined by indicating consecutive places of tokens within the considered Act. A place of a token in the Act is represented by the Location metaclass (Fig. 2). A token generally is represented by Token metaclass, which is specialized by the ControlToken and the ObjectToken metaclasses modelling a control token and object token, respectively. For the sake of the simplicity, an ObjectToken does not contain a value, however it states that a value is present at all. In the context of the Location, a token (referenced by *token* metaattribute) may be situated either at the *node* or the *edge* of the Act.

The Snapshot metaclass represents locations of all tokens at some point (snapshot) of an execution trace of an Act activity (Fig. 2). It corresponds to progress in the Act's behavior execution. A Snapshot refers to the instance of the Act by the *execution*, and to a set of tokens places by the *locations*.

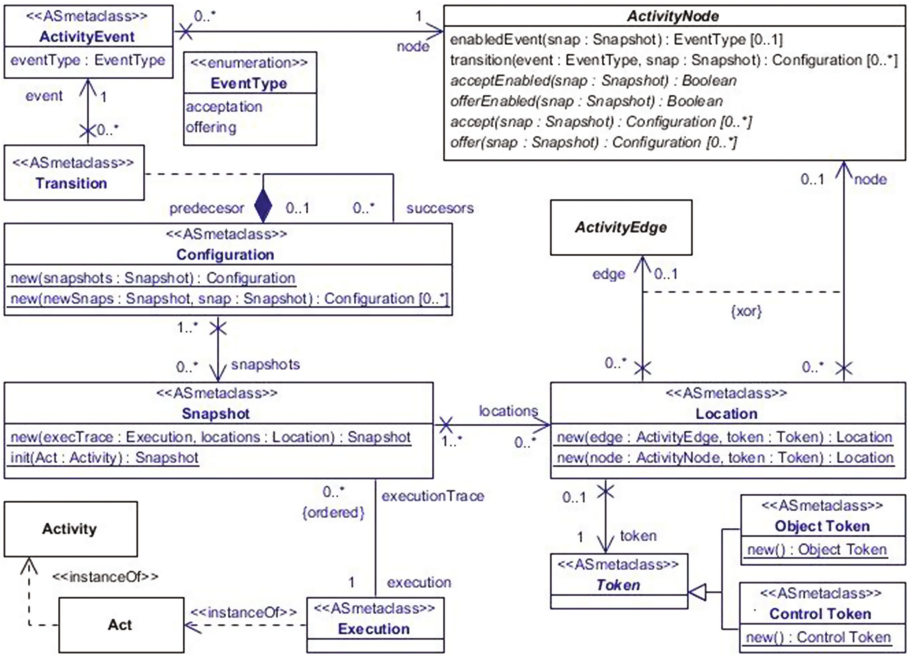


Fig. 1. Activity semantics metamodel.

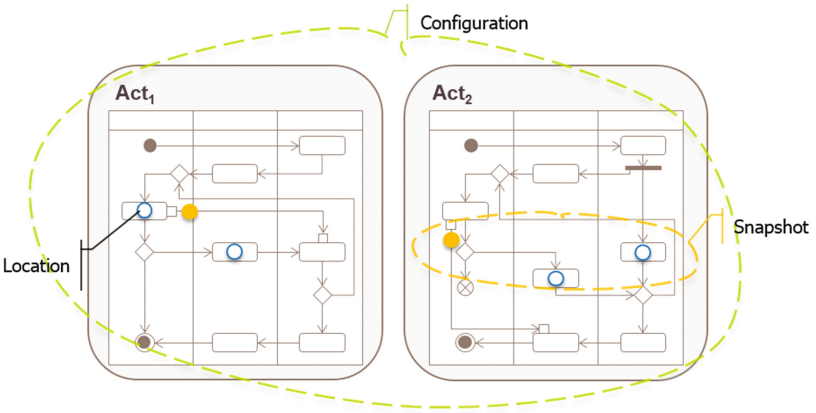


Fig. 2. Example configuration of a set of activities.

The Configuration metaclass represents places of tokens (indirectly via *snapshots*) after an event occurrence and before the very next event occurrence, for all instances of activities (Fig. 2). Or, if it is the Activities' invocation, it specifies initial places of tokens within the instances. The Configuration refers to progress in an execution of the set of Activities.

An event occurrence causing a transition from one Configuration to another is modelled by the Transition association metaclass. The event is specified by an ActivityEvent, which has an event type (*eventType*) and an activity node (*node*) which it refers to. An event type is one of the EventType values, i.e. either the acceptation value, for acceptation events, or the offering value, for offering events.

A Configuration refers to a *predecessor* configuration, unless it is the very first configuration, i.e. one from which it was directly transited.

There may be many successor configurations for a Configuration. Firstly, if there are many enabled events occurrences for the configuration, e.g., many ExecutableNodes in the activities are enabled to accept tokens (e.g., because of concurrent tokens flows), a new Transition is added for each event occurrence. Secondly, there may be many successor configurations, for a single event occurrence. For example, an ExecutableNode having optional outputs (OutputPin) may offer tokens at every subset of the outputs. Thereby, an event occurrence – tokens offering by the ExecutableNode – results in a set of possible successor configurations for each subset of the outputs. Similarly, there might be many possible ways of tokens acceptation, in example presented in Fig 3. The action within the snapshot (a.) can accept a control token from two alternative locations, and similarly can accept an object token (via input pin) from two alternative locations. There are four possible cases (two of them are presented in the Fig. 3 by snapshots (b.) and (c.)).

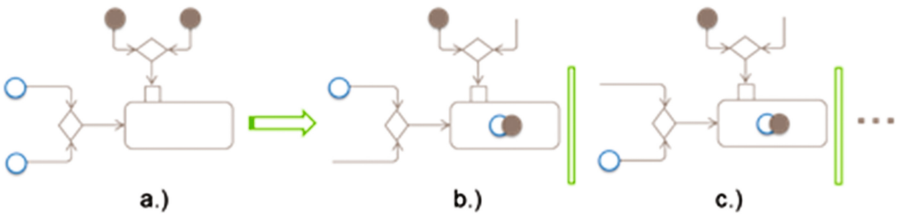


Fig. 3. Possible ways of tokens acceptation.

Metaclasses of the ASmetamodel have operations responsible for creation and expanding a model (a graph) representing semantic of the set of activities – all possible execution traces of the set, Fig. 2. The operations are defined using OCL [7].

For the sake of convenience, we defined operations named *new* supporting instances creations of all ASmetaclasses. In the Snapshot ASmetaclass we defined operation *init*, which sets up tokens initial placement (at the very beginning of an Activity invocation) for a given Snapshot.

We define operations within the UML metaclasses, mainly in ActivityNode. These operations are used to define transitions rules of the Labelled Transition System, introduced in the Subsect. 3.2.

Some of the operations (*acceptEnabled*, *offerEnabled*, *accept*, *offer*) are abstract, because there are specific to particular types of ActivityNode. The *acceptEnabled* (*offerEnabled*) determines whether *acceptation* (*offering*) event is enabled in the context of a node and for a given snapshot. The *accept* (*offer*) operation,

in the context of a node and for a given snapshot, results in a set of successor configurations (transits to new configurations), when the node accept (offer) tokens.

Additionally, there are auxiliary operations supporting LTS definition *enabledEvent* and *transition* common for all ActivityNodes. The *enabledEvent* operation (calling *acceptEnabled* and *offerEnabled* operations) provides available event type for a node within a given snapshot if any. The operation can be seen as a premise for running a transition on the node within the snapshot. The *transition* operation (calling *accept* and *offer*) does enabled transition for a node due to a given event type and snapshot, which results in a set of succeeding configurations.

Because of the space limitation, we put definitions of these operations in the appendix [17].

For the sake of simplicity we write thereafter $c \in Configuration$ meaning that c is an instance of the metaclass Configuration. Similarly, we write thereafter $C \subseteq Configuration$ meaning that C is a set of instances of the metaclass Configuration.

3.1 The Labelled Transition System

The semantics of an activity act_0 is defined in the context of a set of activities (including act_0) $\mathcal{A}$ being used by act_0. We introduce the semantics by means of the Labelled Transition System [6]:

$$LTS(\langle act_0, \mathcal{A} \rangle) = \langle \mathcal{C}, \rightarrow, \mathcal{E} \rangle \tag{1}$$

where:

$\mathcal{C} \subseteq Configuration$ is the set of possible configurations of the $\mathcal{A}$ activities,
$\mathcal{E} \subseteq ActivityEvent$ is the set of the $\mathcal{A}$ activities events,
$\rightarrow \subseteq \mathcal{C} \times \mathcal{E} \times \mathcal{C}$ is the set of allowed transitions between configurations.

We describe transitions $\rightarrow$ as a set of triples $\langle c, e, c' \rangle$, where a triple (a transition) define the very next configuration c', after a configuration c (i.e., to which c is transited), when an event e occurs. The result of applying $\rightarrow$ on c' is a set of new configurations $C' \subseteq \mathcal{C}$. The semantics imposes no order on the particular event to be processed and is nondeterministic.

A single transition: $\langle c, e, c' \in \rightarrow \rangle$ we will note thereafter in following way: $c \xrightarrow{e} c'$.

We distinguish two types of transitions (for each event type: $\rightarrow = \rightarrow_{accept} \cup \rightarrow_{offer}$).

- Transitions when the acceptation type events occur – if an acceptation event is enabled for an activity node within given snapshot (operation *node.acceptEnabled* returns true), then the activity node accepts tokens offered to it from some or all of its incoming edges and starts processing, i.e. for all possible cases of acceptation, new configurations are created (a result of *node.accept* call in stored in C').

$$\rightarrow_{accept} = \left\{ c \xrightarrow{e} c' \middle| c \in \mathcal{C} \wedge e \in \mathcal{E} \wedge e.eventType = acceptation \right.$$
$$\left. \wedge \exists_{s \in c.snapshots} \cdot \text{e}.node.acceptEnabled(s) \wedge c' \in e.node.accept(s) \right\} \quad (2)$$

– Rules for the offering event type – if an offer event is enabled for an activity node within given snapshot (operation *node.offerEnabled* returns true), then the activity node completes execution and token is removed from the node and tokens are offered to some or all of its outgoing edges, i.e. for all possible cases new configurations are created (a result of *node.offer* call in stored in C').

$$\rightarrow_{offer} = \left\{ c \xrightarrow{e} c' \middle| c \in \mathcal{C} \wedge e \in \mathcal{E} \wedge e.eventType = offering \right.$$
$$\left. \wedge \exists_{s \in c.snapshots} \cdot \text{e}.node.offerEnabled(s) \wedge c' \in e.node.offer(s) \right\} \quad (3)$$

We specify transitions for a set of selected kinds of activity nodes. For a given kind of node, a transition is specified for an event if it stems from UML specification. For example the acceptation event is applicable for the FinalNodes, and the ExecutableNodes, however the offering only for the ExecutableNodes [8]. We concretize the *acceptEnabled* and the *accept* operations, if the acceptation event is applicable, and accordingly the *offerEnabled* and the *offer* operations for the offering. Below we discuss transitions for the OpaqueAction, the FlowFinalNode, and the ActivityFinalNode, however for the sake of brevity we present detailed operations only for OpaqueAction and acceptation event.

OpaqueAction – accept

Within a given snapshot, an opaque action can accept tokens, if they are offered to all incoming control flows, and all incoming object flows of the action's mandatory input pins [13, p. 401]. It is defined in the *acceptEnabled* as follows:

$$context\ \mathbf{OpaqueAction} :: \mathbf{acceptEnabled}(snap: Snapshot): Boolean$$
$$body: controlLocationsCases(snap) \rightarrow notEmpty()\ and$$
$$mandatoryInputPins \rightarrow notEmpty()\ implies$$
$$objectLocationsCases(mandatoryInputPins, snap) \rightarrow notEmpty()$$

Tokens are offered to an edge by the source node of the edge. Offers propagate through edges and control nodes until they reach an action [13, p. 374]. As it is discussed in Subsect. 3.1, there are many ways in which tokens can be offered to the action.

We consider variations without repetition of sources of tokens. The *controlLocationCases* operation (see the appendix [17]) returns all mutually exclusive sets of control token locations from which tokens can be accepted by all control incoming edges of the opaque action. Similarly, we do for object token locations by invoking *objectLocationsCases* operation [17], however, ensuring that only mandatory input pins are taken into account. Tokens can be accepted if the control token location set is not empty and the object token location set is not empty, providing that there are any mandatory pins.

Within the given snapshot, when the opaque action begins execution, tokens are accepted from some or all of its incoming edges and a token is placed on the node [13, p. 373]. If there are many ways of tokens acceptation, a way of the acceptance is chosen randomly. Thereby, a new configuration is generated for each possible way, in the following manner:

$$
\begin{aligned}
&context\ \textbf{\textit{OpaqueAction}} :: \textbf{\textit{accept}}(snap: Snapshot): Set(Configuration)\\
&post:\\
&let\ loc: Location = result.snapshots.locations \rightarrow flatten()\\
&\qquad \rightarrow any(l | l.oclIsNew()\ and\ l.node\\
&\qquad = self\ and\ l.token.oclIsNew()\ and\ l.token.oclIsTypeOf(ControlToken))\ in\\
&let\ allInputPinsObjectLocationsCases: Set(Set(Location)) = inputPinCases\\
&\qquad \rightarrow collect(pinCase: Set(InputPin) | objectLocationsCases(pinCase, snap))\\
&\qquad \rightarrow collect\ (case: Set(Location) | case\\
&\qquad \rightarrow reject(l: Location |\ l.node <> null\ and\ l.oclIsType(DataStoreNode)))\\
&\qquad \rightarrow asSet()\ in\\
&mutualExclusiveCases(Set\{controlLocationsCases(snap), allInputPinsObjectLocationsCases\})\\
&\rightarrow forAll(case: Set(Location) | result \rightarrow one(c: Configuration | c.oclIsNew()\ and\ c.snapshots\\
&\qquad \rightarrow excluds(snap)\ and\ c.snapshots\\
&\qquad \rightarrow one(s: Snapshot | s.oclIsNew()\ and\ s.location = snap.locations - case\\
&\qquad \rightarrow including(loc)))\ )
\end{aligned}
$$

A new location *loc* with a new control token is created for the action.

A set of alternative locations for all subsets of input pins, for which tokens can be accepted, is computed. The set is obtained using the *inputPinCases* operation [17] which returns a power set of the input pins. Next, the *objectLocationsCases* operation [17] is applied to each element (a set of input pins) of the power set, and results are collected.

Last but not least, because of the semantics of the DataStoreNode, an object token taken from the data store has to be copied and restored [13, p. 397]. It is imitated by removing the object token from the set of alternative locations.

The operation *mutualExclusiveCases* [17] is invoked to calculate all mutually exclusive locations, i.e. combination of control token locations (result of the *controlLocationsCases* operation call [17]) and object token locations (the *allInputPinsObjectLocationsCases* value).

New snapshots for the alternative locations are calculated and in the result, a new configuration is created for each new snapshot. Finally, the configurations are returned by the *accept* operation.

OpaqueAction – offer

When the opaque action completes an execution a token is removed from the action and tokens are offered to some or all of its outgoing edges, [13, p. 373]. Within a given snapshot, the opaque action completes execution if there is a control token location related to the action. It is defined in the *OpaqueAction::offerEnabled* in the appendix [17].

Control tokens are offered on all outgoing control flows of the opaque action. Object tokens are placed in all mandatory output pins in order to be offered to outgoing object flows of the pins. Depending on a result of the opaque action execution, object tokens may be placed within a subset of optional output pins. Moreover, if object tokens are effectively offered to the ActivityParamterNode or the DataStoreNode, they are immediately propagated to these nodes [13, pp. 396–397]. If there are many ways of tokens offering, a way of the offering is chosen randomly. In such situation, the offer operation generates a set of configurations for the all possible ways. It is defined in the *OpaqueAction::offer* in the appendix [17].

ActivityFinalNode – accept
Within a given snapshot, an activity final node can accept tokens, if they are offered to any incoming control flow. When the activity final node accepts tokens, an activity instance (an execution) referred by the snapshot is finished. It is defined in the *ActivityFinalNode::acceptEnabled* and *ActivityFinalNode::accept* in the appendix [17].

FlowFinalNode – accept
Within a given snapshot, a flow final node can accept tokens, if they are offered to any incoming control flow. When the flow final node accepts tokens, the tokens are removed from an activity instance (an *execution*) referred by the snapshot. If there are no enabled events for any node within the activity instance, the snapshot is removed and the activity instance is destroyed. It is defined in the *FlowFinalNode::acceptEnabled* and *FlowFinalNode::accept* in the appendix [17].

3.2 Reachability Graph

A graph of possible execution traces of an activity act_0 in the context of a set of activities $\mathcal{A}$ is defined as:

$$G(\langle act_0, \mathcal{A} \rangle) = \langle V, A \rangle \tag{4}$$

where:

V – is a set of graph vertices; each vertex is labeled by a configuration which is reachable from the initial configuration c_0 of the activity act_0 (c_0 is an initial configuration when a snapshot of the configuration represents the act_0 instance initialized, i.e. locations are distributed within nodes and edges with respect to rules defined in [13, p. 376]. It is reflected in the *Snapshot::init* [17]).
A – is a set of graph arcs; each arc is labeled by a transition; and has form $\langle u, t, v \rangle \in A$, where $u, v \in V$ are vertices labeled by c_u, c_v : *Configuration*, and t : *Transition*.

The graph is directed. The root of the graph represents an initial configuration c_0, while leafs correspond to final configurations. A sequence of transitions starting from the c_0 results in one of the possible execution traces of the set of activities. The set of all possible execution traces corresponds to semantics of the activities.

We will use the function $label : V \rightarrow Configuration$, which for a given vertex of the graph of possible execution traces returns a configuration labeling the vertex. The graph are constructed iteratively starting from: $V = \emptyset$, $A = \emptyset$.

1. The initial vertex is set up:
   a. The initial configuration c_0 for the activity act_0 is created:
      $c_0 = Configuration :: new(Set\{Snapshot :: init(act_0, Sequence\{\})\})$
   b. Let v_0 be a vertex labeled by the configuration c_0: $V \leftarrow V \cup \{v_0\}$
2. The set of leaf-vertices is defined:
   $V_{leaves} = \{v \in V | label(v) \in c_0 \rightarrow closure(succesors) \rightarrow select(succesors \rightarrow isEmpty())\}$
3. For each leaf-vertex $v \in V_{leaves}$:
   a. For each snapshot of leaf-vertex $snap \in label(v).snapshots$:
      (1) The set of events which enable transitions for snapshot *snap* is defined:
      $EnabledEvents = snap.execution.type.node \rightarrow collect(n|n.enabledEvent(snap))$
      (2) For each enabled event $e \in EnabledEvents$:
         i. The set of new configurations C' are created: $C' = e.node.transition(e)$
         ii. For each new configuration $c' \in C'$:
            (a) The transition t is created: $t = Transition :: new(e.node, e, c, c')$
            (b) Let v' be a vertex labeled by configuration c': $V \leftarrow V \cup \{v'\}$
            (c) Let a be an arc of the form $\langle v, t, v' \rangle$: $A \leftarrow A \cup \{a\}$
   b. Check if any new vertex has been created: $\bigvee_{v \in V_{leaves}} \bigvee_{v' \in V} \bigvee_{t \in Transition} \langle v, t, v' \rangle \in A$, if yes go to step 2.

4 Conclusions

We gave a precise interpretation of a set of Activities in form of set of execution traces. We developed the transformation of a set of activities into the formally precise abstract behavior model, i.e. the graph of possible execution traces. Nodes of the graph correspond to a configuration, arcs correspond to transitions between two configurations, triggered when an event occurs. The provided semantics cover flows of control and object tokens taking into consideration variety caused by control nodes.

The transformation is formulated as the set of OCL rules, which makes it easily portable to transformation frameworks and model-based frameworks. It is the starting point for model analysis, execution (model debugging) tools, or support in checking correctness of model transformations.

In the future works, we will extend the semantics to other parts of the UML Activity notation. We will add support for new type of actions, e.g., call behavior action, send signal action, accept event action, and so far. We will support values transportation in object tokens. Moreover, we want to define semantics for parallel processing.

We want to develop the tool for generating a reachability graph of a set of activities. Next, our goal is to prepare more advanced tool supporting Use-Case specification process, model transformations, etc.

References

1. Daw, Z., Cleaveland, R., Vetter, M.: Formal verification of software-based medical devices considering medical guidelines. Int. J. Comput. Assist. Radiol. Surg. **9**(1), 145–153 (2014)
2. Daw, Z., Cleaveland, R.: An extensible operational semantics for UML activity diagrams. In: Calinescu, R., Rumpe, B. (eds.) SEFM 2015. LNCS, vol. 9276, pp. 360–380. Springer, Cham (2015)
3. Daw, Z., Cleaveland, R.: Comparing model checkers for timed UML activity diagrams. In: Science of Computer Programming, pp. 277–299 (2015)
4. Eshuis, R., Wieringa, R.: Tool support for verifying UML activity diagrams. IEEE Trans. Softw. Eng. **30**, 437–447 (2004)
5. Grönniger, H., Reiß, D., Rumpe, B.: Towards a semantics of activity diagrams with semantic variation points. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part I. LNCS, vol. 6394, pp. 331–345. Springer, Heidelberg (2010)
6. Keller, R.M.: Formal verification of parallel programs. Commun. ACM **19**(7), 371–384 (1976)
7. OMG Object Constraint Language 2.4. http://www.omg.org/spec/OCL/2.4/. Accessed 03 Feb 2014
8. OMG Unified Modeling Language 2.5. http://www.omg.org/spec/UML/2.5/. Accessed 1 Mar 2015
9. Reggio, G., Leotta, M., Ricca, F.: Who knows/uses what of the UML: a personal opinion survey, model-driven engineering languages and systems. In: 17th International Conference, MODELS 2014, Valencia, Spain, pp. 149–165. Springer (2014)
10. Reggio, G., Leotta, M., Ricca, F., Clerissi D.: What are the used activity diagram constructs? – A survey. In: 2014 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), IEEE (2014)
11. Roubtsova, E.: Advances in behavior modeling. In: Advances in Computers, pp. 49–109. Academic Press, Orlando (2015)
12. Störrle, H.: Semantics of control-flow in UML 2.0 activities. In: Bottoni, P., Hundhausen, C., Levialdi, S., Tortora, G. (eds.) Proceedings of the IEEE Symposium on Visual Languages and Human- Centric Computing (VL/HCC), pp. 235–242 (2004)
13. Störrle, H.: Semantics of UML 2.0 acitivities. In: International Symposium on Visual Languages/Human Computer Centered Systems, pp. 235–242 (2004)
14. Störrle, H.: Towards a petri-net semantics of data flow in UML 2.0 activities. Technical report TR 0504, University of Munich (2004)
15. Störrle, H.: Semantics and verification of data flow in UML 2.0 activities. Electron. Notes Theor. Comput. Sci. **127**, 35–52 (2005)
16. Störrle, H., Hausmann, J.H.: Towards a formal semantics of UML 2.0 activities. In: Liggesmeyer, P., Pohl, K., Goedicke, M. (eds.) Software Engineering, Fachtagungdes GI-Fachbereichs Softwaretechnik. LNI, vol. 64, pp. 117–128. GI (2005)
17. Walkowiak, A., Gall, D.: An approach to semantics for UML activities – appendix. https://www.dropbox.com/s/11jrz5zo6lk5vxr/OCL_Appendix.pdf?dl=0. Accessed 31 May 2017

CDMM-F – Domain Languages Framework

Piotr Zabawa[1(✉)] and Bogumiła Hnatkowska[2]

[1] Cracow University of Technology, ul. Warszawska 24,
31-155 Kraków, Poland
Piotr.Zabawa@pk.edu.pl
[2] Wroclaw University of Science and Technology,
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
Bogumila.Hnatkowska@pwr.edu.pl

Abstract. Domain Specific Languages (DSLs) are mini-programming languages which enable their users to abstract from technical details and focus on business domain. DSLs can be used within a framework, i.e. platform for developing software applications. The paper presents such a framework called CDMM-F for building Java applications. The additional tools, prepared by the authors, support DSL definition. The constraints a DSL should fulfill to be CDMM-F compliant are thoroughly described in the paper, expressed in the form of the CDMM meta-meta-model and demonstrated in a case-study. The main advantage of proposed solution is meta-meta-model simplicity and high reusability of DSL elements. Once defined they can be connected in different configurations (contexts) according to the actual needs. The framework architecture that enables this feature is also presented.

Keywords: Domain Specific Language · DSL · Meta-meta-model · Meta-model · Model · Modeling language · Framework

1 Introduction

Domain Specific Languages (DSLs) are widely used in software engineering. They are mini-programming languages which enable their users to focus on a given application or business domain without necessity to address technical problems. The problems are solved "behind the scene", by the DSL designer. In that way DSL languages "offer significant advantages over General Purpose Languages (GPLs)" [1].

Technical problems can be solved also as a part of a software framework, i.e. platform for developing software applications [2]. "A framework may include predefined classes and functions", it "serves as a foundation for programming" [2].

The aim of this paper is to present a partially published framework called Context-Driven Meta-Modeling Framework (CDMM-F) [3] from the perspective of its application to DSLs definition as well as their processing. The tools being a part of the framework support definition of DSL languages with highly reusable parts. The element once defined can be used later in many different applications. What is more, it can be easily configured to work with other elements offering desired functionalities.

The main contribution of the paper is the formal definition of the meta-meta-model (see Sect. 3.1) which all DSLs supported by CDMM-F must be compliant with. Up to

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_24

now [4, 5], the meta-meta-model was described in informal way only, e.g. by presenting its possible instances. The formalization serves two important purposes: (a) eliminates possible misunderstandings of informal specification, (b) is the basis for internal representation of meta-meta-model instances, and explains existing constraints imposed on them. In the context of Model Driven Engineering [6], the formally expressed meta-meta-model is a necessary artifact for definition of meta-models' transformations, both horizontal – at the same level of abstraction, e.g. refactorings or translation to another (modeling) language – and vertical, at different level of abstraction, e.g. model to database schema transformation [7].

The secondary contribution is the new perspective from which CDMM-F framework is presented – as a tool for DSL definition and its further processing.

The literature overview is described in Sect. 2. The CDMM-F framework is presented in Sect. 3. A simple case study of a DSL definition and its further usage within the CDMM-F is given in Sect. 4. Section 5 concludes the paper.

2 Literature Overview

Domain Specific Languages (DSLs) are programming languages developed to facilitate solving problems in a given domain by making programs "easier to understand and therefore quicker to write, quicker to modify, and less likely to breed bugs" [8].

DSLs are classified according to representation they offer to their users into graphical and textual ones. Textual DSLs are further divided into external and internal. An external DSL is represented in another language than the general programming language (GPL, host language) it is working with. An internal DSL is represented within the syntax of general-purpose language, e.g. Java [8].

GPL or modern frameworks like Spring, can be complicated to learn. DSLs can hide this complexity providing more user friendly API, as in the case of CDMM-F.

"A language workbench is an environment designed to help people create new DSLs, together with high-quality tooling required to use those DSLs effectively" [8]. There exist some examples of language workbenches freely available. The popular are: Modeling SDK for Visual Studio (MSDK) [9], or Xtext [10]. Both support definition of external DSL languages for which a transformation to a GPL language is necessary. The former enables definition of graphical DSLs while the latter – definition of textual ones. In both a developer has first to design a meta-model along with abstract, and concrete syntax. In Xtext the abstract syntax "can be described in XMI, XML Schema, Rational Rose, or annotated Java" [10], and the concrete syntax is expressed with the use of BNF notation. In MSDK the abstract syntax takes a form of a graph, but a DSL designer can propose graphical symbols for graph elements (concrete syntax) [9]. The workbenches generate appropriate model editors, and support transformation of model instances to a host programming language on the basis of transformation templates (definition of language semantics). Comparison of the selected tools for DSL definition was given in [11] while an example of their application is presented in [12].

The CDMM-F can be perceived as a partial implementation of a DSL workbench for DSLs being an extension to Java. On the other side, the CDMM-F is also a framework which is able to process specific DSLs.

Typically, a DSL workbench enables definition of tree basic elements: abstract and concrete syntax of a DSL language, and semantics for language elements. In the proposed implementation, the abstract syntax can be defined graphically (this step is optional) in the form of a CDMM specific class diagram with the help of CDMM-Meta-Modeler modeling tool [4]. Concrete syntax takes a form of Spring xml configuration file, in which DSL parts are bound together. Semantics of DSL elements is expressed in Java (implementation of classes and interfaces).

Within CDMM-F framework, definition of the DSL language is considered as a meta-modeling activity while its usage (create instances of classes) as modeling.

3 CDMM-F – Framework Description

The CDMM-F framework is described in the subsections below from the perspective of different meta-levels. Then the characteristics of technologies used for the design and implementation of CDMM-F is presented. The motivation and the role of particular technologies applied in CDMM-F are also explained.

3.1 Meta-Meta Model – Conceptual Level

By analogy to the UML and MOF standards of OMG, the meta-meta model layer is introduced in this section to help determine whether a modeling language is compliant with CDMM. Such a meta-model is named CDMM compliant in this case.

The meta-meta model for CDMM is presented in Fig. 1.

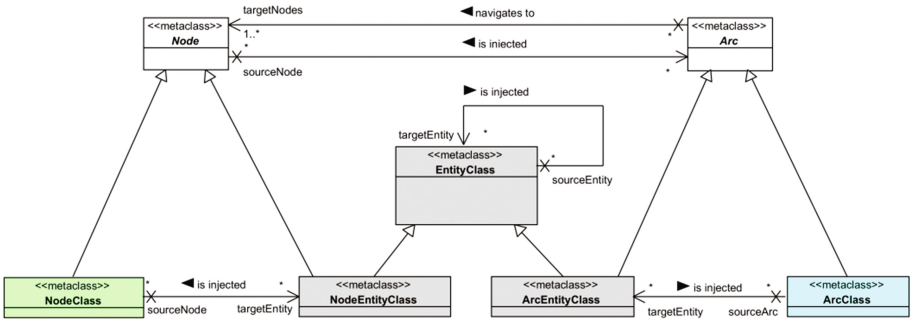


Fig. 1. Meta-meta model – basic elements.

The CDMM compliant modeling language forms a graph constructed from *Node* and *Arc* meta-classes. *Node* meta-class may be empty (*NodeClass*) if the role of it is just to be placed in the meta-model graph node. *Arc* meta-class is responsible of representing and giving access to its' ends (*targetNodes*) are intended to be placed in meta-model graph arcs. Thus, instances of both *Node* and *Arc* meta-classes constitute a meta-model graph. The graph is built at run-time through injections of *Arc* classes to *Node* classes represented by "is injected" association.

EntityClass is used to model entities in the modeled domain, e.g. *Name* in the example presented in Sect. 4. In general, *EntityClass* serves as representation of domain knowledge in terms of value objects. Entities may constitute a hierarchy modeled by self-association introduced to *EntityClass*. The hierarchy may be created at run-time by injecting *Entity* classes to *Entity* classes.

Node and *Arc* meta-classes may have their specializations of *EntityClass*, which are represented in the meta-meta-model by *NodeEntityClass* and *ArcEntityClass*. This way *Node* and *Arc* classes may be enriched by *EntityClass* properties at run-time via injecting *EntityClasse(s)*.

To make understanding of presented concepts easier, a simple example was prepared. Figure 2 presents a CDMM-compliant UML object diagram for genealogy tree meta-model from Fig. 4, explained in the case-study from Sect. 4.

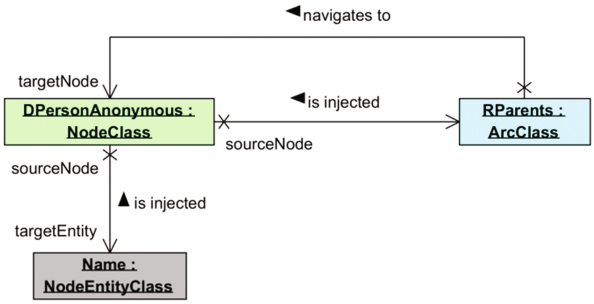


Fig. 2. CDMM-compliant representation of UML class diagram for genealogy tree.

The diagram presented in Fig. 2 illustrates the mapping between meta-model elements and meta-meta-model elements given in Fig. 1. The meta-model is created by injecting *RParents* class and *Name* class to *DPersonAnonymous* class. The *RParents* class plays the role of an *Arc* in the meta-model graph and it points to *DPersonAnonymous* class as *targetNode*. The attributes contained in the *Name* class are associated to the *DPersonAnonymous* class by injecting the *Name* class to the *DPersonAnonymous* class.

When the meta-model instance (the model) is created one could traverse the graph in the direction of the navigable associations. As long as the traversal starts in *DPersonAnonymous* class all elements of the model may be visited.

The meta-model is created at run-time from the meta-model definition (see Sect. 3.2 for details). The meta-model instance (model) is created by a programmer through the appropriate CDMM-F framework API as it is shown in the Sect. 3.3.

3.2 Meta Model

In order to define a CDMM compliant meta-model some elements should be defined by the modeling language designer. The most important are classes as they must be implemented in an object-oriented programming language.

Arc classes must contain the appropriate references to their ends (*targetNodes*), which will be initialized later, on the stage of the meta-model construction (for meta-model classes) and model construction (for meta-model objects).

Node class can be empty. In this case it is the place for entity class injections and takes the form of *NodeClass*. This is also the place for *Arc* class injections.

The meta-model designer may also introduce entity or value object classes to the meta-model *Node* and/or *Arc* classes. In this case the classes must be also implemented and the designer decides were these classes will be injected into the meta-model graph classes. These entity classes may be organized in hierarchies as entity class may be injected into the other entity classes.

All these classes, that is *Node*, *Arc* and entity classes, are subject of reuse. Their reusability has dual nature: they can be reused between different meta-models and also between meta-models and end-user applications.

The meta-model being the graph consisting of interrelated classes is created by the CDMM-F while loading the meta-model definition file (application context). This way all classes are interrelated at run-time and the CDMM-F compliant meta-model is created.

The CDMM-F user may create instances of meta-model classes and traverse the links between them in the model. The characteristics of the model is presented in Sect. 3.3.

3.3 Model

Models are defined by users through the CDMM-F API defined by the meta-model designer. The main characteristics of this API is that it requires casting mechanisms to get access to model elements. The reason of that is the fact that the meta-model is created at run-time and the meta-model graph does not contain compile-time relations. The user can query the meta-model for its structure and use it further to create instances of meta-model elements as well as to traverse the graph by its own.

3.4 Framework Architecture

The CDMM-F framework was implemented according to the list of key technical assumptions:

- object-oriented (o-o) paradigm must be used to express meta-models and models
- meta-model must be created at run-time from existing classes which are completely unrelated at compile-time
- the meta-model graph traversal is handled by casting in place of calling built-in traversing methods.

Lack of compile-time relationships significantly increases the reusability of all meta-model elements, and the run-time creation of a meta-model makes change introduction much easier than the one possible at compile-time.

The important problem handled by the CDMM-F is to provide injection mechanism. Without it the meta-model could not be created at run-time. The injection mechanism is supported by dependency injection (DI) based on the inversion of control concept. Dependency inversion is needed to provide navigability in the opposite direction than the injection – see Fig. 1.

Another problem is how to eliminate compile-time dependencies from the meta-model class hierarchy. The AOP paradigm can be applied to move these dependencies, more specifically the navigable association between *Node* and *Arc* classes from class-object oriented paradigm layer to the layer of aspects.

In the case of CDMM-F both DI and AOP are used together with Java EE beans concept. The combination of such the architectural elements makes it possible even to define injections at run-time through the Java EE framework configuration XML files, namely through the Java EE application context files. Both DI and AOP can be applied to beans and all user-defined classes (*Node*, *Arc* and entity) are represented by Java EE beans.

4 Case Study

A simple and frequently met in tutorials [13] case-study of genealogy tree is presented in this section in order to illustrate how to define a CDMM-F compliant meta-model consistent with the CDMM compliant meta-meta model and how the CDMM-F API can be used.

The case-study is limited to the two-level genealogy tree presented in Fig. 3.

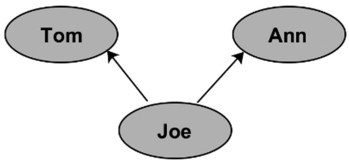


Fig. 3. Sample genealogy tree.

There are two parents: father Tom and mother Ann that have one child, the boy called Joe.

4.1 Meta-Model Definition

The possible CDMM-F compliant meta-model for genealogy tree is presented in Fig. 4. The main element on it is *DPersonAnonymous* class which models a person node (parent or child). We want each person to have a name (represented by the *Name* entity class). Peoples are connected with *RParents* arc class.

The coloring convention known from papers [5] is reused in Fig. 4. According to it, red types represent elements defined inside the CDMM-F, green – meta-model graph nodes, blue – meta-model graph arcs and grey – structural responsibilities (entities).

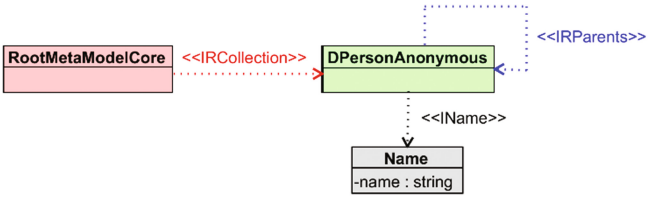


Fig. 4. CDMM-F compliant genealogy sample meta-model.

The *RootMetaModelCore* class plays the role of the meta-model graph root [14]. The meta-model root is the starting point of the meta-model traversal process, like in the case presented on the Listing 2.

It is required for both node and arc classes to implement appropriate interfaces – see Fig. 5. The used naming convention assumes that the name of interface is preceded with "I" letter while "D" prefix is used for node classes and "R" prefix is used for arc classes.

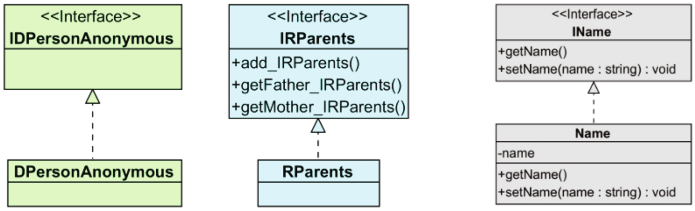


Fig. 5. CDMM-F compliant genealogy sample meta-model – definition of interfaces.

The operations an interface of arc classes declares fully depend on the invention of the designer of the meta-model.

It is clear while looking at Fig. 5 that the node class *DPersonAnonymous* does not contain attributes and both node class and its interface do not contain operations. The node class constitutes the place where the interface *IName* of an entity class may be injected together with its implementation. Both *DPersonAnonymous* and *RParents* classes including the methods shown in the *IRParents* interface must be implemented by the person who defines the meta-model.

As it was mentioned in Sect. 3.4, the meta-model concrete syntax takes a form of XML file being the definition of Spring+AspectJ application context. Thanks to that additional structural responsibilities (like *Name*) may be injected to the meta-model graph elements in the aspect-oriented way. An example for the meta-model from Fig. 4 is shown on the Listing 1.

```
<import resource="../metamodel-core-context.xml"/>
<import resource="../appContextNamedElement.xml"/>

<!-- Root -->
<bean
class="com.componentcreator.metamodel.coremetamodel.root.RootMet
amodelCore"
  id="root" scope="singleton"></bean>

<!-- Meta-model node classes -->
<bean
class="com.componentcreator.metamodel.coremetamodel.gen.domainsi
mpl.DPersonAnonymous"
  id="class" scope="prototype"></bean>

<!-- Meta-model arc classes -->
<bean
<!-- ... -->
>
</bean>

<!-- Responsibility injections -->
<aop:config>
  <aop:aspect id="holderA" ref="holderAAspect">
    <aop:declare-parents
<!-- ... -->
    />
  </aop:aspect>
</aop:config>
```

Listing 1. CDMM-F meta-model definition for sample genealogy meta-model.

The meta-model must be associated to the CDMM-F's *RootMetamodelCore* class. The relationship is responsible for getting access to each meta-model element while traversing the meta-model from the root.

It is worth noticing that meta-models are reusable to a large extent. The types *IName* and *Name* may be taken from other projects (of modeling languages or even from end-user applications in Java) and may be injected into the newly created meta-model without any compile-time change. The same can be done with hierarchies.

The Spring+AspectJ application context file form the Listing 1 is supported by CDMM-Meta-Modeler Eclipse PlugIn [4, 5] from the graphical definition (CDMM compliant diagram) of the meta-model for convenience.

4.2 Model Definition

The source code snippet for the CDMM-F API client implemented with the help of JUnit4 (similar to a unit test) and illustrating how the model (meta-model instance) can be created, and how its nodes are visited (traversed) is shown in the Listing 2.

```
public class TestGenealogyPerson {
  private final String APP_CFG_FNAME = "aGenealogy.xml";

@Test
public void test() throws Throwable {
  // getting access to the interface
  ICoreMetamodelAPIdynamic ifc =
Configuration.getCoreMetamodelAPIdynamic(APP_CFG_FNAME);

  // getting access to the meta-model root
  RootMetamodelCore root = ifc.getRoot();

  // creating meta-model node class instances
  DPersonAnonymous tom =(DPersonAnonymous) ifc.getDomain(
       DPersonAnonymous.class, "tom");
  ((Name)tom).setName("Tom");
  // ...

  // adding the child by the proxy to the meta-model root
  BaseMetamodelCoreProxy rootAccessor = ifc.getAccessor(
       RootMetamodelCore.class, IRCollection.class);
  rootAccessor.add(root, joe);

  // getting access to the meta-model
  // DPersonAnonymous class
  BaseMetamodelCoreProxy pAccessor = ifc.getAccessor(
       DPersonAnonymous.class, IRAssociationPair.class);

  // adding Joe's parents
  pAccessor.add(joe, tom, ann);

  System.out.println("Joe's father: " +
  ((Name)pAccessor.getFather(child)).getName());
  System.out.println("Joe's mother: " +
  ((Name)pAccessor.getMother(child)).getName());

  // closing allocated meta-model resources
  ifc.close();
}
```

Listing 2. The CDMM-F unit test for the sample meta-model.

The objects *rootAccessor* and *personAccessor* above are proxy objects belonging to the CDMM-F query language and were added to the framework to simplify the API. The result of this test execution is presented on the Listing 3.

```
Joe's father: Tom
Joe's mother: Ann
```

Listing 3. Test execution result.

The result of the test confirms the CDMM concept feasibility for the case-study in the form the CDMM-F requires.

5 Summary

Software reusability is a very desired software feature which can be achieved by application of selected design principles, e.g. loose-coupling, high-cohesion [15]. These principles can be implemented using different means, e.g. interfaces and dependency injections. The design elements that have the higher reusability potential are domain classes.

The paper presents CDMM-F framework that supports development of Java applications operating on domain objects. These objects are instances of DSL languages defined separately. DSL classes can be combined in a desired way in declarative manner via XML configuration file. In result one can change the structure of the graph only by replacing this file. It is possible because the framework bases on interfaces and uses aspects to influence the classes' hierarchy.

The implementation of CDMM-F framework imposes some restrictions on applicable domain languages. These restrictions are expressed by a meta-meta model presented in the paper. Similarly to other DSL definitions [9, 10] it is required that the elements of DSL form a graph consisting of nodes and arcs. What is different, structural features can be assigned to both. It is also possible to define an aggregation hierarchy of classes and – in result – their instances.

The CDMM-F was described in the paper from the DSL perspective. In the consequence, the paper is focused on the representation of the graph structure of the CDMM compliant meta-model, i.e. on the meta-meta-model, and on the way the meta-model instance graph (model) can be created via the CDMM-F API. The important element of DSLs which is DSL application business logic is not described in the paper as it is subject of future implementation.

References

1. Vasudevan, N., Tratt, L.: Comparative study of DSL tools. Electron. Notes Theoret. Comput. Sci. **264**(5), 103–121 (2011)
2. The Tech Terms Computer Dictionary. https://techterms.com/definition/framework. Accessed 19 July 2017
3. Zabawa, P.: Context-driven meta-modeling framework (CDMM-F) – context role. Tech. Trans. **1-NP**, 105–114 (2015)

4. Zabawa, P., Fitrzyk, G.: Eclipse modeling plugin for context-driven meta-modeling (CDMM)-meta-modeler. Tech. Trans. **1-NP**, 115–125 (2015)
5. Zabawa, P., Fitrzyk, G., Nowak, K.: Context-driven meta-modeler (CDMM)-meta-modeler application case-study. Inf. Syst. Manag. **5**, 144–158 (2016)
6. Rodrigues da Silva, A.: Model-driven engineering: a survey supported by the unified conceptual model. Comput. Lang. Syst. Struct. **43**, 139–155 (2015)
7. Mens, T., Van Gorp, P.: A taxonomy of model transformation. Electron. Notes Theoret. Comput. Sci. **152**, 125–142 (2006)
8. Fowler, M., Parsons, R.: Domain Specific Languages. Addison Wesley, Upper Saddle River (2011)
9. Modeling SDK for Visual Studio – Domain-Specific Languages. https://msdn.microsoft.com/en-us/library/bb126259.aspx. Accessed 19 July 2017
10. Bettini, L.: Implementing Domain-Specific Languages with Xtext and Xtend. Packt Publishing, Birmingham (2013)
11. Hnatkowska, B., Klekotka, A.: Comparison of the most popular tools for DSL definition. In: Borzemski, L., et al. (eds.) Information Systems Architecture and Technology: New Developments in Web-Age Information Systems, pp. 273–283. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2010)
12. Hnatkowska, B., Kasprzyk, K.: Integration of application business logic and business rules with DSL and AOP. e-Informatica Softw. Eng. J. **4**(1), 59–69 (2010)
13. Getting Started with Domain-Specific Languages. https://msdn.microsoft.com/en-us/library/ee943825.aspx. Accessed 19 July 2017
14. Harary, F.: The number of linear, directed, rooted, and connected graphs. Trans. Am. Math. Soc. **78**, 445–463 (1955)
15. Larman, C.: Applying UML and Patterns—An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edn. Prentice Hall, New Jersey (2004)

Complex Process Modeling

An Analytical Modeling Approach to Cyclic Scheduling of Multiproduct Batch Production Flows Subject to Demand and Capacity Constraints

Grzegorz Bocewicz[1(✉)], Peter Nielsen[2], Zbigniew Banaszak[2], and Robert Wójcik[3]

[1] Faculty of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland
bocewicz@ie.tu.koszalin.pl
[2] Department of Materials and Production, Aalborg University, Aalborg, Denmark
z.banaszak@wz.pw.edu.pl
[3] Department of Computer Engineering, Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland
robert.wojcik@pwr.edu.pl

Abstract. This paper addresses the problem of determining the admissible configuration and cyclic operation of batch plants in which a variety of products are simultaneously processed along different product-dedicated production lines. The objective is to determine the required number and size of AGVs as well as the levels of product inventories given the takt times assumed for each kind of product. The major complication of this design problem lies in the large number of trade-offs involved, for instance the merging of input/output buffer capacities versus the frequency of AGV circulation, and the length of takt time versus inventory levels. By using an analytical model encompassing bottleneck-driven representation of cyclic schedules it is shown that this problem can be formulated as a constraint satisfaction problem. Its task-oriented representation, aimed at reducing the size of the search space, is discussed. An illustrative example proving the feasibility of the proposed approach is provided.

Keywords: Cyclic scheduling · Production flow · Takt time · Batch size · Input output buffer · Production flow

1 Introduction

Cyclic scheduling is one of the most effective methods of planning in transport systems and operational planning in manufacturing systems [12]. In transport systems this approach is predominantly used in areas related to the transport of people including: rail transport, urban transport, intercity bus transport, etc. Rhythmic delivery repeated at regular intervals is also a feature of systems of cyclic delivery of goods, such as food products or consumables, to distribution centres.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_25

Some of the most important cyclic scheduling problems are optimization problems such as the Basic Cyclic Scheduling Problem (BCSP) [9, 12]) and its extensions associated with scheduling in production job-shops (so-called Job-shop Problems): the general Cyclic Job-shop Problem (CJP) [2], the Cyclic Flow-shop Problem (CFP) [7, 12]), the Cyclic Open-shop Problem (COP), and the Cyclic PERT-Shop Problem (CPSP) [11]. With the exception of BCSP, all the problems listed above are NP-hard problems which means when encountered in practical problems sizes they can only be solved using AI methods [12, 14].

The growing market demand for multi-product manufacturing gives rise to widespread interest in models of multi-machine cyclic production systems, i.e. models oriented towards cyclic production methods in which a batch (a set) of items is produced at fixed time intervals (cycle periods) [8, 13]. Optimization of a process usually amounts to minimizing the cycle period or ensuring that a certain (usually minimum) part set (MPS) of products is manufactured [1, 18]. Products in an MPS are only released for production after all products from the previous-cycle MPS have been manufactured. In other words, successive MPSs (i.e. products in these MPSs) are produced in a cyclic manner one after another, which means that products from MPSs manufactured in different cycles are never mixed.

One generalization of this model is the concept of *heijunka* (production leveling), understood as a form of cyclic scheduling and defined as "the distribution of production volume and mix evenly over time" [15]. As in the case of cyclic production oriented towards MPS scheduling, *heijunka* requires that a base period, referred to as EPEI (Every Part Every Interval), is determined [10, 15]. The base period is a period during which the whole mix of products has to be produced. At the a tool in the core of the principle of production levelling is the heijunka box. It is a cyclic production schedule divided into a grid of boxes in which the columns represent a specific period of time and the rows represent product types [10].

In the context of the concepts of cyclic production and production levelling, as discussed above, the problem of cyclic scheduling of multi-product batch production flows subject to demand and capacity constraints, which is the focus of the present study, boils down to determining the method of configuring production flows (in particular transport between workstations, the size of production batches and the capacity of input/output buffers) under assumed production takt times and the associated scheduling cycles. In our discussion, it is assumed that the production routes of the different products intersect at shared workstations (used in accordance with the mutual exclusion policy), and that the end of the cycle of each cyclic schedule is a multiple of the production takt time determined by the bottleneck of the production system. This means that the production takt time of a cyclic schedule of a multi-product batch manufacturing system is determined by the bottleneck station, i.e. the most heavily utilized workstation. A scheduling period is then the period during which the whole mix of products has to be produced, with its individual takts determining the times at which the production of the particular batches of products making up the mix has to be completed. The set of admissible schedules identified in this way can be used to determine the cyclic schedules that minimize the volume of work in progress or the total capacity of the input/output buffers. The next Section discusses the assumptions of the adopted modelling framework. Section 3 presents an example illustrating the

relationships that connect the size of production batches and buffer capacities with production takt time and the cycle period of a cyclic schedule. In Sect. 4, the problem of prototyping alternative production flow configurations that meet the given constraints imposed on output rate is formulated. Section 5 shows how a production flow variant that minimizes the total storage capacity of an intermediate storage buffer while satisfying the constraints imposed by the production takt time can be determined. Section 6 contains a discussion of the conclusions and possible directions of future research.

2 Modelling

The cases of multi-product batch production systems discussed below are limited to the class of flow production systems composed of production lines. Different production lines are dedicated to different products. The analysis concerns methods of organizing production governed by cyclic schedules in which a cycle is a multiple of the time of operations performed on the bottleneck of the production system. A bottleneck is understood as the workstation that has to handle the most workload.

2.1 Model Assumptions and Limitations

The following assumptions are considered:

1. A multi-product flow production system encompasses a set of production lines.
2. Each production line produces one type of product.
3. The production route of each line passes through at least one workstation belonging to another route (a graph of production routes is a coherent graph).
4. Each production workstation is equipped with a storage buffer of known capacity.
5. Each job (operation on the production route) can be processed on only one workstation at a time.
6. No workstation can process more than one job at a time.
7. The processing of a job cannot be interrupted.
8. The jobs must be processed in the same technological order.
9. Transport operations between pairs of workstations are carried out cyclically using means of transport of a given capacity (e.g., Automated Guided Vehicles, AGVs).
10. The total time of the operations executed by a means of transport cannot be longer than the time of workstation operations.
11. It is assumed that product loading and unloading times are included in the times of operations executed at the individual workstations.
12. The model is multi-product, where actions and flow of materials take place along different, product-type-oriented production routes.
13. The model is multi-period, where actions take place in multi-periods.
14. Customers' demands specified by production takt times are fixed and known.
15. An integer number of batches are transported.

2.2 Model Formulation

The model involves the following sets, parameters and variables:

Sets

R set of resources (workstations), indexed by k,
W set of products, (multimodal processes) indexed by i,
P set of transportation means e.g. AGVs (local processes), indexed by j,
Q_k set of products using resource k.

Parameters

m number of resources,
n number of products manufactured,
l number of means of transport,
R_k resource k,
W_i product i,
$ts_{i,k}$ operation time for one product item W_i on resource R_k,
Δt travel time between workstations,
TP_i* maximum production takt time for product W_i.

Variables

T period of system,
TP_i production takt time for product W_i,
$b_{i,k}$ batch size of product W_i processed on resource R_k,
$bmax_i$ maximum production batch size for product W_i,
u_k size of input/output buffer on resource R_k,
tz_k occupation time for resource R_k,
$to_{i,k}$ processing time of product W_i on resource R_k,
$v_{i,k}$ frequency (per period T) of delivery of product W_i to resource R_k,
$c_{i,k}$ capacity of the means of transport (AGV) for moving product W_i to resource R_k.

The relationships among the above variables are described by the following constraints:

- Period T of a system is determined by the bottleneck understood, in this case, as resource R_k with the longest occupation time tz_k,

$$T = max\{tz_1, \ldots, tz_k, \ldots, tz_m\}, \tag{1}$$

  where occupation time tz_k of resource R_k is defined as the sum of times $to_{i,k}$ during which the resource is occupied by products from set Q_k (the set of products using resource R_k):

$$tz_k = \sum\nolimits_{i \in Q_k} to_{i,k}. \tag{2}$$

- Processing time $to_{i,k}$ of a batch of product W_i on resource R_k is given by:

$$to_{i,k} = v_{i,k} \cdot ts_{i,k} \cdot c_{i,k} + \min\{v_{i,k}, v_{i,k+1}\} \cdot \Delta t, \tag{3}$$

where:

$$c_{i,k} = min\{b_{i,k}, b_{i,k-1}\}, \tag{4}$$

$$v_{i,k} = max\left\{\frac{bmax_i}{b_{i,k}}, \frac{bmax_i}{b_{i,k-1}}\right\} = \frac{bmax_i}{min\{b_{i,k}, b_{i,k-1}\}}, \tag{5}$$

$$bmax_i = \max\{b_{i,1}, \ldots, b_{i,k}, \ldots, b_{i,n}\}. \tag{6}$$

By substituting expressions (4) and (5) into (3), we obtain:

$$to_{i,k} = ts_{i,k} \cdot bmax_i + \min\{v_{i,k}, v_{i,k+1}\} \cdot \Delta t. \tag{7}$$

- The size of buffer u_k on resource R_k is determined by the maximum size of the batches processed on this resource

$$u_k = \min_{i \in Q_k}\{b_{i,k}\}. \tag{8}$$

- Production takt time for product W_i is defined as the number of product items $bmax_i$ produced per period T:

$$TP_i = \frac{T}{bmax_i}. \tag{9}$$

3 Motivating Example

The elements of the model proposed in this study are illustrated using the example of the system in Fig. 1 (representing a part of real flexible manufacturing system [17]). In this system, given is a set of production tasks (jobs) associated with the rhythmic (flow) production of differently-sized groups of different products W_1 and W_2 and a set of workstations and transportation sectors encompassing resources $R_1 - R_7$ and $R_8 - R_{19}$ respectively. The job execution times on workstations $ts_{i,k}$, the size of batches processed $b_{i,k}$, the delivery rate $v_{i,k}$, and the capacity of the means of transport $c_{i,k}$ (see Fig. 1) are given.

The goal is to find a cyclic schedule which guarantees the production of the products in batches of a given volume, available at production takt times TP_1 and TP_2. In other words, what is sought is the answer to the question: What production takt times are achievable in the given system? Travel time between operations (using transport resources) is expressed in unit values (Δt = 1 unit of time - u.t.).

The relationships (1)–(9) describing the conditions that govern the cyclic production flow were used to determine the values of production takt times achievable in the considered system. The workstation processing times $to_{i,k}$ obtained are summarized in

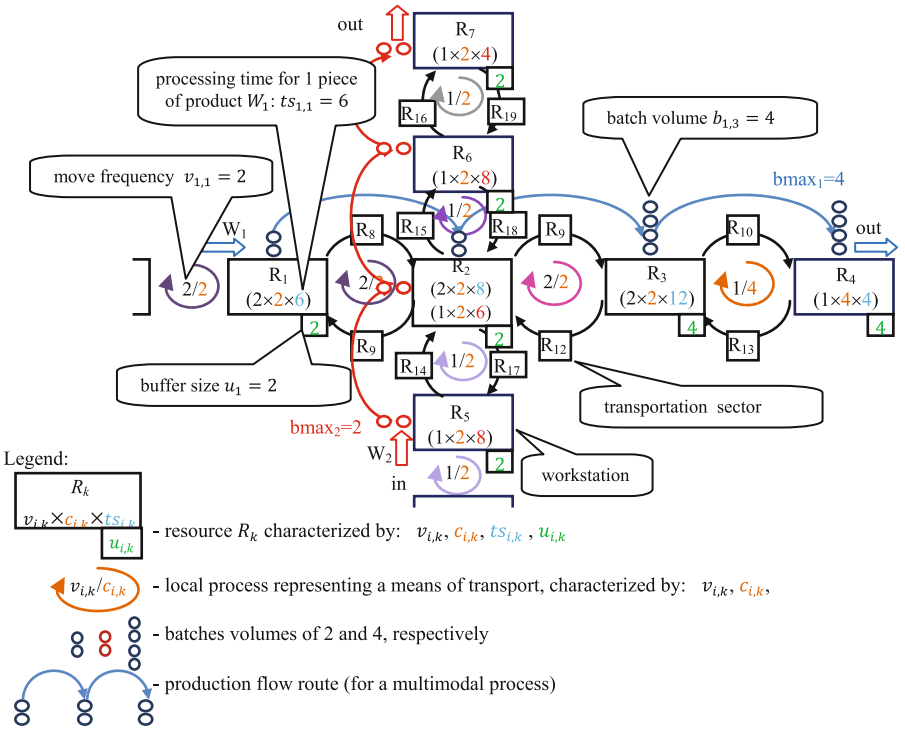


Fig. 1. A schematic diagram of production flow of products W_1 and W_2

Table 1. The resulting period (1) of the cyclic schedule is $T = max\{26, (34+13), 49, 17, 17, 17, 9\} = 49$ u.t. and the respective production takt times (9) for products W_1 and W_2 are $TP_1 = 12.25$ u.t. and $TP_2 = 24.5$ u.t. The bottleneck of the production system is the most heavily loaded workstation R_3, for which the occupation time is $tz_3 = to_{1,3} = 49$ u.t. A cyclic schedule representing the execution of (transport and workstation) operations in time was determined by solving a relevant constraint satisfaction problem *CS* [16] presented in studies [3–5]. The schedule is shown in Fig. 2. The Fig. 2 shows that products W_1 and W_2 flow through the line at production takt times of $TP_1 = 12.25$ u.t. and $TP_2 = 24.5$ u.t. The takt values obtained are determined by workstation (process step) R_3 which constitutes the bottleneck in the production flow.

Table 1. Workstation processing time $to_{i,k}$ for a batch of product W_i on resource R_k

$to_{i,k} = ts_{i,k} \cdot bmax_i + \min\{v_{i,k}, v_{i,k+1}\} \cdot \Delta t$, $\Delta t = 1$ u.t.			
	$bmax_1 = 4$		$bmax_2 = 2$
R1	$to_{1,1} = 6*4+2 = 26$ u.t.	R5	$to_{2,5} = 8*2+1 = 17$ u.t.
R2	$to_{1,2} = 8*4+2 = 34$ u.t.	R2	$to_{2,2} = 6*2+1 = 13$ u.t.
R3	$to_{1,3} = 12*4+1 = 49$ u.t.	R6	$to_{2,6} = 8*2+1 = 17$ u.t.
R4	$to_{1,4} = 4*4+1 = 17$ u.t.	R7	$to_{2,7} = 4*2+1 = 9$ u.t.

The location of bottlenecks in multi-product batch flow production systems depends on the sizes of available input/output buffers u_k, and, consequently, the admissible batch sizes.

Buffers ensure that one cannot introduce more production orders into the system than what the system can handle. A question that suggests itself at this point is whether it is possible to find system parameters (e.g. batch sizes) which would guarantee the required production takt times, with the location of the bottleneck remaining unchanged?

4 Problem Statement

Given is a multi-product batch flow production system which meets the assumptions formulated in Sect. 2. It is assumed that the times of the technological operations $ts_{i,k}$ executed as part of the individual orders are known. The capacity of input/output buffers u_k and the size of production batches $b_{i,k}$ may vary. The goal is to find production flow configurations in which jobs are processed in a fixed sequence corresponding to a cyclic schedule that times production according to production takt time to meet customer expectations regarding batch sizes of the individual product groups. To find solutions to this problem, one needs to answer questions belonging to two groups of problems known in the literature as analysis and synthesis problems [4, 5]. The following question is an example of the former group of problems:

- Is it possible, given the structure of the production system, to obtain fixed cyclic execution of processes that would meet the expectations of customers regarding timely production of batches of orders?

A corresponding synthesis (reverse) problem is given below:

- Are there technological solutions (product batch sizes/workstation buffers) for the processes which enable timely production of the ordered product range that would satisfy the constraints imposed by the structure of the production system?

The analysis problem, considered in Sect. 3, can be formulated in the following way: Given is a system with a known set of resources R, in which products W are produced. Known are times $ts_{i,k}$, travel times between workstations Δt and the sizes of production batches $b_{i,k}$. The goal is to find an answer to the question of what production takt times TP_i are achievable in this system? The solution to this problem, reduced to the form of a constraint satisfaction problem CS [5, 6], is the production flow configuration shown schematically in the Gantt diagram of Fig. 2. Because this type of problem must take into account the decision variables defining operation start times, the size of the search space is described by an exponential function:

$$f(q) \geq d^q, \tag{10}$$

where: d is the size of the smallest domain of variables defining operation start times (see Fig. 1) and q is the number of operations executed in the system (for the system considered here, $q = 32$, $d = 50$).

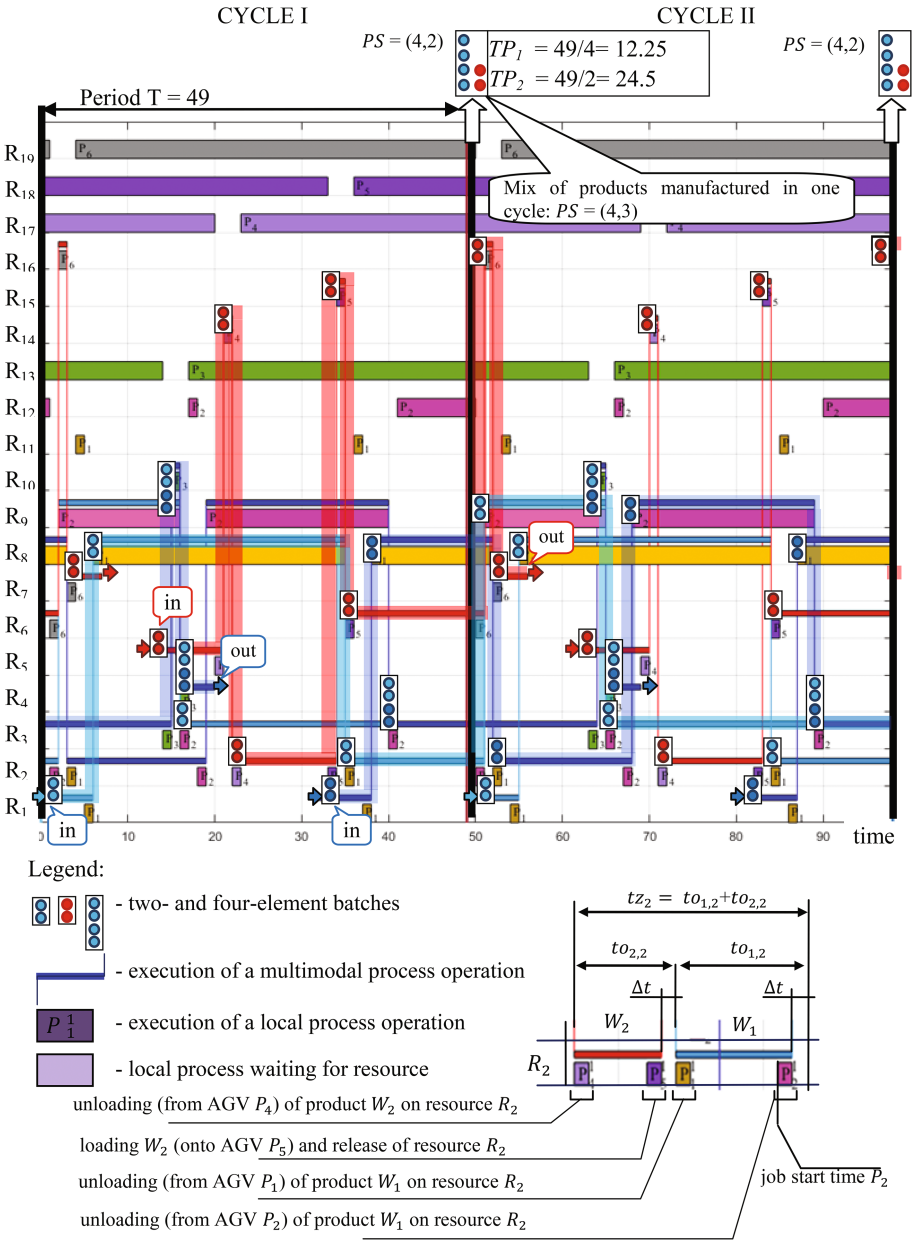


Fig. 2. A Gantt chart of production flow in the system shown in Fig. 1

To illustrate the synthesis problem, let us consider the instance below: Given is a system with a known set of resources R for production of goods W. Known are times $ts_{i,k}$ and travel times between workstations Δt. Which batch sizes $b_{i,k}$ for each

production line guarantee the desired production takt times TP_i^*? It is easy to see that the problem formulated in this way can also be given by *CS*, whose search space size, similarly to the case above, is estimated by the exponential function of the form (10). It should be emphasized that, in the general case, depending on the form of the adopted set of constraints, the set of admissible solutions may be an empty set, i.e. there may be no batch sizes $b_{i,k}$ which guarantee the desired parameters of the production flow. To exclude such situations, the proposed approach assumes that the set of constraints relating behavior and structure parameters (constraints (1)–(9)) is known for a system meeting the assumptions formulated in Sect. 2. In other words, to find whether there exists a solution to a given synthesis problem, one needs to solve the following decision problem:

Given: $TP_i^*, ts_{i,k}, \Delta t, = 1...n; k = 1...m$
Decision variables: $b_{i,k}, i = 1...n, k = 1...m$
Goal functions: $TP_i \leq TP_i^*, i = 1...n$
Set of constraints:

$$C = \begin{cases} TP_i = \frac{T}{bmax_i}, i = 1...n \\ T = max\{tz_1, \ldots, tz_k, \ldots, tz_m\}, k = 1...m \\ tz_k = \sum_{i \in Q_k} to_{i,k}, i = 1...n, k = 1...m \\ to_{i,k} = ts_{i,k} \cdot bmax_i + \min\{v_{i,k}, v_{i,k+1}\} \cdot \Delta t, i = 1...n, k = 1...m \\ v_{i,k} = \frac{bmax_i}{\min\{b_{i,k}, b_{i,k-1}\}}, i = 1...n, k = 1...m \\ bmax_i = \max\{b_{i,1}, \ldots, b_{i,k}, \ldots, b_{i,n}\}, i = 1...n, k = 1...m \\ u_k = \min_{i \in Q_k}\{b_{i,k}\}, i = 1...n, k = 1...m \end{cases} \quad (11)$$

The computational complexity of this problem, as previously stated, is estimated by an exponential function (10). In contrast to the previous problem, however, the search space in the considered system is smaller: $q = 8$, $d = 4$ (among others, relative to the previously considered case, where: $q = 32$, $d = 50$). In the problem discussed, the variables determining operation start times x are omitted (similarly to the cases from [3–5]) and a less numerous set of variables $b_{i,k}$ (batch sizes) is used in their place. In other words, the existence of a cyclic behavior with the desired production takt times TP_i^* is assessed solely on the basis of system structure variables.

A limitation of this approach is that, while the set of constraints that form the system of nonlinear diophantine equations offer the possibility of reducing the amount of computational effort, in certain cases (depending on the values of the variables $b_{i,k}$), the system of equations may be inconsistent, resulting in an empty set of admissible solutions. This means that also in this case, to solve the problem of decidability, it is enough to solve an appropriate corresponding *CS* problem. An alternative approach makes it necessary to determine sufficient conditions, the fulfilment of which guarantees the existence of admissible solutions. For the sake of further analysis, let us adopt the following formulation of the investigated reverse (synthesis) problem with reference to the *CS* structure discussed so far:

$$CS = (B, D, C) \tag{12}$$

where: $B = \{b_{i,k} | i = 1...n, k = 1...m\}$ is a set of decision variables (batch sizes), $D = \{d_{i,k} | i = 1...n, k = 1...m\}$ is a set of domains of variables $b_{i,k}$, and $C -$ (11).

5 Computational Experiment

Returning to the previous example, let us consider a situation in which the size of production batches $b_{i,k}$ is the same for every workstation operation on each product W_i, i.e., $B_i = b_{i,1} = b_{i,2} = b_{i,3} = b_{i,4} \in \{1, 2, 3, 4\}$, $i = 1, 2$. Assuming that the previously determined production takt times $TP_1 = 12.25$ u.t. and $TP_2 = 24.5$ u.t. provide constraints on production flow, we will be looking for an answer to the question of whether batch sizes B_1 and B_2 which guarantee production takt times for products W_1 and W_2 of respectively $TP_1 \leq 13$ u.t and $TP_1 \leq 25$ u.t., exist?

In other words, the goal will be to find a production flow pattern (a cyclic schedule) which satisfies the expectations of the customers, such that at least two pieces of product W_1 and at least one piece of product W_2 are manufactured within a time interval corresponding to one cycle of the schedule. For the purpose of further discussion, it is assumed that demand is determined by the Part Set defined as sequence $PS = (a_1, \ldots, a_i, \ldots, a_n)$, the successive elements of which stand for the number of pieces of products W_1, W_2, … , W_n released in one production cycle. In the example discussed here, demand is $PS = (2, 1)$.

The set of admissible solutions to problem CS (12) (OzMozart environment, Intel Core 2 Duo 3 GHz, 4 GB RAM, computation time < 1 s), which encompasses solutions that satisfy the required constraints, contains three production batch size variants. The solution that guarantees the highest efficiency of the system, i.e. the solution with the smallest total production takt time value, is the variant in which products W_1 and W_2 are produced in batch sizes $B_1 = 4$ and $B_2 = 3$, respectively. For this variant, the individual product items flow at production takt times $TP_1 = 13$, $TP_2 = 17.33$ u.t.

To generalize this example, let us consider the system of Fig. 1, for which different ways of configuring the production flow are considered (in terms of production batch sizes $b_{i,k}$, $i = 1, 2; k = 1...4$) under constraints:

i. $b_{i,k} \in \{1, 2, 3, 4\}$ (the batches are limited in size – a maximum of four items)
ii. $TP_1 \leq 13$ u.t.
iii. $TP_2 \leq 17.33$ u.t. (TP_1 and TP_2 are constrained by the solutions for $B_1 = 4$ and $B_2 = 3$)

Table 2. Size of production batch $b_{i,j}$ of product W_i processed on resource R_k

$to_{i,k} = ts_{i,k} \cdot bmax_i + \min\{v_{i,k}, v_{i,k+1}\} \cdot \Delta t \cdot q_{i,k}$, $\Delta t = 1$			
	$bmax_1 = 4$		$bmax_2 = 3$
R1	$b_{1,1} = 1$	R5	$b_{2,1} = 1$
R2	$b_{1,2} = 4$	R2	$b_{2,2} = 3$
R3	$b_{1,3} = 4$	R6	$b_{2,3} = 3$
R4	$b_{1,4} = 1$	R7	$b_{2,4} = 1$

The goal is to find an answer to the following question: What production batch sizes $b_{i,k}$ guarantee the minimum total size (capacity) of the input/output buffer used?

The best of all feasible solutions obtained as the results of *CS* problem (12) resolved is shown in Table 2. The obtained production batch sizes guarantee that the system of Fig. 1 operates according to a cyclic schedule with period $T = 52$ u.t. and production takt times of respectively $TP_1 = 13$ u.t. and $TP_2 = 17.33$ u.t. It is worth noting that, at the total input/output buffer capacity of **15** (i.e., a capacity that is 10 items smaller than the one corresponding to the solution following $B_1 = 4$ and $B_2 = 3$), the production flow is the same as in the previous case. For the system parameters given in Table 2, a cyclic schedule was determined which represents the pattern of (transport and workstation) operations in the production flow under consideration. This chart was developed on the basis of the solution of the *CS* problem [3–5], i.e. with the omission of the analytical form (10). The schedule is shown in Fig. 3. As it is easy to note, the resulting product mix (Part Set) $PS = (4, 3)$ satisfies the production constraints given.

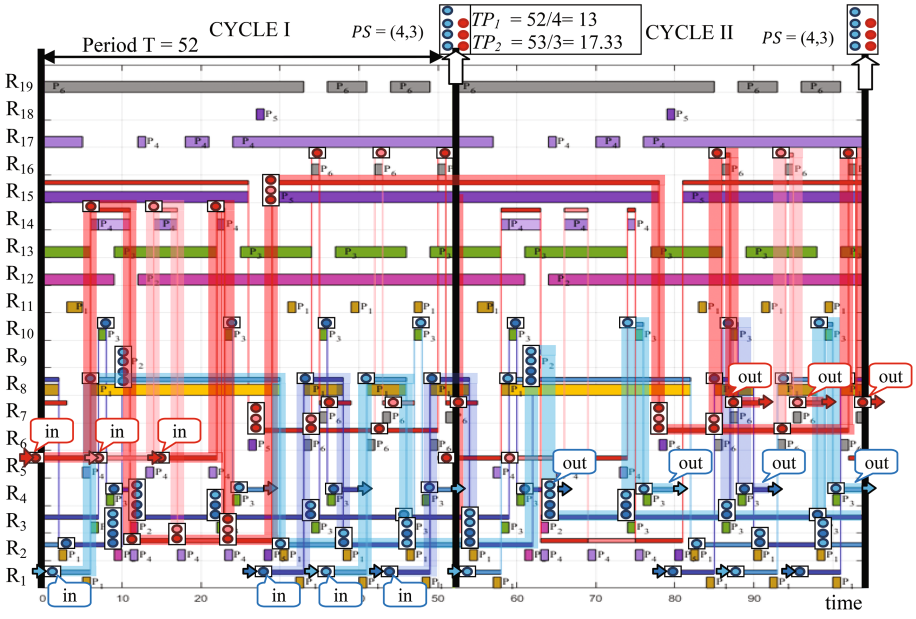


Fig. 3. A Gantt chart of production flow in the system from Fig. 1, with batch sizes from Table 2

6 Concluding Remarks

The approach presented in this study implicitly assumes that production batches of different products manufactured on a single workstation have either equal repetition periods or are integer multiples of one another. The model that adopted here enables

prototyping of the pattern of production flow. In particular, it allows one to evaluate alternative choices of number of batches or batch size (when input/output buffer capacities are given), or buffer capacity (when the number of batches are known).

Future research should focus on finding sufficient conditions that would ensure the consistency of the system of diophantine equations that comprise an analytical model of a production flow configuration. Apart from the research perspective presented in this article, other directions of study worth mentioning are those which take into account the uncertainty of processes and production order data as well as those related to the determination of interference-robust cyclic schedules.

References

1. Al-Ashhab, S.M.: An optimization model for multi-period multi-product multi-objective production planning. Int. J. Eng. Technol. IJET-IJENS **16**(01), 43–56 (2016)
2. Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., Węglarz, J.: Handbook on Scheduling: From Theory to Applications, p. 647. Springer, Berlin (2007)
3. Bocewicz, G., Banaszak, Z.: Scheduling for multimodal cyclic transport systems. Pomiary Automatyka Robotyka **2**, 106–113 (2013)
4. Bocewicz, G., Muszyński, W., Banaszak, Z.: Models of multimodal networks and transport processes. Bull. Pol. Acad. Sci. **63**(3), 635–650 (2015)
5. Bocewicz, G., Nielsen, I.E., Banaszak, Z.A.: Production flows scheduling subject to fuzzy processing time constraints. Int. J. Comput. Integr. Manuf. **29**(10), 1105–1127 (2016)
6. Bocewicz, G., Jardzioch, A., Banaszak, Z.: Modelling and performance evaluation of fractal topology streets network. In: Distributed Computing and Artificial Intelligence, Advances in Intelligent Systems and Computing, vol. 474, pp. 483–494 (2016)
7. Bożejko, W., Uchroński, M., Wodecki, M.: Block approach to the cyclic flow shop scheduling. Comput. Ind. Eng. **81**, 158–166 (2015)
8. Dobson, G., Yano, C.A.: Cyclic scheduling to minimize inventory in a batch flow line. Eur. J. Oper. Res. **75**, 441–461 (1994)
9. Kampmeyer, T.: Cyclic scheduling problems. Ph.D. Dissertation, Fachbereich Mathematik/Informatik Universit at Osnabruck (2006)
10. Korytkowski, P., Grimaud, F., Dolgui, A.: Exponential smoothing for multi-product lot-sizing with heijunka and varying demand. Manag. Prod. Eng. Rev. **5**(2), 20–26 (2014)
11. Kubale, M., Nadolski, A.: Chromatic scheduling in a cyclic open shop. Eur. J. Oper. Res. **164**(3), 585–591 (2005)
12. Levner, E., Kats, V., de Pablo, D.A.L.: Cyclic Scheduling in robotic cells: an extension of basic models in machine scheduling theory. In: Multiprocessor Scheduling: Theory and Applications, pp. 1–20. Itech Education and Publishing (2007)
13. Moreno, M.S., Montagna, J.M.: A multiperiod model for production planning and design in a multiproduct batch environment. Math. Comput. Model. **49**(7–8), 1372–1385 (2009)
14. Munier, A.: The basic cyclic scheduling problem with linear precedence constraints. Discret. Appl. Math. **64**(3), 219–238 (1996)
15. Pascal, D.: Lean Production Simplified, 2nd edn. CRC Press, New York (2007)
16. Sitek, P., Bzdyra, K., Wikarek, J.: A hybrid method for modeling and solving supply chain optimization problems with soft and logical constraints. Math. Prob. Eng. **2016**, 16 (2016). doi:10.1155/2016/1532420

17. Tamimi, A.M., Abidi, M.H., Mian, S.H.: Analysis of performance measures of flexible manufacturing system. J. King Saud Univ. Eng. Sci. **24**(2), 115–129 (2012)
18. Voudouris, V.T., Grossmann, I.E.: Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. Ind. Eng. Chem. Res. **32**(9), 1962–1980 (1993)

Quantitative Methods of Strategic Planning Support: Defending the Front Line in Europe

Andrzej Najgebauer(✉), Ryszard Antkiewicz, Dariusz Pierzchała, and Jarosław Rulka

Faculty of Cybernetics, Military University of Technology, Kaliskiego Str. 2, 00-908 Warsaw, Poland
{andrzej.najgebauer, ryszard.antkiewicz, dariusz.pierzchala, jaroslaw.rulka}@wat.edu.pl

Abstract. The paper is devoted to a quantitative approach to support of strategic planning to control international conflicts. The MUT team, taking part in Polish-American "tabletop" games, has proposed a set of methods and tools to analyze possible moves to resolve a hypothetical conflict along a front line in northeastern Europe. Both the model and some experimental results are presented in the paper. Decision support was proposed as a set of statistical, optimization and simulation tools for conflict scenario preparation and analysis. The range of quantitative methods and the choice of appropriate research tools in the experiments are described. The strategic game was divided into two phases: D1 – escalation of the conflict in the B states – determination of effectiveness measures, such as the opponent's estimation of the probability that the conflict will escalate (Bayesian Network Model); D2 – conventional conflict: Evaluation of effectiveness measures on our side (evaluation of force potential and structure required, simulation analysis of combat clashes). The presentation below focuses on D1.

Keywords: Decision support · Cast logic · Strategic planning

1 Introduction

Many supporting tools have been developed for making complex political, economic or military decisions. A quantitative evaluation of different variants of complex decisions is especially important because it allows one to compare variants and to choose the best one. We are especially interested in decisions connected with planning of complex operations. The implementation such operations is very often a non-deterministic process, and we need an appropriate model to evaluate measures of its quality. In this paper, we propose a stochastic network model of complex activities (operations) that applies the concept of CAST Logic [1, 2, 5, 8] and stochastic PERT analysis [1]. Application of CAST Logic enables modelling of causal and stochastic relations between series of activities. The concept of CAST Logic was presented for the first time in this paper [2] and subsequently developed in many others [4–8].

Especially important is paper [8], which removes some shortcomings of earlier versions of CAST logic implementation in influence networks and takes account of

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_26

time-dependency in complex operations. In this paper, we modeled time-dependency using a stochastic PERT model and a CAST logic model in compliance with paper [8]. An interesting proposal for modeling stochastic dependences between actions was given in the paper [3].

This paper is organized as follows: Sect. 2 presents our model and its implementation as an MCA package; Sect. 3 focuses on an analysis of political-military conflict based on MCA package, and Sect. 4 draws some summary conclusions.

2 Description of the Research Tools

The MCA software package for (M)odeling (C)omplex (A)ctivities is a specific conflict scenario editor/simulator with many embedded tools, based on a network model that is briefly described below. It was designed based on the legacy version mentioned in [1].

We have assumed that Complex Activities on each side of the conflict can be defined as the following network according to COA in [1]:

$$S_p = \langle G, \Phi, \psi \rangle \tag{1}$$

where:

$G = \langle D, U \rangle$, G – Berge's graph without loops, D – set of graph vertices, where vertices are related to decisive points of Complex Activities, $D = \{x^p, 1, 2, ..\bar{d}, x^k\}$,

U – set of graph edges – edges define the sequence of reaching the decisive point,

$\Phi = \{f | f : D \rightarrow R\}$ – family of functions defined at graph vertices,

$\Phi = \{\tau_{min}, \tau_{max}, \tau_p, pb, F, pr, PR\}$ and $\Psi = \{f | f : U \backslash \{(d,k) : k = x^k\} \rightarrow [-1,1]\}$ – family of functions defined at graph edges, $\Psi = \{h, g\}$.

We assume that values of the following functions are determined by planners during the planning process:

$\tau_{min}(d)$ – the shortest time needed to complete tasks to achieve decisive point d,

$\tau_{max}(d)$ – the longest time needed to complete tasks to achieve decisive point d,

$\tau_p(d)$ – the most probable time needed to complete tasks to achieve decisive point d,

$pb(d)$ – baseline probability of completing tasks to achieve d, $g((d,k))$ – the value of this function reflects the negative influence of achieving decisive point d on the probability of achieving decisive point k, $h((d,k))$ – the value of this function reflects the positive influence of achieving decisive point d on the probability of achieving decisive point k.

By applying the above model, we will further show the method of evaluating three important quantitative characteristics of Complex Activities in Conflict progress:

$F(d,t)$ – probability of completing tasks that allow the achievement of decisive point d in a time shorter than t, completion of the above tasks does not guarantee achieving the decisive point,

$pr(d)$ – probability of achieving decisive point d in unlimited time,

$PR(d,t)$ – probability of achieving decisive point d in a time shorter than t.

The idea of the $pr(d)$ evaluation is based on CAST logic described in paper [1]. According to CAST algorithm, we should set the following values as input data:

$pb(d)$ – the baseline probability of completing tasks that allow decisive point d to be achieved; this is the probability of achieving d without the influence of any other events,

$g((d,k))$ – the value of this function reflects the negative influence of achieving decisive point d on the probability of achieving decisive point k,

$h((d,k))$ – the value of this function reflects the positive influence of achieving decisive point d on the probability of achieving decisive point k.

Determination of values of the above functions is a separate problem. It is mostly assumed that they are fixed by experts applying their own knowledge and experience.

Let us denote $Pred(k) = \Gamma^{-1}(k)$.

If $\Gamma^{-1}(k) = \{d\}$, then it is assumed that:

- if decisive point d is achieved than:

$$P(k|d) = \begin{cases} pb_k + h(d,k) \cdot (1 - pb_k), \text{ if } h(d,k) \geq 0 \\ pb_k + h(d,k) \cdot pb_k, \text{ if } h(d,k) < 0 \end{cases}$$

- if decisive point d is not achieved than:

$$P(k|\sim d) = \begin{cases} pb_k + g(d,k) \cdot (1 - pb_k), \text{ if } g(d,k) \geq 0 \\ pb_k + g(d,k) \cdot pb_k, \text{ if } g(d,k) < 0 \end{cases}$$

where $P(k|d)$ means the conditional probability of achieving point k, given that point d was achieved ($P(k|\sim d)$ d not achieved).

In case $\left|\Gamma^{-1}(k)\right| > 1$, we have defined additional quantities:

$$y^k = (y_d)_{d \in \Gamma^{-1}(k)},$$
$$Y^k = \left\{ y^k = (y_d)_{d \in \Gamma^{-1}(k)} : y_d \in \{0,1\} \right\}$$

where:

$$y_d = \begin{cases} 1 & \text{if point } d \text{ achieved} \\ 0 & \text{otherwise} \end{cases}$$

$$h_d^k(y_d) = \begin{cases} h(d,k) & y_d = 1 \\ g(d,k) & y_d = 0 \end{cases}$$

$$h(y^k) = [\prod_{d \in \Gamma^{-1}(k): h_d^k(y_d) < 0} \left(1 - \left|h_d^k(y_d)\right|\right) - \prod_{d \in \Gamma^{-1}(k): h_d^k(y_d) \geq 0} \left(1 - \left|h_d^k(y_d)\right|\right)]$$
$$/max[\prod_{d \in \Gamma^{-1}(k): h_d^k(y_d) < 0} \left(1 - \left|h_d^k(y_d)\right|\right), \prod_{d \in \Gamma^{-1}(k): h_d^k(y_d) \geq 0} \left(1 - \left|h_d^k(y_d)\right|\right)].$$

Now we can calculate conditional probability $P(k \mid y^k)$ using the following formula:

$$P(k|y^k) = \begin{cases} pb_k + h(y^k) \cdot (1 - pb_k), \text{ if } h(y^k) \geq 0 \\ pb_k + h(y^k) \cdot pb_k, \text{ if } h(y^k) < 0 \end{cases}.$$

The unconditional probability *pr(k)* of achieving the decisive point k is computed via the expression:

$$pr(k) = \sum_{y^k \in Y^k} P(k|y^k) \cdot \prod_{d \in \Gamma^{-1}(k)} pr(d)^{y_d} \cdot (1 - pr(d))^{1-y_d}.$$

Applying formulas for *pr(k)* and *F(d,t)*, we could compute the probability of achieving decisive point d in a time shorter than t via the expression:

$$PR(d,t) = F(d,t) \cdot pr(k).$$

The editor's implementation was conducted with recommendations based on object-oriented methodologies. The main programming language is Java. There are two versions of the software package. The first version was prepared as the standalone application for ease of use. The second version was implemented as an applet and deployed on a server in a set of decision support services.

3 Computational Experiments

3.1 Network Model of Complex Activities the Sides of Conflict

As an example, we propose to consider a conflict situation between the sub-coalition *N*, where the countries *U, P, B (B = Li, L, E)* are part of *N*, and the opposing state *R*, including exclave *K*, in coalition with *Be*. The main goal of *R* is to destabilize the internal situation in state *L* – the context is protecting *R* minorities in *L*. The main goal of sub-coalition *N* is to stop the escalation of the conflict in *B*.

Overarching objectives of *R*:

- Conduct operations in *L* and *Li* that will split *N* and further erode confidence in *N's* ability to defend its flank;
- Create a strategic dilemma for *N* by conducting sub-conventional operations to cause *N* to deploy and sustain forces in the *B* states, then use flanking capabilities in the *K* exclave and *Be* to pressure *N* lines of communication (LOCs) by land, sea and air;
- Reinforce the *K* exclave and *Be* to "harden" them against *N* retaliatory actions.

Overarching objectives of *N*:

- Provide foreign internal defense (FID), other military support to *L*;
- Conduct limited operations to protect air, land and sea lines of communication to *L*;
- Defend *P* borders against SOF raids from the *K* exclave and *Be*;
- Deter *R* conventional invasions of *P* and *B*.

To achieve these objectives, each side of the conflict can make moves described in the model presented in point 2. The influence of these moves is expressed by the evolution of the unconditional probability of achieving a decisive point (in other words, the move's success).

The assumptions

1. A collection of values of the function *h(g)*:
   Decisive impact: *h = 1*; Very strong impact: *h = 0,9*; Strong impact: *h = 0,75*; Average impact: *h = 0,5*; Weak impact: *h = 0,25*; Very weak impact: *h = 0,1*; No impact: *h = 0*;
2. Time of implementation steps:
   a. times are expressed in days;
   b. the nodes corresponding to the countries, and not the operations, assign lead times *c = [0, 0, 0]*;
3. *PB* – the baseline probability of completing tasks without the influence of any other events (Fig. 1).

The decisive point 1, or so-called "*Conflict escalation in L*", can be also called the end-state. The secondary factors also include decisive points:

– Preventing the escalation of the conflict situation in *L*,
– Reinforcement of the borders of country *P*;
– Protection of lines of communication (LOC's) to country *L*;
– Deterrence of country *R's* conventional military invasions of *P* and other countries;
– Inconsistent reactions of third-party countries;
– Expected losses of country *R* resulting from conflict escalation;
– Improvement of the authority image of *R*,

which can affect end-state 1. Each of these decisive points can be affected by primary factors.

Time is described in vector *c = [The shortest time of completion of tasks, The most probable time of completion of tasks, The longest time of completion of tasks]*.

Preventing the escalation of the conflict situation in *L*:

– Reinforcement of *L's* security forces by the *N* (+) (*PBA = 0,4; h = 0,9; g = −0,25; c = [0, 0, 0]*);
– Deployment of third-party (sub-coalition of *N*) special forces in country *L* and special forces operations in ports and bridges, (*PBAi = 0,8; h = 0,9; g = 0,0; c = [3, 6, 9]*);
– Actions of *R* special forces and pro-*R* paramilitary groups ("green men") in *L* and *Li* (*PBAii = 0,8; h = −0,5; g = 0,0; c = [0, 1, 3]*);
– Information war (against incitement of the *R* population) (+) (*PBAiii = 0,75; h = 0,5; g = −0,75; c = [1, 3, 6]*), R actions (−), N actions (+);
– Cyber defense (+) (*PBAiv = 0,5; h = 0,5; g = −0,75; c = [1, 3, 6]*).

Reinforcement of country *P*'s borders (*PBB = 0,5; h = 0,5; g = −0,25; c = [0, 0, 0]*):

– Deployment of ground forces along the borders (+) (*PBBi = 0,9; h = 0,9; g = 0,0; c = [3, 6, 9]*);
– Monitoring of borders by technical means (+) (*PBBii = 0,75; h = 0,75; g = 0,0; c = [1, 3, 6]*).

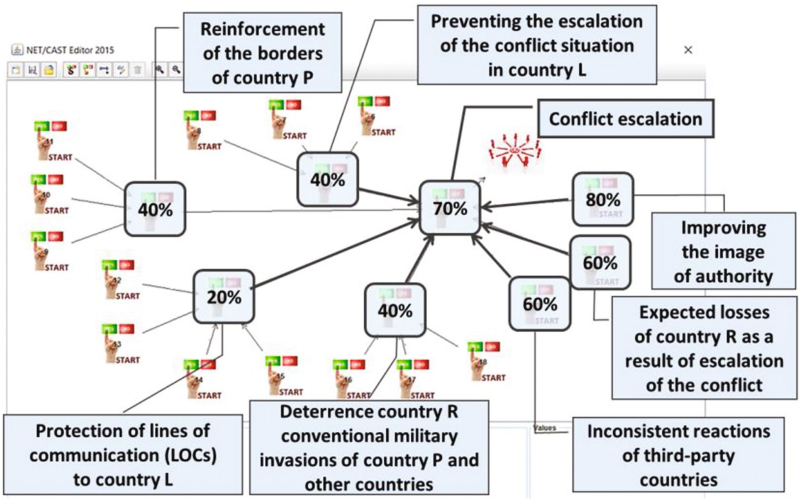


Fig. 1. Initial state of conflict – baseline probability of completing tasks

Protection of lines of communication (LOCs) to *L* (*PBC = 0,2; h = 0,5; g = −0,3; c = [0, 0, 0]*):

- Monitoring and reinforcement of borders (+) (*PBCi = 0,75; h = 0,5; g = −0,1; c = [1, 6, 9]*);
- Monitoring of communication lines (+) (*PBCii = 0,9; h = 0,5; g = −0,1; c = [1, 3, 6]*);
- Reinforcement of naval and air patrols (+) (*PBCiii = 0,75; h = 0,5; g = −0,1; c = [1, 3, 6]*);
- Deployment of land forces to protect lines of communication (+) (*PBCiv = 0,6; h = 0,8; g = −0,2; c = [5, 7, 9]*).

Deterrence of *R* conventional military invasions of *P* and *B* countries (+) (*PBD = 0,25; h = 0,75; g = −0,5; c = [0;0;0]*):

- Increasing combat readiness of forces in *P* and *B* countries (+) (*PBDi = 0,9; h = 0,25; g = −0,25; c = [1, 3, 6]*);
- Reinforcement of the armed forces in *P* and *B* countries by the contingent of *N* (+) (*PBDii = 0,75; h = 0,75; g = −0,75; c = [3;6;9]*, taking into account just deployed forces);
- Actions of *R*, intensive exercises in the region, cyber actions, force deployment (−) (*PBDiii = 0,75; h = −0,5; g = 0,2; c = [0, 3, 6]*).

Military reinforcement of *K* exclave and *Be* (−) (*PBE = 0,75; h = −0,75; g = 0,5; c = [0;0;0]*):

- Reinforcement of missile (ground-air, ground-ground) units near the border and *Be* (+) (*PBEi = 0,9; h = 0,9; g = −0,5; c = [7;14;14]*);
- Intensive movements and reinforcement of missile units in border countries to increase effective range (+) (*PBEii = 0,75; h = 0,25; g = −0,25; c = [5;7;10]*);

– Military exercises near the western border of *Be* (+) (*PBEiii = 0,75; h = 0,25; g = −0,25; c = [7;10;14]*);
– Blocking the creation of a corridor into the *K* exclave by *N* activities in *P* and *Li* (−) (*PBEiv = 0,75; h = −0,5; g = 0,75; c = [5;10;15]*).

3.2 Experiments' Outcomes

The analytical process is illustrated by the following graphs – Figs. 2, 3, 4, 5 and 6.

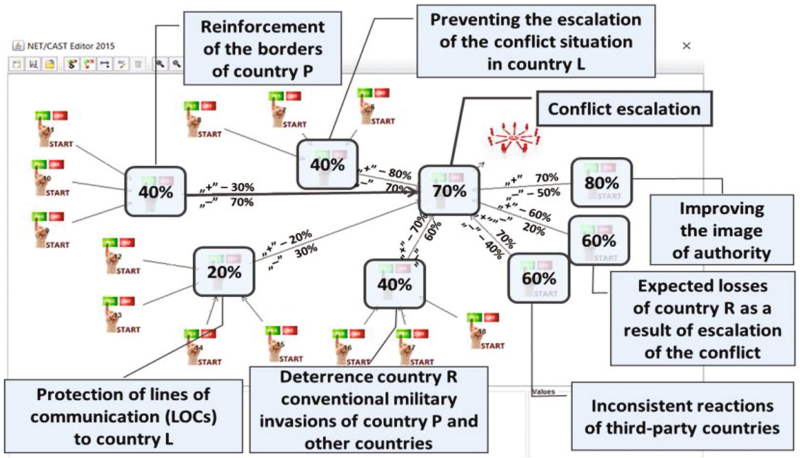


Fig. 2. The analysis of positive or negative influences of preceding decisive points on the conflict escalation

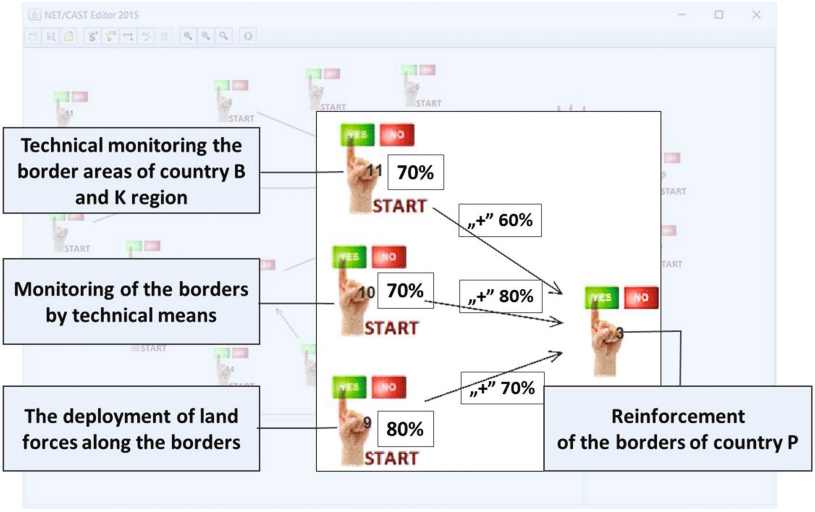


Fig. 3. Factors of chosen decisive point

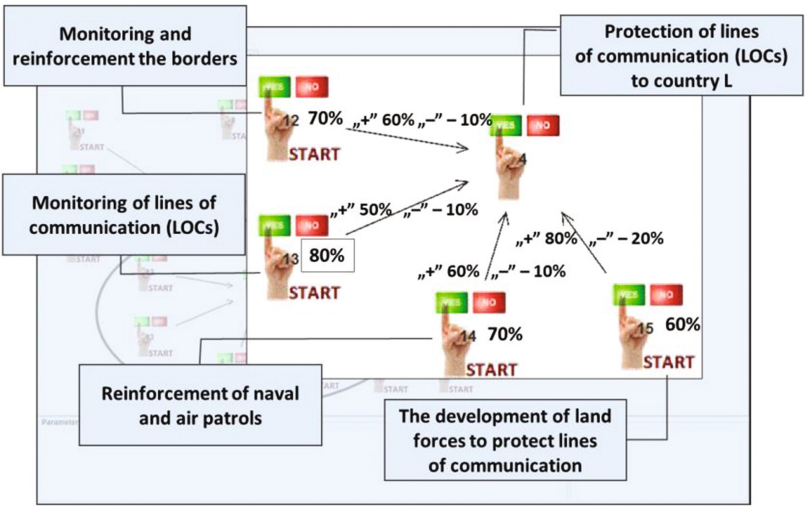


Fig. 4. Influences on protection of lines of communication

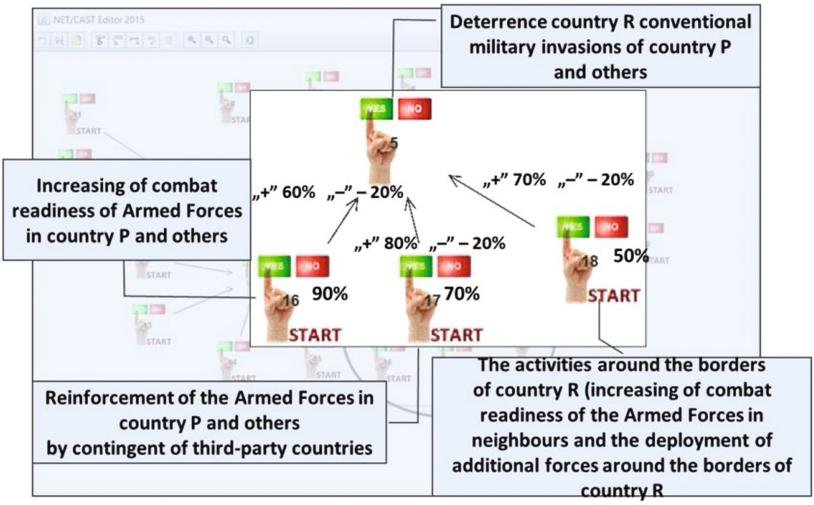


Fig. 5. Deterrence

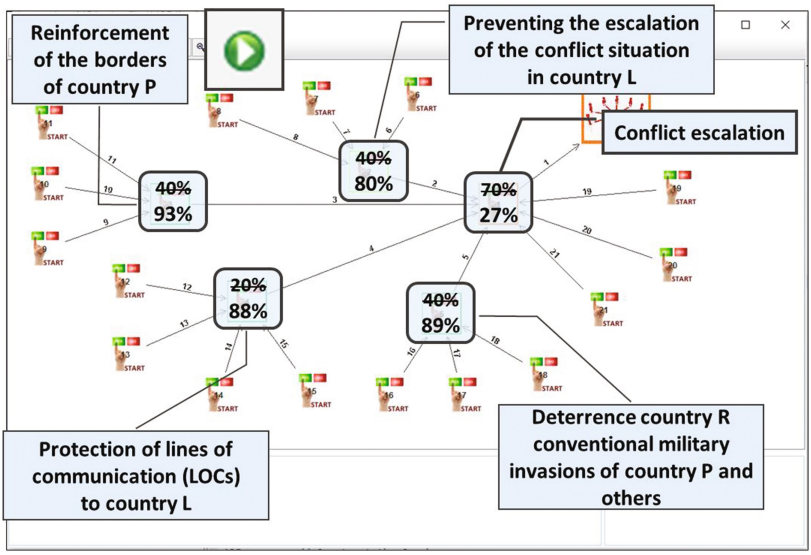


Fig. 6. Final evaluation of probability of conflict escalation (reduction)

4 Conclusions

This paper presents a set of research tools to support strategic analysis in crisis situations. The solutions offered by the war game carried out by an international Polish-American team and the board-game simulation for the Polish Ministry of Defense undoubtedly require further evaluation using supporting tools and analysis. Studying concepts for using Allied forces with various weapons systems and combat methods to stabilize the situation in the affected region requires verification through quantitative methods, including statistics, optimization and simulation.

We have made calculations and carried out experiments for the conflict's first phase (destabilization of L and subsequently the other B states). First, we presented the method and the results of a network analysis of complex strategic actions taken on a variant proposed by the American team. In this variant, the probability of escalation, even with appropriate countermeasures, was 0.7 over a period of 13 days.

In a variant proposed by our team, correct moves reduce the probability of conflict escalation to 0.27 in 13 days. In order to assess the chances of blocking an operation to carve out a corridor to exclave K, we used the optimization calculation method and simulation in addition to network analysis. This allowed us to estimate the potential losses and structure of friendly forces depending on the opponent's decisions. The proposed method is extremely flexible and can be used to support the development of CONOPS (concept of operations) in various conflicts.

In conclusion, it should be noted that the methods presented in this paper need constant calibration and testing. A significant advantage they offer is the opportunity for flexible research and simulation of conflicts at various stages. This can prove

helpful in making decisions about developing force structures, including selecting weapons systems appropriate to the needs and capabilities of our armed forces.

References

1. Antkiewicz, R., Gąsecki, A., Najgebauer, A., Pierzchała, D., Tarapata, Z.: Stochastic PERT and CAST logic approach for computer support of complex operation planning. In: ASMTA 2010, Lecture Notes in Computer Science, vol. 6148, pp. 159–173. Springer, Heidelberg (2010)
2. Chang, K.C., Lehner, P.E., Levis, A.H., Zaidi, A.K., Zhao, X.: On causal influence logic. Technical report, George Mason University, Center of Excellence for C31 (1994)
3. Drużdżel, M.J.: Rapid modeling and analysis with QGENIE. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 4, pp. 157–164 (2009)
4. Haider, S., Levis, A.H.: Effective courses of action determination to achieve desired effects. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **37**(6), 1140–1150 (2007)
5. Rosen, J.A., Smith, W.L.: Influence net modeling with causal strengths: an evolutionary approach. In: Proceedings of the Command and Control Research and Technology Symposium, Naval Post Graduate School, Monterey, 25–28 June 1996
6. Wagenhals, L.W., Levis, A.H.: Course of action development and evaluation. In: Proceedings of the 2000 Command and Control Research and Technology Symposium (2000)
7. Wagenhals, L.W., Levis, A.H.: Course of action analysis in a cultural landscape using influence nets. In: Proceedings IEEE Symposium on Computational Intelligence for Security and Defense Applications, Honolulu (2007)
8. Zaidi, A.K., Mansoor, F., Papantoni-Kazakos, T.P.: Theory of influence networks. J. Intell. Rob. Syst. **60**, 457–491 (2010)

Implementing BPMN in Maintenance Process Modeling

Małgorzata Jasiulewicz-Kaczmarek[1], Robert Waszkowski[2(✉)], Mariusz Piechowski[3], and Ryszard Wyczółkowski[4]

[1] Poznan University of Technology, Strzelecka St. 11, Poznan, Poland
[2] Military University of Technology, Kaliskiego St. 2, 00-908 Warsaw, Poland
robert.waszkowski@wat.edu.pl
[3] IQ Software, Koronna St. 3, 60-652 Poznan, Poland
[4] Silesian University of Technology, Akademicka St. 2A, Gliwice, Poland

Abstract. Due to the fact that the quality of the entire process is more and more dependent on the maintenance process, must be carefully designed and effectively implemented. There are various techniques and approaches that may be applied for efficient and effective maintenance management. This study presents the application of the modeling tools and process management within BPM practices in the maintenance processes in a medical devices manufacturing company.

Keywords: Maintenance process · Business process management · Business process modeling

1 Introduction

There is a continual focus on cost reduction to attain increased revenue generation [10]. Therefore, due to its role in the corporate long-term profitability, more and more significance is put on maintenance. Cost reduction within the maintenance organization does not concern to reduce the quality or the level of service. It concerns an increased control of the maintenance organization and also related areas. In order to control the maintenance organization properly information about occurring events are needed [4]. Data is the foundation for conclusions and decisions, without effective data gathering cannot incidents be truly investigated and defined. To gather and analyze data manually a tremendous amount of both time and effort is required. Due to this, many companies develop and use computer programs in this field. These programs, or systems, are called computerized maintenance management systems (CMMS) and they are designed to gather all data related to maintenance and to file it in the history of corresponding asset [2, 13]. One of the first steps in the maintenance process automation is to define a business process model with all its elements.

This study presents the application of the modeling tools and process management according to the BPM practices in the maintenance processes for a medical devices manufacturing company. The paper is organized as follows. Section 2 describes the maintenance management issues. Section 3 describes the business process modeling

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_27

issues. Section 4 presents the case study based on maintenance processes in medical devices manufacturing company. Section 5 presents our conclusions.

2 Maintenance Management Framework

The production and its operational aspects such as quality, costs, capacity, safety and environment are influenced by maintenance of the equipment [7]. Therefore, due to its role in the corporate long-term profitability, more and more significance is put on maintenance. Many industries today are focusing their attention on the maintenance as the business function. Maintenance effectiveness, to a large extent, depends on the quality of the knowledge of the managers and maintenance operators, and the effectiveness of internal & external collaborative environments. But, maintenance effectiveness depends also on the quality, timeliness, accuracy and completeness of information related to machine degradation state, based on which decisions are made. To monitor the condition of the equipment, the equipment data needs to be collected and processed, providing basis for the planning and decisions [5]. Different data sources are used in maintenance decision making processes. These data sources are for example failure data, practical experience, results of technical analysis, condition-monitoring

Table 1. Causes of uncertainty [15]

Cause	Description	Impact
Lack of information	Decision maker does not have the information needed	A situation of uncertainty can be transformed to a situation of certainty by gathering more and better information
Abundance of information (complexity)	More data is available than a person can digest	A transfer to certainty cannot be achieved by gathering more data, but rather by transforming the available data to appropriate information
Conflicting evidence	There might be information available pointing to a certain direction, but some information is available pointing to another direction	An increase of information does not reduce uncertainty. Checking and deleting wrong information might help to transform the situation to certainty
Ambiguity	Linguistic information has entirely different meanings	Ambiguity can be classified also as lack of information because more information helps to move towards certainty
Measurement	An 'imagined' exact property cannot be measured perfectly	This can also be considered as a lack of information
Belief	All information available to the observer is subjective as a kind of belief in a certain situation	This situation is questionable and it can be considered as lack of information

measurements and operating data, or a combination of these data resources. It is necessary to have the correct data, since it is essential in assisting the maintenance actor in the decision-making process [1, 6, 14]. Any kind of information gaps make it difficult to make the right decision in a particular situation and thus increase uncertainty. According to Zimmerman [15], "Uncertainty implies that in a certain situation a person does not dispose about information which quantitatively and qualitatively is appropriate to describe, prescribe or predict deterministically and numerically a system, its behavior or other characteristic." Details of the causes of uncertainty are presented in Table 1.

To manage the collected data efficiently, the maintenance process requires the certain level of computer support [9, 11]. The computerized maintenance process includes the organization of maintenance data, work order planning and scheduling, labor information, reporting and analysis, and inventory management [8]. One of the first steps in the maintenance process automation is to define a business process model with all its elements.

3 Business Process Modelling

Business process presents a set of connected activities which jointly create the output which has value for the customer. Process activities as the elements of the business processes are realized by the process participants. Process participants can be divided into process owners which do the process planning, organization and monitoring, and into participants who carry out the process tasks. Another element of the business process is the event, which represents the result of the activity and also it can be the cause for beginning the certain activity. Every process can have its performance indicators exemplified in different forms, such as number of occurrences of certain event, time for activity execution, number of errors in the process, costs etc. [12]. To define a process, to analyze it, improve it and implement it in organization, the whole set of techniques and methods can be used, and they are unified under the paradigm of Business Process Management.

The Business Process Management is an oriented methodology for the identification, design, implementation, documentation, measurement and control of business processes, making them flow from end to end. Thus, the activities and tasks cross the functional barrier to add value to the customer.

Business process management includes Business Process Modeling (BMP) approach. BPM is mostly about organized process description, including tasks, activities and flows, with use of graphical methods. Process modeling goals include those connected with organization's performance and those connected with developing IT systems supporting management. Considering the first group of goals, modeling is usually focused on core processes, which are substantial for company. In such case, key processes are modeled and links between activities are analyzed. This enables identification of processes' flows and recognition of activities critical for processes. On the other hand, business process modeling for IT systems development requires mapping all the processes that are going to be included in the system. In such case, modeling is an element of IT system analysis that results in functional specification of the system. It includes formalized information on elements of the system, i.e. its structure,

processes, relations and links between data. Nevertheless, in both cases modeling is simplification of reality that allows to present processes clearly and helps in understanding complex relations between activities.

BPM approach is greatly dependent on IT which supports its realization. The most important solutions within the area are Business Process Management Systems (BPMS) and Workflow Management System (WMS). According to Workflow Management Coalition (WfMC) these systems manage workflow with software that manages processes, communicates with users and benefits from other applications and IT tools.

The most popular notation for processes modeling is the BPMN notation. The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the process, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor these processes. Thus, BPMN closes the gap between process design and process implementation. BPMN is an element supporting business processes modeling and can be successfully applied for modeling complex processes. Graphical presentation of processes makes their analysis and modification easier, at the same time providing a wide range of applications of the notation as the method that is basic for business process management.

Business process modeling should be the common solution for defining core activities in maintenance. Well defined processes benefit in limiting uncertainties in decision making processes, eliminating unnecessary activities, managing human and material resources in rational way, and in consequence limiting cost. Moreover, business process modeling should be the standard solution for designing, developing and implementing IT system for maintenance processes.

4 Maintenance Processes Modeling – Case Study

The object of the research is a medical devices manufacturing company. It is a multi-department company, benefiting from many production facilities, equipped with machines and production lines of diversified technological profile (largely associated with each other), constructed on the basis of various components: electrical, hydraulic, mechanical, hybrid and working 24 h. Due to planned implementation of the next mode of CMMS namely "Supervision and coordination system", it was necessary to analyze processes flows, verify them and develop new processes, which required application of BPMN.

The following chapter introduces modelling of "Recovery process". The modeling of the current process was based on interviews with employees of the maintenance, production and procurement departments. For the implementation of this work, the authors used the approach proposed by [3] (Fig. 1). Baldam et al. [3] have adopted a model, to describe the BPM methodology, considering four steps that comprise a BPM cycle. These steps are: BPM planning; modeling and optimization of processes; implementation procedures; and control and data analysis.

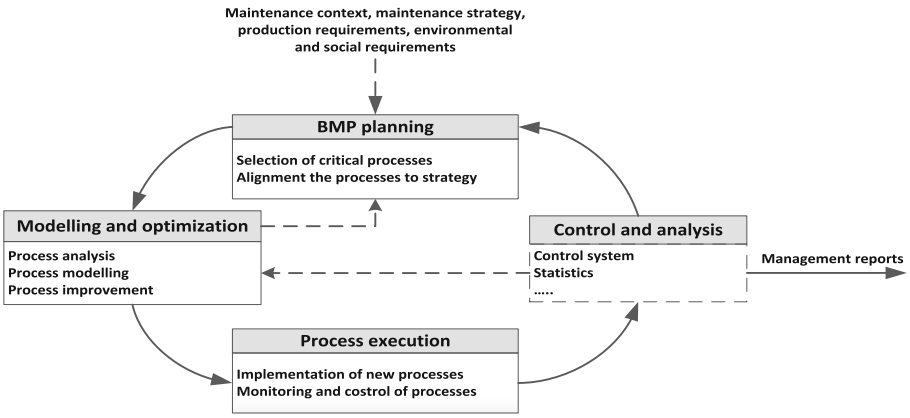


Fig. 1. BPM cycle [3]

BPM planning included analysis of the activities that contributes to the achievement of organizational objectives of the factory, specifically in the maintenance department, through the survey of the main weak points of the current maintenance process and the identification of improvement opportunities. For this survey, interviews were conducted with all participants of the "Recovery process", such as machine operator, coordinator of the production, maintenance technicians, maintenance department, purchasing department and others involved.

The next step "Modelling and optimization" includes two major activities: modelling the current state and the proposed state, which we used, the BPMN notation:

- modelling the current state (as is): construction of the model based on analysis of current documentation, interviews and brainstorming with those involved in the process;
- optimization and modelling of future state (to be): reduction of bureaucracy and duplicated tasks; reducing time; use simple language; activities standardization; definition of those involved in the process; definition of business rules; and use of information technology.

Three actors are involved in the "recovery process": production department (DP), maintenance department (DT) and quality department (DQ). Following, there are presented the details of each of the activities the recovery process is composed of. When reporting a failure, the reporting person (machine operator) determines the type of failure: mechanical, electrical, other, which should be understood as: type of failure unknown or simultaneous occurrence of mechanical and electrical failure). After a failure has been recovered, the DT responsible for recovery shall send the information to the DP. The production department runs a series of tests on a given machine to confirm the effectiveness of the repairs carried out. After the test series, the quality department (DQ) performs quality assessment of the defined product parameters. If the product meets the requirements, DQ approves DP production. The production department sends the information concerning the launch of the production to the

planning department (DPP). In case of a negative quality assessment, the quality department (DQ) will forward the information to the technical department (DT) expecting repairs to be completed correctly. The new process model is shown in Fig. 2.

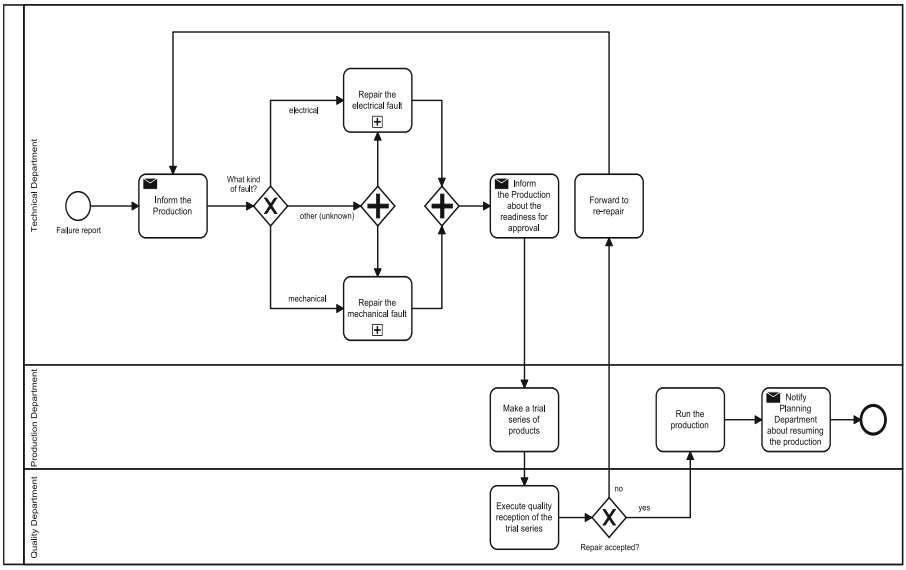


Fig. 2. BPMN diagram of the recovery process.

After the operator has reported a failure (Fig. 3), the recovery task can be accomplished in two ways: by using own resources (DT) or external resources (external service). In the first option, there is possibility of the lack of necessary resources for task execution, e.g. lack of parts. In such a case, the Procurement Department (DZ) will be asked to purchase and deliver those parts and, upon delivery, work is continued until it is completed. In most cases, all the resources necessary to eliminate the failure will be available and the task will take place in one time. In the second scenario, the task is transferred to an external company that takes over the entire process until the task is completed. Irrespective of the option selected, the flow of information across departments is an important part of the process, as the date of recovery can affect the tasks performed by other departments: DP – Production Department, DPP – Planning Department. In the model described, if a failure has not been recovered within 1 (or 2) h after notification, the departments indicated must take corrective action. After that time, the relevant information reaches the DP and DPP departments. The DPP department is taking steps to implement the procedures for changing the production plans, and the DP department, depending on the level of execution of the production order (85 or 95%), is preparing to execute the next order (setups, replenishing, …).

Process execution is the next stage. In this stage, proposed process was evaluated experimentally in a web environment. This was done by performing the following tasks: data modeling, generation of forms and definition of business rules and

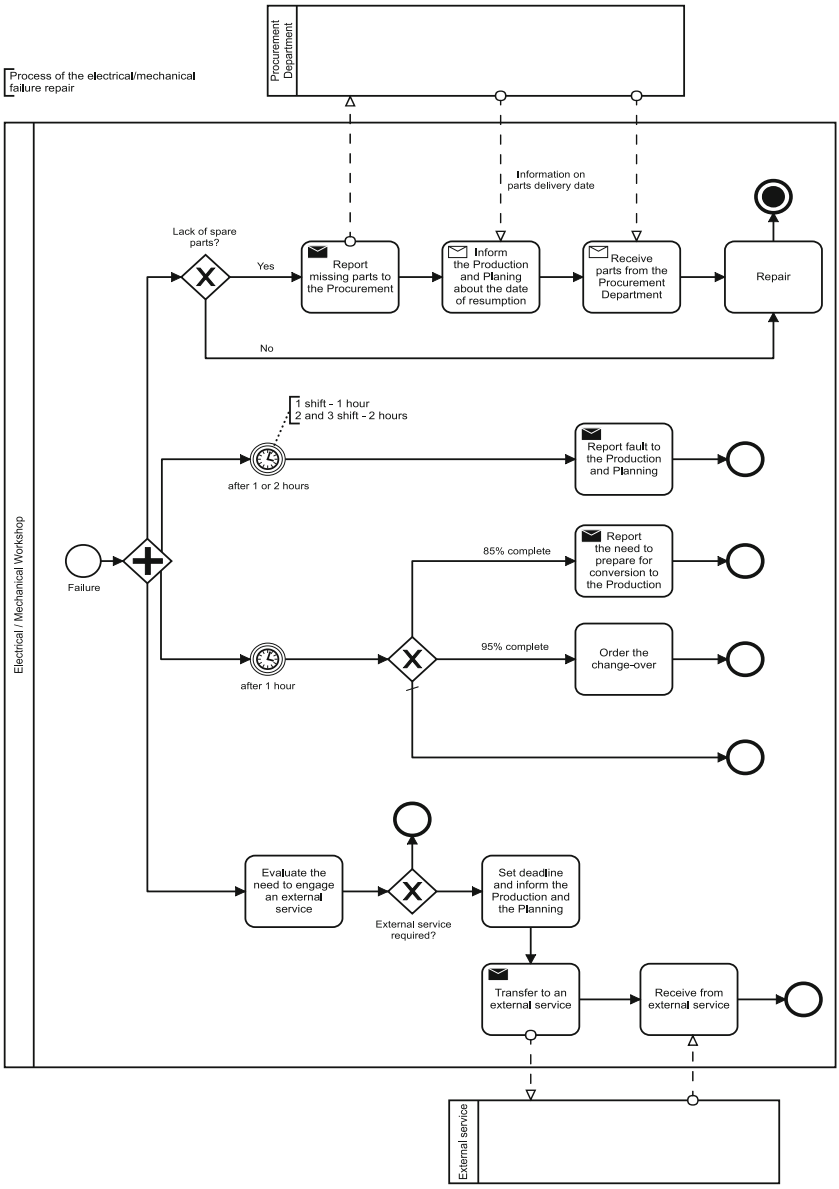


Fig. 3. BPMN diagram of the electrical/mechanical failure repair process.

participants. The data model contains all the information required in the process and how they relate to the entities. To perform the simulation management of the maintenance process, the following information was required: problem description; its location; phone; e-mail, comments on the requested service; analysis of received data;

observations on the need for correction of a request; technician name that will perform the service; check on the need of material request, etc.

Once set the process data model, the forms to be displayed to the users have been created. These forms were developed through attributes added to the master and parametric entities in the data model. Each activity has a specific form to be completed by the assigned user to do so. Figure 4 shows a specific example of the form used for the opening of a maintenance request.

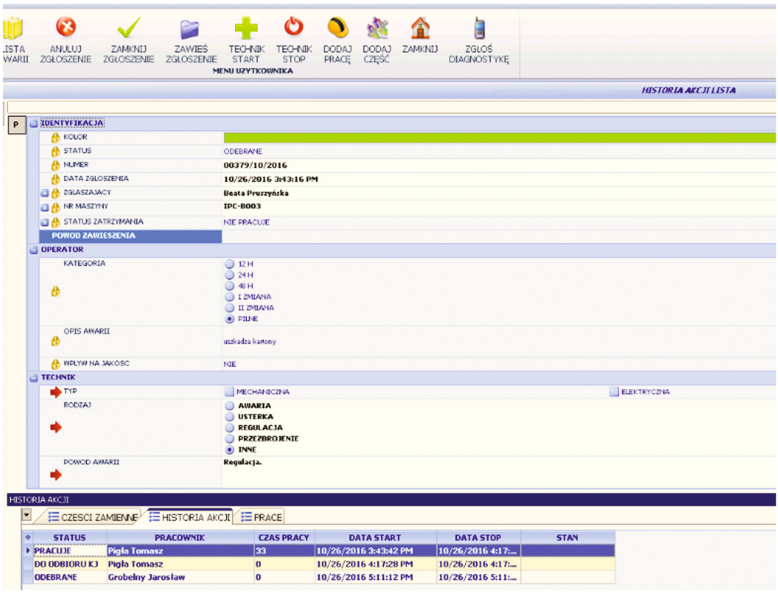


Fig. 4. Form for the opening of a maintenance request.

In the next step, the process was configured according to the rules of the factory. Each gateway defined in the process needs to be predetermined a condition to follow each path. Next, all the areas and workstations involved in the process were registered. For each task, a position responsible for its implementation was designated. Only the people assigned for this position have access to relevant data.

The final stage is "Process simulation and analysis". The business process management combined with the BPMS systems enables complete monitoring and evaluation process. This is the final stage of the work, in which process the benchmarks were analyzed with the objective of evaluating the proposed process. The resulting performance measurement information was used to process the feedback, reviewing the initial planning and again identifying improvement opportunities. Critical factors for the company were, inter alia, the timing of work, delays, the number of correct repairs (approved without quality department compliance), active working time of the maintenance staff, spare parts delivery time and the ability to determine the status of

operations performed on the machine. With the simulation process on a BPMS environment, these factors were analyzed and changes were made to the process.

The activities presented above were the part of the implementation of the next CMMS mode "Supervision and coordination system" in the company. By comparing historical data for 2015 (before implementation) and 2016 (after implementation), it can be stated, inter alia, that the time of the repair was shortened by on average 19%; Procurement delays have been reduced by 46%, thanks to information on tasks (mainly failures) provided to the planning and production department, which has allowed for effective revision of production plans; increased lead-time accuracy by approximately 43%, as well as reduced waiting time for spare parts (Fig. 5).

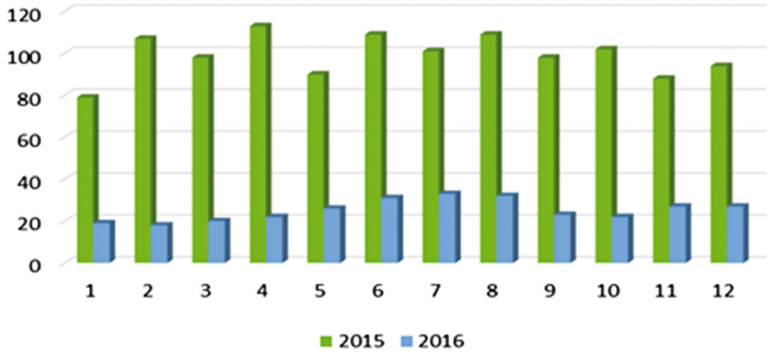


Fig. 5. Average waiting time for parts - before and after implementation (in hours)

5 Summary

An important prerequisite of successful process management is process-oriented thinking. This means that the main focus is not on the department-based division of tasks, but rather on individual employees as they are the ones to perform the relevant process steps. The paper includes presentation of the application of notation and method within the concept of management processes for the creation and evaluation of the maintenance process in a medical devices manufacturing company. By diagramming each of the activities that make up the maintenance process using the BPMN notation, this is disclosed in a graphic, clear and standardized manner, allowing that all involved are aware of the whole and can contribute with suggestions for optimization process. Using the BPMS system, each participant knows exactly the tasks that must perform and when toper form them, as it has an automatic time control system of the activities according to the rules and expectations of time previously set by the process owner.

It should be borne in mind that a good model of the process not only provides the opportunity to clearly illustrate the complex links between activities and better understand them. It can be a substitute research object and enable running experiments that cannot be executed on a real object - especially simulation and analysis of

long-term processes or critical-critical conditions such as the minimal resource of maintenance that is still able to handle event (failure) streams of specified intensity.

References

1. Aljumaili, M., Tretten, P., Karim, R., Kumar, U.: Study of aspects of data quality in eMaintenance. Int. J. COMADEM **15**(4), 3–14 (2012)
2. Bagadia, K.: Computerized Maintenance Management Systems Made Easy: How to Evaluate, Select, and Manage CMMS. McGraw-Hill Professional, New York (2006)
3. Baldam, R., Valle, R., Pereira, H., Hilst, S., Abreu, M., Sobral, V.: Gerenciamento de processos de negócios – BPM – Business Process Management. Ed. ÉricaLtda, São Paulo (2011)
4. Bartkowiak, T., Pawlewski, P.: Reducing negative impact of machine failures on performance of filling and packaging production line - a simulative study. In: Proceedings - Winter Simulation Conference, pp. 2912–2923 (2017). Article no. 7822326
5. Bjorling, S.E., Galar, D., Baglee, D., Singh, S., Kumar, U.: Maintenance knowledge management with fusion of CMMS and CM (2013). http://worldcomp-proceedings.com/proc/p2013/DMI8002.pdf
6. Campos, M.A.L., Márquez, A.C.: Modelling a maintenance management framework based on PAS 55 standard. Qual. Reliab. Eng. Int. **27**, 805–820 (2011)
7. Jasiulewicz-Kaczmarek, M.: Role and contribution of maintenance in sustainable manufacturing. In: Bakhtadze, N., Chernyshov, K., Dolgui, A., Lototsky, V. (eds.) Manufacturing Modelling, Management, and Control, Part 1, 7th IFAC Conference on Manufacturing Modelling, Management, and Control, vol. 7, pp. 1146–1151. International Federation of Automatic Control (2013)
8. Jasiulewicz-Kaczmarek, M., Piechowski, M.: Improvement of the process of information management in maintenance - a case study. Applied Mechanics and Materials, vol. 795. pp. 99–106. Trans Tech Publications, Switzerland (2015)
9. Kans, M.: IT practices within maintenance from a systems perspective: study of IT utilisation within firms in Sweden. J. Manuf. Technol. Manag. **24**(5), 768–791 (2013)
10. Kłosowski, G., Gola, A.: Risk-based estimation of manufacturing order costs with artificial intelligence. In: Ganzha, M., Maciaszek, L., Paprzycki, M., (eds.) Proceedings of the 2016 Federated Conference on Computer Science and Information Systems (FEDCSIS). pp. 729–732. IEEE, New York (2016)
11. Lopez-Campos, M., Cannella, S., Bruccoleri, M.: E-maintenance platform: a business process modelling approach. DYNA **81**(183), 31–39 (2014)
12. Saniuk, A., Caganova, D., Cambal, M.: Performance management in metalworking processes as a source of sustainable development. In: 22nd International Conference on Metallurgy and Materials - METAL 2013, pp. 2017–2022. TANGER, Czech Republic, Brno (2013)
13. Trapani, N., Macchi, M., Fumagalli, L.: Risk driven engineering of prognostics and health management systems in manufacturing. IFAC-PapersOnLine **48**(3), 995–1000 (2015)
14. Tretten, P., Ramin, K.: Enhancing the usability of maintenance data management systems. J. Qual. Maint. Eng. **20**, 290–303 (2014)
15. Zimmermann, H.-J.: An application-oriented view of modelling uncertainty. Eur. J. Oper. Res. **122**, 190–198 (2000)

Centralized and Distributed Structures of Intelligent Systems for Aided Design of Ship Automation

Ryszard Arendt(✉), Andrzej Kopczyński, and Przemysław Spychalski

Gdansk University of Technology, Gdansk, Poland
ryszard.arendt@pg.gda.pl

Abstract. A design process and accepted solutions made during this process, often base on non-formalized knowledge, obtained from designer (expert) intuition and practice. There are no formalized rules assuring the correctness of design solutions. The analysis of design process of ship automation, including ship power system, shows that this process can be supported by application of the artificial intelligence elements. The article presents two artificial intelligence technologies of computer-aided design regarding ship's power subsystems automation (expert system and the multi-agent environment). The paper discusses applications developed using both approaches and presents their performance comparison concerning requests processing efficiency.

Keywords: Computer-aided design · Expert system · Multi-agent system · Ship automation

1 Introduction

The process of computer-aided design, considering the intensive development of ships automation, puts on designers still growing demands of technical nature, the necessity of intensive knowledge complementation and frequent use of different expert opinions.

Growing competition, a drop in ship prices and expensive credits, force yards to considerably optimize the production cycle of ships. The most important factor of shipbuilding efficiency is designer's work, especially in early stages of design process, where content of a contract, the range and functions of ships, equipment prices and proposed suppliers of equipment are defined.

The whole ship design process consists of four basic stages: the preliminary design, the commission project, the technical project and the working project with strictly defined ranges of documentation approved by classification societies, shipyard owners, equipment suppliers and shipyard production departments.

Within the preliminary design (PD), the general concept and range of automation of the ship's power system are provided. The project includes elements, such as automation range, monitoring of the systems operation and proposed technical functionalities, realized by on-board computer or by local automation devices.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_28

After the shipping company has accepted the preliminary design and their requests have been taken into account, a commission project (CP) is drawn up (where the type of the main power plant is determined, together with the engine room systems and supportive devices).

Within the technical project (TP) the following items are prepared: the documentation required by the classification societies, a list of automation apparatus, block- and function diagrams of the engine room automation, the layout of monitoring stations, diagrams of power supply for automation devices, a list of computer input and output signals and diagrams of the engine room systems.

Once the comments of the classification society and the requirements of the shipping company have been accepted, the automation equipment suppliers are chosen and the supply contracts are signed, the working project (WP) is started, i.e. detailed diagrams of the automation systems for all the automated devices are prepared. When the ship has been built, final documentation (FD) is provided to the shipping company and the classification society.

Making decomposition of a design process we can distinguish some characteristic design stages related with the type of a project. We can highlight the following tasks:

- Text descriptions creation for all types of projects;
- Block and schematic diagrams creation, especially for TP, WP and FD;
- Application of data concerning component elements of ship power system for PD, CP and TP;
- Application of data regarding classification societies for all projects;
- Design calculations conduction, related with static properties for early stages of projects and dynamic characteristics for TP and WP;
- Design decision development concerning accepted technical solutions and applied component elements (connected with decisions and preferences of ship owner from CP), recommendations of a shipyard, costs, producers, reliability and many other factors.

Decisions made during design realization and accepted solutions often base on non-formalized knowledge, obtained from designer's (expert) intuition and practice. There are no formalized rules assuring standardization of design solution development [1, 2].

The analysis of design process of ship automation, including ship power system, shows that this process can be supported by an application of artificial intelligence elements [3–5]. Authors' attention has turned to an application of knowledge-based systems, as well as new technology solutions, such as multi-agent systems (MAS).

In the paper, a structure of an expert system, module of simulation calculations, and module supporting design decision making, developed in earlier grants are presented. The application of the multi-agent system for aided design of a ship automation is presented as well.

2 General Structure of the System

The previous research work enabled us to develop the general structure of a knowledge-based system for aided design of ship power systems automation (Fig. 1) [5–7].

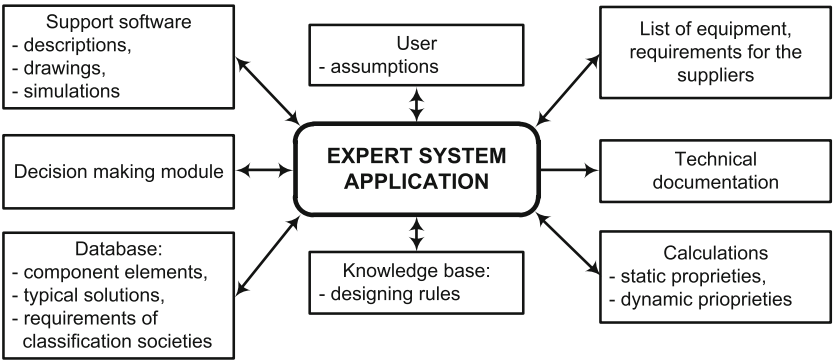


Fig. 1. General structure of knowledge-based system for aided design of ship power systems automation

An application of a shell expert system Exsys, executing many useful applications is assumed. The expert system works also as invocation environment for a number of external programs, which aid the design process [8]. Design assumptions are passed to the system by a user (expert) in dialogue form. Processing of submitted data is realized using rules from a knowledge base. The mentioned rules controls invoked programs, as well as input data transfers and data extractions from the received responses. The expert system uses the following auxiliary programs:

Database built using MS Access program, which contains descriptions of applied component elements used at design, typical technical solutions (subsystem structures) and chosen classification societies demands.

Auxiliary programs for technical descriptions and block diagrams creation, which include AutoCAD, CorelDRAW, Visio, Word and other programs.

Decision-making module, which is an independent knowledge-based system, which sends the chosen decision variants to the expert system. There are many decision models, which can be applied to computer aided design. Taking into account the flexibility of created knowledge-based system, the AHP (analytic hierarchy process) method is accepted. The structure of module that aid an automated choice of ship's control subsystem elements is presented in Fig. 2.

The performance of this module was tested on an exemplary decision-making problem concerning a selection of temperature sensors in fuel transport system to Diesel engines of a ship's main propulsion [7]. The most important procedures realized by the modules include:

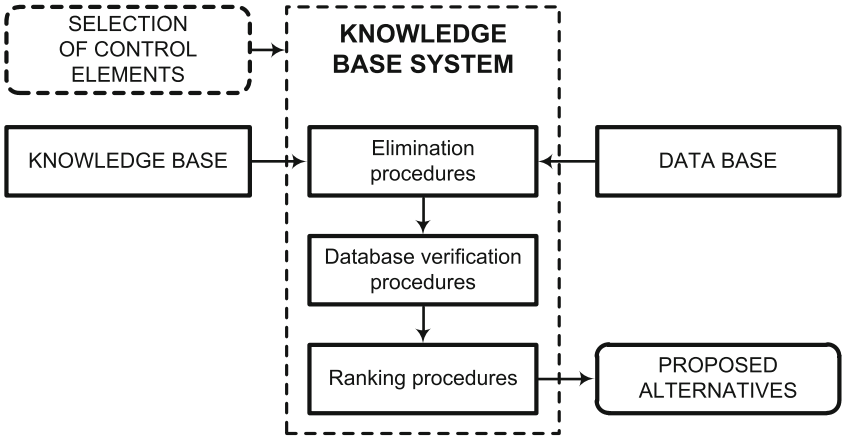


Fig. 2. Structure of the module in knowledge-based system for aided design decision making

- Searching the database for an objects that meet given criteria;
- Creating the pairwise comparison matrices for expert knowledge acquisition (for the objects that have been found);
- Object correctness verification and writing data to the hard disk drive;
- Utilizing the knowledge accumulated by the user in a form of calculating ranking vectors and their aggregation according to accepted attributes;
- Generating reports.

Simulation module, that includes: simulation program, mathematical models library, structure models library and parameter models base (Fig. 3). The general type of each element is represented by the mathematical model (a given structure of the model and mathematical operators). The detailed type could be achieved by introducing properly identified model parameters. The choice of parameters is possible by execution of identification process using genetic algorithms. Structure models library includes mathematical models of chosen subsystems (the library is not included in Fig. 3).

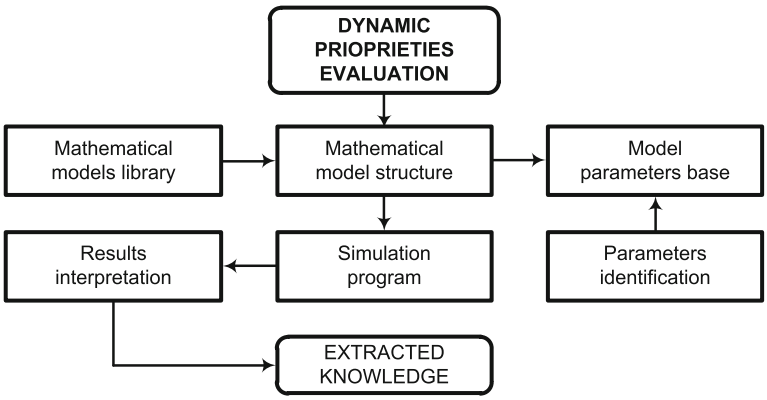


Fig. 3. Simulation module structure for dynamic evaluation of designed system

The attempts of automatic generation of model structures from its description using the expert system were made [4]. Results of simulation time responses of the investigated system should be evaluated according to the calculations of designed subsystem. The necessary knowledge should be extracted from generated plots and used by the expert system.

The described assumptions were used do built a practical implementation of an expert system for aided design of ship thruster subsystem (electrical section). The next chapter presents the application details regarding the developed system.

3 Application of Expert System for Design of Ship Thruster Subsystem

The previous experience with expert system building, results in a hybrid expert system creation, which supports the choice of elements of ship's thruster subsystem. The design process of ships and their construction have special requirements, among others, in newly built ships a designer should not introduce too many innovations. Ship owners require high reliability, predictability of the ship endurance and the assessment of long-term operating costs.

The proposed aided design system of thruster subsystems is mainly designed for PD. It is useful for the design of newly built ships, as well as for renovation of existing units. The system can be also used in the later stages of design CP, TP, WP and FD.

The aided design of the thruster subsystem is based on the structure shown in Fig. 1. Similarly to the classic expert system, the proposed system uses a knowledge base, database, chaining forward and backwards. Moreover, the system uses results of simulations carried out in real time (online). These results allow for the assessment of thruster structures, selected by the expert system, as well as the generation of design solutions. The built system includes elements such as:

- Library of ship thruster structures;
- Library of mathematical models of elements of ship thruster structures [8];
- Identification module of mathematical model parameters using the genetic algorithm;
- Database of catalogue data and the database of mathematical model parameters of thrusters component elements;
- Database with knowledge about ship thrusters;
- Module of automatic generation of the simulation models in workspace;
- Module of automatic extraction of knowledge from simulation results;
- Reports generator (the report includes an analysis of the tested subsystem of ship thruster).

The expert system is built from the following tools: Exsys Developer, MS Access, Matlab/Simulink, Genetic and Evolutionary Algorithm Toolbox.

The expert system operation starts from the conversation mode, to collect information from the user (designer expert). Then, the system performs automatically a series of tasks:

- Select ship thruster structure;
- Select elements of ship thruster structure;
- Demand simulation calculations of the ship thruster structure from external application;
- Assign the knowledge extraction from simulation results to external application;
- Generate the final report.

External application (Matlab/Simulink) connects mathematical models of particular components of the thruster structure in a form of simulation model. Then the system assigns the parameters to mathematical models and performs the simulation. Finally, the simulation results are returning to the expert system.

The expert system invokes the external application (Matlab) to perform evaluations of the simulations results. During the evaluation, basing on characteristics, the application extracts the necessary knowledge to make a decision, whether to accept or reject the tested structure.

The results from the application return to the expert system. The rules, used in evaluating the simulated structure, were developed using expert knowledge base and rules of classification societies (i.e. PRS - polish register of shipping). Finally, the expert system generates a report from accomplished activities.

Thruster parameters selected by the user (during the conversation), the results of calculations made by the expert system, elements of selected structure, data sheets selected elements of the structure, knowledge obtained from simulation of the structure, information if requirements set by selected classification society are met, as well as final conclusions, are presented in the report.

When the structure does not meet the requirements, the application of an expert system suggests what further steps can be taken to improve the properties of the structure. In the future a fully automated selection of the thruster structure and further research to reach the expected or the best solution is planned.

Example parts of the report generated by the expert system is shown in Figs. 4 and 5. The Fig. 4 shows: a fragment of a report containing the structure description (Fig. 4, p. 1.5), selected component elements for chosen structure (Fig. 4, p. 1.6), and catalogue data of component elements (Fig. 4, p. 1.7). The Fig. 5 (p. 1.10) shows some conclusions generated by the expert system.

When the created structure does not meet requirements set by given classification society and the expert system suggests how to improve the properties of tested structure, then the user of an expert system is able to get a direct access to auto-saved *.mdl file with the structure model for debug purposes. This step is optional (not included in a typical workflow of an expert system), however it is required to analyze the raw data obtained from tests conducted in simulation module. Basing on the system report conclusions and model characteristics analysis, the user can improve the created structure and rerun the simulation tests with new parameters (in order to verify correctness of this changes).

1.5 The chosen structure contains following electrical equipment:
- The power of each generator installed on the ship 2800 *kW*,
- The number of generators installed on the ship 3 *el.*,
- The feeder cable length of the induction motor 15 *m,*
- Cross section of the feeding cable the induction motor 25 mm^2,
- Electrical conductivity of the feeder cable 58.6 $S*m/mm^2$.

1.6 The structure with a direct start was selected. Elements of the chosen structure are:
- The Diesel engine,
- A synchronous generator,
- An induction motor,
- A propeller shaft,
- A controllable pitch propeller,

1.7 Catalog data of Selected components from the database are:

1.7.1 Induction motor
- The number in the database = 1
- Type of the electric motor = *AMA 400L6D VAMH*
- Engine power in [*kW*] = 800
- Rated voltage [*V*] = 440
- Full load current *In* [*A*] = 1331

Fig. 4. Part of the report generated by expert system - parameters, structure, components

1.10 Final conclusions of the expert system report:
- The number of generators installed on the ship is insufficient. According to the regulations of classification societies should be installed at least one generator backup in case of failure.
- The structure does not meet the requirements for voltage drop *Ug* in a transitional state for the chosen classification society – when the generator is load by starting induction motor.
  A momentary drop voltage of generator *Ug* when the induction motor is turning on is 35.8913%.
  Allowable momentary drop in voltage generator *Ug* under the provisions of the classification society is 20%.
  Settings of voltage controller should be re-selected.

Fig. 5. Part of the report generated by expert system - final conclusions

4 Multi-agent Based Approach for Design of Ship Thruster Subsystem

The implemented expert system was primarily designed as a standalone application. This approach contains certain disadvantages and limitations. Among them can be distinguished the centralization of all algorithms within a single hardware unit that led to the excessive workload during simulation tests (caused by Exsys and Matlab background activities). The single CPU is forced to process every request by itself, thus the expert system must be installed on a high-performance workstations [3].

Another aspect is a concurrent access to the resources. Only one user could take advantage of the system at a time (the system is not capable to provide another interface for a subsequent user, therefore simultaneous multi-user operations are not available).

Furthermore, the scheduled updates of database or knowledge base prevent from the use of an expert system for a certain time period. Implemented system is characterized by sequential conduction of design processes. Each subsequent process can be carried out, only when its predecessor was completed (it is not possible to conduct several design processes in parallel) [3].

Mentioned limitations could be overcome by application of the multi-agent based approach. Multi-agent systems (MAS) evolved from the research related with distributed artificial intelligence. This modus operandi assumes that the computational system consists of autonomous, intelligent instances (software agents). Agents are capable to communicate with each other in order to achieve some designated, collective goals. Mutual interactions and the possibility to execute unique algorithms are an indispensable feature of software agents. Therefore, this solution (MAS) is frequently applied for resolving complex problems with dispersed nature [9, 10]. One of those issues is the aided design of control systems.

Certain attempts to apply multi-agent architecture in the systems for computer-aided design (CAD) of a ship automation have been made [11, 12]. However, the aim of those researches is rather focused on enabling a cooperative design in terms of diverse CAD environment integration within a particular system or to manage the knowledge sharing between geographically distributed design teams. In this article authors introduced an application of the multi-agent system in computer-aided design, regarding the parallelization of design algorithms. In the described approach the specialized software agents were implemented as follows: interface agent, master agent, simulation agent and evaluation agent. The new instance of every agent's type might be invoked in order to satisfy the request processing demand.

The multi-agent system that utilizes the described assumptions is presented in the Fig. 6 (hardware layer perspective). Interface Agent gathers and initially validates the data from user designer. Master agent delegates tasks to computational agents and additionally enables the machine learning algorithms. simulation agent carries out dynamic calculations for the designed control structure. evaluation agent assesses the suitability of generated solution and provides a report to the system user. The main advantage of this approach is the scalability of an entire multi-agent system and the ability to conduct several design processes in parallel. Particular design processes are divided and distributed between separate computers. Implemented queueing mechanism allows to allocate the given design step on appropriate hardware unit taking into account its capabilities and current workload.

This solution led to reduction of the time of solving design tasks, compared with the centralized approach. A comparison of request processing efficiency (time required to develop the solutions for three the same design processes), regarding the centralized approach (expert system) and distributed approach (multi-agent system) is shown in the Fig. 7.

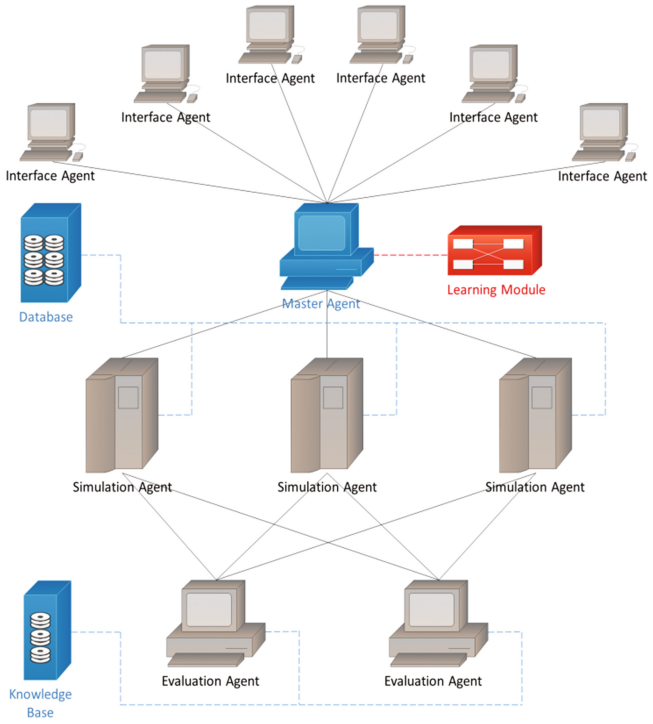


Fig. 6. Multi-agent system for aided design of selected control systems (hardware layer)

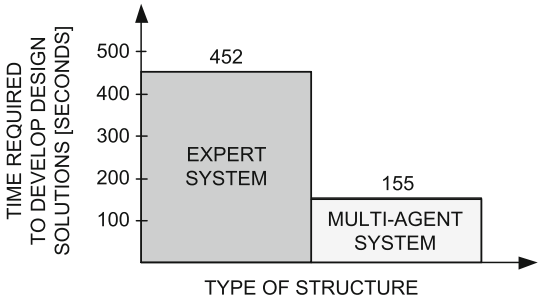


Fig. 7. Performance analysis on request processing efficiency of expert system and multi-agent environment (lower is better)

5 Summary and Conclusions

The process of designing system automation is complex and difficult to formalize. Furthermore, it requires processing of large data amounts. In designing ship automation systems there is no fully systematized knowledge about the designing process because often the only information source are the people involved in solving problems in this

area. Apart from that, there is also the output of many projects made on ships. This knowledge should be a subject to standardization. It should be systematized and documented, which is partially done in the expert system (in the knowledge base and in the form of reports). Dedicated application of an expert system for aided design of ship system automation met many difficulties. Detailed database of component elements is necessary. Mathematical models of component elements with identified parameters should be used. Furthermore, it is very complicated to create a dedicated knowledge base. Despite the mentioned problems the achieved results are satisfactory and the support of an expert system is considerable. The centralized architecture of the expert system contains certain disadvantages and limitations that might be overcome by application of the multi-agent based approach. The main advantage of this solution is the scalability and ability to conduct several design processes in parallel. Multi-agent approach is more efficient compared with centralized expert system and suitable for the CAD domain. After implementation of multi-agent architecture the time required to develop solutions for exemplary design processes was significantly reduced.

References

1. Park, J.H., Storch, R.L.: Overview of ship-design expert systems. Expert Syst. **19**, 136–141 (2002)
2. Yang, H.Z., Chen, J.F., Ma, N., Wang, D.Y.: Implementation of knowledge-based engineering methodology in ship structural design. Comput. Aided Des. **44**, 196–202 (2012)
3. Arendt, R., Spychalski, P.: An application of multi-agent system for ships power systems design. In: Proceedings of 20th International Scientific Conference Transport Means, pp. 380–384 (2016)
4. Arendt, R.: The application of an expert system for simulation investigations in the aided design of ship power systems automation. Expert Syst. Appl. **27**, 493–499 (2004)
5. Arendt, R., van Uden, E.: A decision-making module for aiding ship system automation design: a knowledge-based approach. Expert Syst. Appl. **38**, 410–416 (2011)
6. Arendt, R.: Wykorzystanie modeli systemu elektroenergetycznego statku. Przegląd Elektrotechniczny **87**, 206–211 (2011)
7. Arendt, R.: Modelowanie wpływu zasilania na pracę odbiorników systemu elektroenergetycznego statku. Przegląd Elektrotechniczny **88**, 141–146 (2012)
8. Kopczyński, A.: Mathematical models in design process of ship bow thrusters. In: Proceedings of 20th International Scientific Conference Transport Means, pp. 328–331 (2016)
9. Russel, S.J., Norving, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall, Englewood Cliffs (2010)
10. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, 1st edn. The MIT Press, Cambridge (1999)
11. Xiangzhong, F.: Ship collaborative design based on multi-agent and ontology. In: Proceedings of 5th International Conference on Cooperative Design, Visualization and Engineering, pp. 249–252 (2008)
12. Chao, K.M., Norman, P., Anane, R., James, A.: An agent-based approach to engineering design. Comput. Ind. **48**, 17–27 (2002)

Simulation-Based Analysis of Wind Farms' Economic Viability

Joanna Wyrobek[1(✉)], Jarosław Wąs[2], and Marek Zachara[2]

[1] Cracow University of Economics, 27 Rakowicka Av., 31-510 Krakow, Poland
wyrobekj@uek.krakow.pl
[2] AGH University of Science and Technology, 30 Mickiewicza Av., 30-059 Krakow, Poland
{jarek,mzachara}@agh.edu.pl

Abstract. In the following paper, we propose an utilization of a simulation - based approach for a wind farm viability analysis. The simulation is created in Erlang language, and it takes into account a set of parameters selected on the basis of a bibliographical analysis. A number of simulations were performed with the discussed model and selected results have been presented in this paper. It was found out that the economic viability of wind farms is strongly correlated with a number of external conditions, including the environmental factors, organizational costs and governmental support.

Keywords: Wind farms · Economic viability · Break-even

1 Introduction

Our population is eventually going to exhaust the fossil fuels that are economically reasonable to extract. Since the global demand for energy is not likely to decrease significantly, the world needs an increasing amount of energy from renewable sources. Of these sources, wind farms are one of the primary contributors, with over 400 GW in power capacity available in 2015, second only to hydropower capacity [1].

Designing and building of a new wind power plant poses however certain challenges related to both technical and economical issues that are faced by the new installation. The new power plant has to comply with certain technical and legal requirements but its ultimate success or failure will depend on a number of factors, some of which are listed below:

- wind characteristic at the plant site, its predictability, average and maximum strength, etc.;
- power generated by the turbines at specific wind speeds;
- impact of one turbine on the output of others nearby;
- turbine failure rates, and types of damages;
- management and logistics of the maintenance teams.

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_29

Some of these factors are outside investor's control (e.g. the wind patterns), but can be estimated and factored in. Others are fully controllable, like the turbine placement or the management of the maintenance teams. In most of the cases, a new potential build of a wind power plant can modeled to evaluate its feasibility under assumed conditions.

This article will focus on economic modeling of a new wind farm, to determine it predicted profitability under defined conditions and identify the impact of certain design parameters. Such profitability requires to model the following aspects of the enterprise:

- wind conditions;
- turbine placement;
- turbine type(s);
- maintenance staff and procedures;
- turbine controllers and connection to the power grid.

The models of a wind farm usually consist of at least four elements: wind field, turbines, the controller and the power grid. Their relation is presented in Fig. 1. In this figure, *stat* refers to the on/off state of the turbine, C_t is the turbine's torque, P_e is the generated power $fatigue$ is the estimated wear-off, V_{meas} is the wind speed measured at the turbine while V_h is the wind speed at the node's level. P_{avail} is the available power, P_{dem} is the demanded power and P_{ref} is the reference power of the turbines.

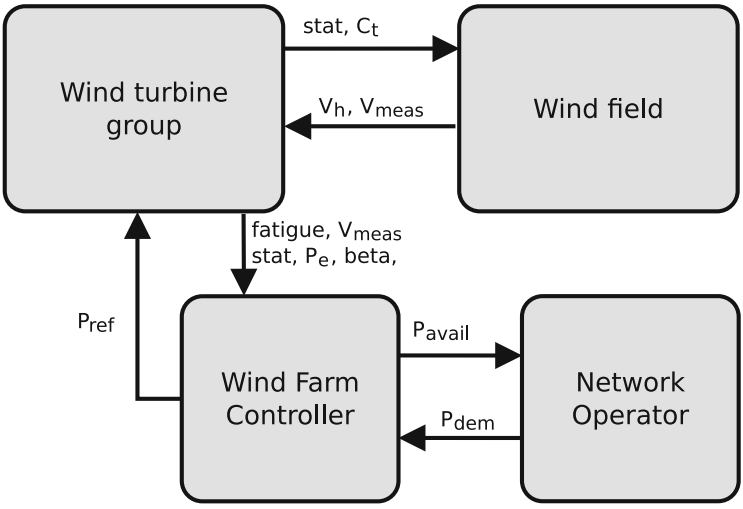


Fig. 1. Sample model of a wind farm, from [8].

To model the behavior of the turbine and the generated output, a general electromechanical model can be used that provides the turbine's power output as a function of various inputs (type, wear-off, rotor speed, etc.).

2 Wind Power Generation

In order to calculate the efficiency of a wind turbine, a number of steps need to be performed. First, one shall estimate the wind power available at the specific site. This is usually done by measuring the wind speed over a course of a year at specific intervals (e.g. every 10 min). The results can then be aggregated and interpolated using e.g. Weibull's distribution [6]:

$$P(V) = \frac{k}{A}\left(\frac{V}{A}\right)^{k-1} e^{-(\frac{V}{A})^k} \tag{1}$$

where:
P – is the power of the wind the turbine can utilize
V – is the wind speed
where:
A,k – model parameters
V – is the wind speed

Then, the mechanical energy the turbine can utilize may be calculated according to the below equations, which are taken from [6]:

$$P = \frac{1}{2}\rho S V^3 C \tag{2}$$

where:
P – is the power of the wind the turbine can utilize
ρ – is the density of the air
S – is the effective area of the rotor
V – is the wind speed

the C parameter is the Betz coefficient defined as follows:

$$C = \frac{1}{2}\left(1 + \frac{V_0}{V_1}\right)\left(1 - \frac{{V_0}^2}{{V_1}^2}\right) \tag{3}$$

where:
V_0 – is the air speed approaching the turbine
V_1 – is the air speed behind the turbine

The coefficient reaches maximum of 0.593 when V_0 is three times higher than V_1, however in real conditions it rarely achieves more than 0.4.

During the next step the torque available to the power generator is calculated based on the electromechanical construction of the specific turbine. One of the models for such calculations was proposed in [3] and is based on the following elements:

- Joint innertion of the rotor and its shaft I_r.
- An absorber with a viscosity of B_r representing the bearings before the gear box.

- Weightless spring with elasticity K_d and viscosity of B_d. Deformation of the spring in radians measured as θ mirrors the deformation of the shaft.
- A gear box with N_g ratio.
- An absorber with a viscosity of B_g representing the bearings after the gear box.
- Joint innertion of the gear box, high-speed shaft and the generator I_g.

The model is described by the following differential equations:

$$I_r \omega'_r = T_a - K_d \theta - B_d \left(\omega_r - \frac{\omega_g}{N_g} \right) - B_r \omega_r \tag{4}$$

$$I_g \omega'_g = T_g - \frac{K_d}{N_g} \theta - \frac{B_d}{N_g} \left(\omega_r - \frac{\omega_g}{N_g} \right) - B_g \omega_g \tag{5}$$

$$\theta' = \omega_r - \frac{\omega_g}{N_g} \tag{6}$$

where the aerodynamical torque T_a and the generator's torque T_g are considered inputs and the results are ω_r and ω_g. The generator is trying to deliver an electrical power P_e tied to the reference power signal P_0. The power depends on the torque available at the high-speed shaft and in a simplistic scenario can be represented as:

$$P_e = \omega_g T_g \tag{7}$$

Real-world generators cannot change their torque immediately. The delay is modeled as a first-order relation between the expected and actual generator's torque [3]:

$$T'_g = \frac{1}{\tau_g} (T_{g,ref} - T_g) \tag{8}$$

where τ_g is a time constant. Since the requested torque can be derived from:

$$T_{g,ref} = \frac{P_0}{-\omega_g} \tag{9}$$

which results in:

$$T'_g = \frac{1}{\tau_g} \left(\frac{P_0}{\omega_g} - T_g \right) \tag{10}$$

The mechanism that changes the angle of the blades can be modeled as follows [3]:

$$\beta = \frac{1}{\tau_b} (\beta_{ref} - \beta) \tag{11}$$

In case of a simple turbine's controller, the turbine load is switched based on the measured wind speed. This controllers calculates and sets the β_{ref} parameter in order to not exceed the maximum rotation, but at the same time maintain the maximum available efficiency for the available wind speed.

Additionally, when modeling a turbine and its power generation the wear-off of the turbine's part and the related decrease of its efficiency must be taken into the account. One of the simpler models is based on the Wöhler's curve [4,5]:

$$D_s = k^{-1} \int_{-\inf}^{\inf} \omega \Phi_x(\omega)^{\frac{m}{2}} d\omega \tag{12}$$

where:
$\Phi_x(\omega)^{\frac{m}{2}}$ – is the amplitude of the frequency ω
k – is the parameter denoting specific material wear-off

There are number of other models and modeling techniques involved, including modeling of the wind field through the wind farm area and modeling of the interface to the power grid and its operator. These are, however, outside the scope of this article.

3 The Proposed Model

In order to perform the simulation-based estimation of a wind farm profitability, the model was designed which incorporated a number of assumptions.

The farm's turbines are placed evenly at the site, with similar distances to their neighbors. The wind is modeled with its speed changing both periodically (e.g. during the day) with a random component. The model also allows for different wind strength at different places within the farm area.

The energy produced is calculated using the equations described in the previous section.

Additionally, there is a probability of malfunction that may happen to any of the turbines. There is also a required set of maintenance teams which are dispatched to take care of the reported malfunctions. A malfunction can be "major" and "minor" and depending on this factor it will take adequate time for the maintenance team to fix the malfunction. As it turned out, the management of the malfunctions and the maintenance teams is on of the most important issues contributing to the wind farm's profitability (or lack thereof).

3.1 Input and Output of the Proposed Model

The aim of the simulation based on created the model is to identify the output parameters for predefined initial conditions. In order to simulate operational and economic aspects of a wind farm we have identified some basic parameters. The input includes following primary parameters:

simulation period - duration time of a simulation;
turbines count - number of wind turbines [days];
territorial span - area, where turbines are allocated [km × km];
vehicles count - number of technical support cars [units];
minor repair - time amount for minor repair of a turbine [seconds/repair];

major repair time - time amount for major repair of a turbine [seconds/repair];
examination time - time to identify the scale of turbine failure (minor/major) [seconds];
average wind speed - average (base) wind speed [m/s];
wind variations minor - difference between minimum and maximum winds in a particular day [m/s];
wind variations major - the difference between wind speed in a windy day and a windless (low wind) day [m/s];
annual turbine income - annual income of 1 turbine for medium wind, assuming no failure [USD/turbine];
annual team cost - annual maintenance costs of the technical support team [USD/team/year];
wind sampling period - sampling time of calculated wind velocity [seconds];
turbine breakdown sampling period - turbine's breakdown sampling time [seconds];
number of repair teams - number of repair teams available [decimal number];
probability of a minor breakdown - probability of a minor turbine breakdown per year [percentage value/turbine];
cost of a minor repair - cost of a minor turbine repair [USD/repair];
probability of a major breakdown - probability of a major turbine breakdown per year [percentage value/turbine];
annual costs of turbine's major repair - cost of a major turbine repair [USD/repair];
examination time for a repair team - examination time necessary for a repair team to determine whether they deal with a minor or major breakdown [seconds];
variation factor - scale of wind velocity variation between different turbines [decimal value];
energy hour cost - electricity selling price to the grid [USD/kWh].

Outcomes of the simulation include:

- Revenue from individual wind turbines over time [USD/hour];
- Costs of individual wind turbines over time [USD/hour];
- The energy that turbines generate over time [kWh];
- Total wind farm economic output over time [USD/hour];
- Locations of technical support teams in time units [Cartesian Plane coordinates].

3.2 Main Assumptions of the Model

The farm model assumes that wind turbines are distributed in evenly spaced areas. Turbines break down with a certain probability, described by a pseudo-random number, and if its value is below a certain threshold, then the it is assumed to have failed. The probability of a minor and a major failure is determined separately. Each turbine is a separate process that has its own failures.

When a failure occurs, the process called Dispatcher sends a support team to a site. After examination time support team informs the dispatcher whether a minor or a major failure occurred and what is the expected repair time.

Wind is a separate process that is modeled based on a sine-line, because usually the strongest wind blows in the morning and evening. Input variables are minimum wind power and amplitude. In addition, a random amplitude coefficient was introduced in the amplitude range, which, if negative wind force, was reduced to zero (no wind).

If the variable "windiness" is set to 0, all turbines have the same wind force. If it is 0.5 then the difference is 50%, if the value is 2 the difference is 200%, etc.

The energy generated by the turbine is calculated using the formulas presented in the theoretical part of the paper. The power generated per wind unit (in $\frac{\mathrm{m}}{\mathrm{s}}$) is calculated as follows:

$$P = \frac{1}{2} * 1.225 \frac{\mathrm{kg}}{\mathrm{m}^3} * \pi * (1.67\,\mathrm{m})^2 * 0.5 * (\frac{116}{5.24} - 5) * e^{(-21/5.24)} * (1 \frac{\mathrm{m}}{\mathrm{s}})^3 \quad (13)$$

Which resolves to:

$$P = 0.838 \frac{\mathrm{kg} * \mathrm{m}^2}{\mathrm{s}^3} = 0.838\,\mathrm{W} = 0.838e^{-3}\,\mathrm{kW} \quad (14)$$

3.3 Activity Diagrams

Each turbine is a separate process, for which failures are generated with a random process including minor and major ones. In our simulation support (repair) teams are under the control of a process called dispatcher. Dispatcher sends inquiries to the turbine and repair process and gets answers about their condition. The actors' answers are defined as follows: turbines correspond to where they are, how much they have earned so far and when they break and when they are repaired, while support teams will localize where they are and when they will finish repairing the turbine. Two following components should be also mentioned: Chronicler - collects information and calculates how much money a wind farm earns, while time source sends the message that contains and all the actors respond when something happens to them - the turbine breaks down or the repair ends when teams are free or where they are going. The source of the time determines which event (failure, free team) will be the first and "jumps" until the time it sends another "signal" to the actors - what to do with them.

3.4 Additional Information

Three-wing turbines and typical energy efficiency were adopted, with no additional mechanism for adjusting the speed of rotation of the wings as a function of wind. The applied approach is the implementation of the theoretical formulas for converting the amount of energy generated by the turbine depend on the wind force. A constant conversion of wind energy into revenues has been adopted.

Erlang was chosen for the simulation because the entire model is based on, almost continuous, exchange of messages between turbines, repair teams and dispatchers. Also, the time source sends time pulses to all participants in the simulation. Erlang is a language created for working on many processes, so it made implementation easier.

We have applied different processes in Erlang including: actors, breakdowns, chronicler, dispatcher, events, timesources, wind and turbines characteristics etc.

4 Simulation Results

We have tested several scenarios regarding: different numbers of wind turbines, numbers of technical support teams and wind conditions.

Due to limited room we include in the paper only sample results of our simulations. In this sample simulation it was assumed that the wind farm include 10 turbines and one technical support (repair) team. Results are presented in Polish zlotys, simulation results were converted into this currency (prices of green certificates were originally given in PLN). The results of the simulation showed that the further increase in the number of turbines (as compared to previous scenario) improved the financial performance of the farm and the number of failures was still low enough to support one technical support team. A slight increase (from 15 to 16 $\frac{\mathrm{m}}{\mathrm{s}}$) of average wind speed also improved farm performance and energy output.

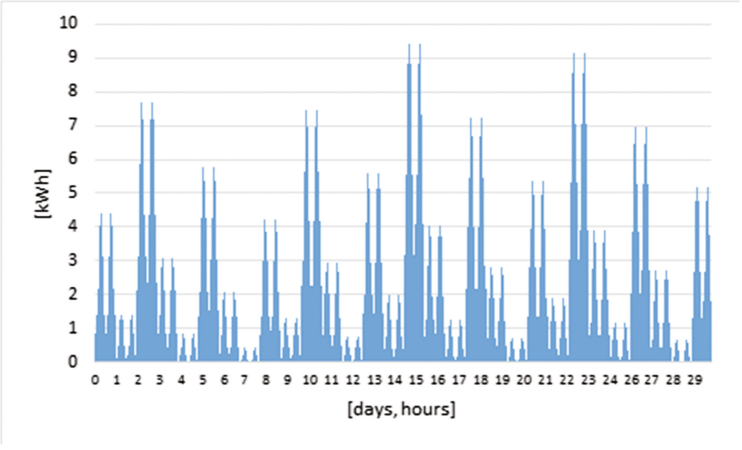


Fig. 2. Graph of generated energy in kWh per hour by turbine 1. Graphs for consecutive turbines are similar.

As you can see in the Fig. 2 amount of generated energy is strongly correlated with wind velocity and the final financial result (Fig. 3).

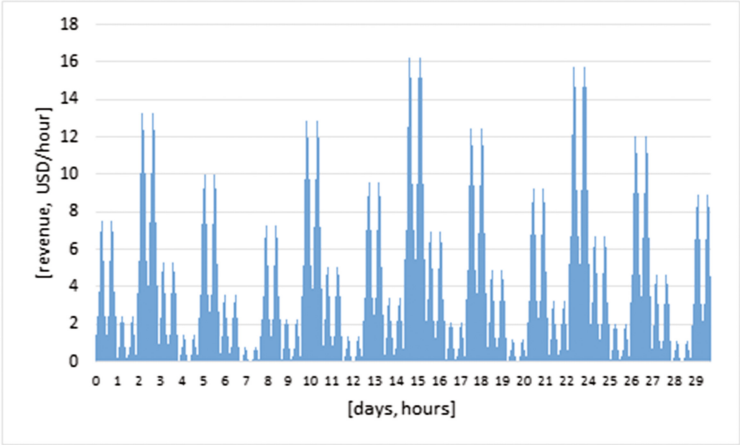


Fig. 3. Financial result of the farm.

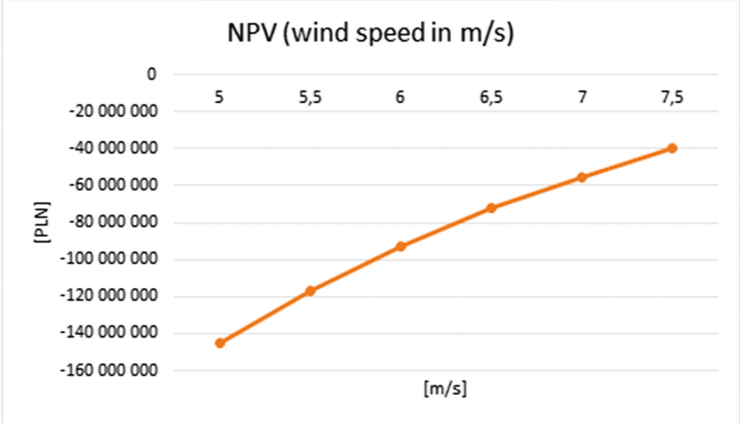


Fig. 4. Profitability of a wind farm for an average wind speed in the area.

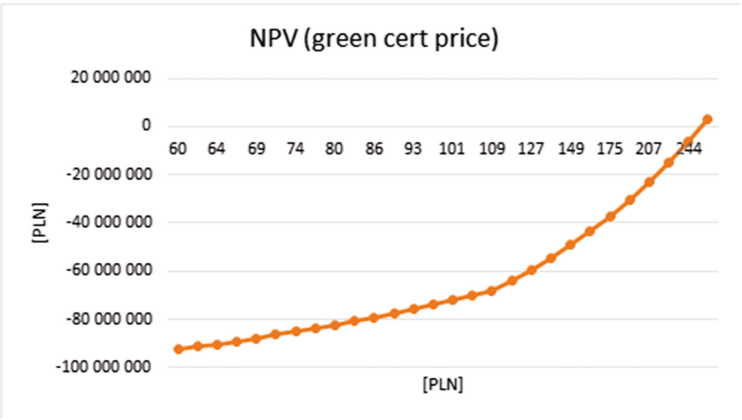


Fig. 5. Change in the profitability as a function of the projection period average "green certificate" price.

5 Conclusions

The model of the wind farm was used to assess the potential profitability of the farm, based on the number of economic parameters, including the price of the energy and the price of the "green certificates" which are used to support the development of renewable energy sources. These parameters were retrieved from published materials, namely [2,7,9].

Of these parameters, the price of the "green certificates" is the most volatile, with the price per MWh in Poland changing from 240 PLN (approx. 55 EUR) to 60 PLN (approx. 14 EUR) within one year. Another parameter that varied greatly between samples were maintenance costs which ranged from 2.5 to 14 EUR per MWh. Other parameters, like management & supervision, insurance, etc. also varied, but do not have significant impact on the profitability of the farm, as they are, on average, an order of magnitude lower.

The following figures present the impact of the most important factors on the profitability of a wind farm. As can be seen in Fig. 4, the wind speed has an important impact, as the power generated grows with the average wind speed. The growth is non-linear because of turbine wear and the need to switch off turbines at certain speed to minimize the risk of damage. In the considered scenario, half of the turbines were switched off when the wind reached speed of 23 m/s and all of them were switched off at 27 m/s.

The price of the "green certificates" has even more profound impact. Considering the change in price of these certificates that happened between 2016 and 2017, it can be clearly seen in Fig. 5 that such change implies a difference between profitability of the enterprise and a major loss.

The analysis and preliminary results show that economic viability of wind farms is highly susceptible to a number of external conditions, including the environmental factors, organizational costs and government support. Only under

correct set of circumstances they can be profitable and can quickly start generating losses. In some countries (like Poland), another factor that has a serious impact on the whole enterprise is the current exchange rate, because most of the equipment is imported and paid for in EUR or USD, while the earnings are in the local currency.

6 Future Research

Presented model, despite its internal intricacy, requires further developments. Authors plan to improve the function which describes wind behaviour and introduce to the model a simple wind field which would allow to model wind behaviour after it went through a turbine. Adjustments are also planned for more sophisticated regulators.

References

1. Renewables 2016 global status report. http://www.ren21.net/wp-content/uploads/2016/06/GSR_2016_Full_Report.pdf
2. Gnatowska, R., Wąs, A.: Analiza opłacalności inwestycji w produkcję energii ze źródeł odnawialnych na przykładzie farmy wiatrowej przy założeniu wsparcia rządu dla "zielonej energii". Inżynieria i Ochrona Środowiska **18**(1), 23–33 (2015)
3. Hammerum, K., Brath, P., Poulsen, N.K.: A fatigue approach to wind turbine control. J. Phys. Conf. Ser. **75**, 012081 (2007). IOP Publishing
4. Hansen, A.D., Sørensen, P., Iov, F., Blaabjerg, F.: Power control of a wind farm with active stall wind turbines and ac grid connection. In: European Wind Energy Conference, EWEC 2006, Athens (2006)
5. Hansen, M.O.: Aerodynamics of Wind Turbines. Routledge, New York (2015)
6. Jąderko, A., Kowalewski, M.K.: Wyznaczanie parametrów wiatru w energetyce odnawialnej. Przegląd Elektrotechniczny **91**(1), 148–151 (2015)
7. Mielcarek, J.: Analiza projektu farmy wiatrowej za pomocą rachunku kosztów docelowych. Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu (2014)
8. Soltani, M., Knudsen, T., Bak, T.: Modeling and simulation of offshore wind farms for farm level control. In: European Offshore Wind Conference and Exhibition (EOW) (2009)
9. Suska-Szczerbicka, M., Weiss, E.: Ocena opłacalności przedsięwzięcia inwestycyjnego produkcji energii elektrycznej farmy wiatrowej. Rynek Energii **1**, 104–111 (2013)

A Hybrid Genetic Algorithm for Hardware–Software Synthesis of Heterogeneous Parallel Embedded Systems

Mieczysław Drabowski(✉) and Kazimierz Kiełkowicz

Department of Computing Sciences, Cracow University of Technology, Kraków, Poland
{drabowski, kkielkowicz}@pk.edu.pl

Abstract. The paper includes a proposal of a new algorithm for hardware–software synthesis of heterogeneous parallel embedded systems. Optimal scheduling of tasks, optimal partition of resources and allocation tasks and resources are fundamental problems in this algorithm. In the former synthesis methods, software and hardware parts have been developed separately and then connected in the process of so-called concurrent synthesis. The objective of this research is to present the concept of coherent approach to the problem of system synthesis, i.e. a combined solution to task scheduling and resource partition problems. The approach is new and original and allowing synergic design of hardware and also software controlling the performance of a computer system. This is an approach which we call a coherent parallel synthesis.

This paper shows the algorithm, based on genetic method assisted with simulated annealing strategy and shows the results of selected representative computational experiments into different instances of system synthesis problems which prove the correctness of the coherent synthesis concept and indicate methods solving these problems.

Keywords: Synthesis · Synergic · Scheduling · Allocation · Partition · Optimization · Genetic · Simulated annealing · Boltzmann tournament

1 Introduction: Coherent Synthesis of Computer System

The aim of computer aided design of complex systems is to find an optimum solution consistent with the requirements and constraints enforced by the given specification of the system. The following criteria of optimality are considered: costs of system implementation, its operating speed and power consumption. The identification and partitioning of resources between various implementation techniques is the basic matter of automatic design. Such partitioning is significant, because every complex system must be realized as result of hardware implementation for its certain tasks. Additionally scheduling problems are one of the most significant issues occurring in design of operating procedures responsible for controlling the allocations of tasks and resources in complex systems. In new design methods – in the coherent parallel synthesis – which model and approach were presented in [1], the software and hardware components are developed together and parallel and then coherent connected together, which

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_30

of the final solution decreased the costs and increased the speed. The resources distribution is to specify, what hardware and software are in system and how to allocate theirs to specific tasks, before designing execution details.

Due to the fact that synthesis problems and their optimizations in decision version are NP-complete we apply meta-heuristic approach, in this paper genetic with simulated annealing.

2 Evolutionary Algorithm of Coherent Synthesis of Parallel Embedded Computer Systems

In order to eliminate solution convergence in genetic algorithms, we use data structures which ensure locality preservation of features occurring in chromosomes and represented by a value vector. Locality is interpreted as the inverse of the distance between vectors in an n-dimension hyper-sphere [2]. Then, crossing and mutation operators are data exchange operations not between one-dimensional vectors but between fragments of hyper-spheres. Thanks to such an approach, small changes in a chromosome correspond to small changes in the solution defined by the chromosome. The presented solution features two hyper-spheres: task hyper-sphere and resource hyper-sphere. The solutions sharing the same allocations form the so-called clusters. The introduction of solution clusters separates solutions with different allocations from one another. Such solutions evolve separately, which protects the crossing operation from generating defective solutions. There are no situations in which a task is being allocated to a non-allocated resource. Solution clusters define the structures of the system under construction (in the form of resources for task allocation). Solutions are the mapping of tasks allocated to resources and task scheduling. During evolution, two types of genetic operations (crossing and mutation) take place on two different levels (clusters and solutions). A population is created whose parameters are: the number of clusters, the number of solutions in the clusters, the task graph and resource library. For the synthesis purposes, the following criteria and values are defined: optimization criteria and algorithm iteration annealing criterion if the solution improvement has not taken place, maximum number of generations of evolving within clusters solutions, as well as the limitations—possibly the biggest number of resources, their overall cost, total time for the realization of all tasks, power consumption of the designed system and, optionally, the size of the list of the best and non-dominated individuals [3].

2.1 Selected Data Structure

"Found solutions" structure.
The structure contains the information about the outcome structure and functionality of the system architecture; thus

- The task and resource allocation.
- Optimized criteria.
- Solution ranking (the number of solutions in population which have not dominated this solution).

"Allocation of resources" structure.
The structure contains the information about the allocated resources in frames of solution cluster. This is the table representing the number of allocated solutions of each type. The table size equals the number of available resource types.

Example 1: The table of resource allocation contains the information that system structure consists of: five resources of RE1 type, two of RE2 and one of RE4: Table 1.

Table 1. Example 1

RE1	RE2	RE3	RE4
5	2	0	1

"Task attribution to resources" structure.
This structure describes behavior of tasks in system (attributing and schedule):

- Vector of lists describing task order in individual resources. Each list responds to one allocated resource.
- Vector including the times of beginning and ending of task execution.
- Vector describing task allocation to selected resource.

Example 2: The table of task attributing to resources contains the information that: tasks A and B will be executed by resource P, task C by resource R and task D by resource Q: Table 2.

Table 2. Example 2

Task A	Task B	Task C	Task D
P	P	R	Q

Cluster solutions describe the structure architecture of the system (allocated recourses). The solutions reflect the task allocation to recourses and task scheduling. During evolution two types of operations (crossbreeding and mutation) take place; they happen on two levels: cluster and solution. Clusters are the environment for the solution. Both solutions and the environments in which they occur are subject to evolution. The illustration Fig. 1 shows the relation which occur between the main data structure in algorithm.

"Task scheduling" structure.
It contains functions for task scheduling:

- Initial scheduling—ASAP algorithm (A Soon As Possible) [4].
- Mutation of schedule.
- Crossing of schedule.
- Function for the counting schedule length $\boldsymbol{C_{max}}$.

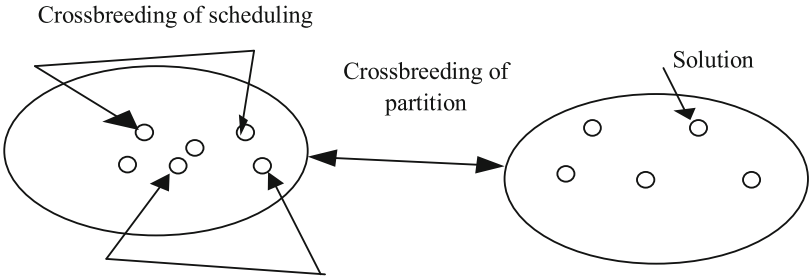


Fig. 1. Relations between clusters and solutions – crossing operation

"Global temperature" structure.
It contains information about the parameters of algorithm global temperature:

- Current temperature.
- Cooling ratio.
- Temperature step.

During the performance of algorithm, the parameter "temperature" will diminish with accordance to the function:

$$f(x) = e^{-ax} \tag{1}$$

where: ***a*** – is cooling ratio. Argument x is increased during the performance of algorithm by step of temperature reduction Fig. 2.

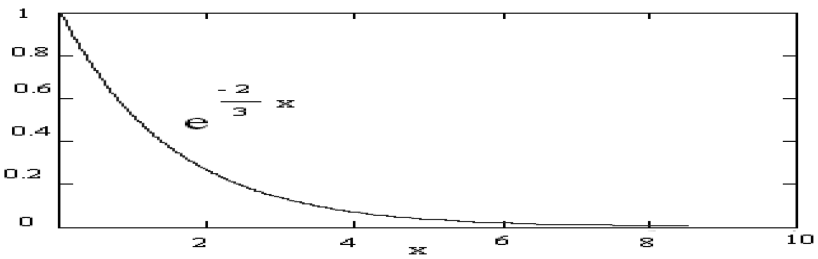


Fig. 2. Chart of reducing the temperature of algorithm

2.2 Resource Partition Algorithm

It tasks into account the data of task graph, pool of resources as well as criteria of optimality. The aim of this algorithm is to determine resources for execution all tasks in accordance with the criteria of optimality.

The aim of initializations of algorithm is to construct such structure of system which is the initial, the simplest and based on available resources and at the same time satisfying the demands of functions and given criteria.

Algorithm executes the following steps:

- Construction of tasks graph.
- Creating resources.
- Creating population.
- Initialization of hyper-sphere.
- Population initialization.
- Solution evaluation.
- Cluster evaluation.

Solution evaluation.
The following values are calculated depending on optimization criteria:

- Resource cost (processors: general, dedicated and operating memory).
- Task completion time.
- Power consumption.

The whole cost is the cost sums of allocated recourses; the entire time of executed tasks is the time of completing all the tasks on all the allocated recourses, the whole power consumption is the sum of power consumption taken by the selected recourses. If, for the representing individual, the unlimited solution of the optimization criteria excesses the maximal available number of value, the individual is fined. The suitable value for this individual is multiplied and the chance of survival definitely decreases. As a result of the above mentioned operations, we receive the table comprising the value of optimized criteria (time, cost, power consumption).

The following stage is to determine solution ranking. Ranking of a solution is the number of solutions in population which do not dominate this solution. Solution is dominated if each of its optimized criteria value is lower or equal to the value for the dominated solution (optimization in sense of Pareto) [5].

2.3 Main Algorithm of Resource Partition

The input data for resource partition (after initialization) are the task digraph, the library of available resources and the optimization criteria, and its goal is to divide tasks into the software and the hardware part and to select resources for the realization of all the tasks with established optimization criteria. The diagram of the algorithm of resource partition is showed on Fig. 3.

Clusters are reproduced with the use of genetic operators: crossing and mutation. At the reproduction stage, the cluster population is doubled and its initial size is restored at the elimination stage. This method was introduced arbitrarily and ensures that within a population some new individuals appear and fight for survival with their parents. The mutation operator creates one, and the crossing operator two new clusters. The likelihood of using either of the genetic operators is defined by the algorithm parameters.

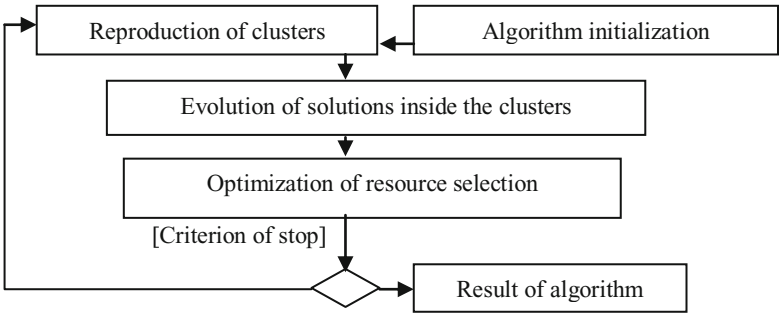


Fig. 3. Algorithm of resource partition

The algorithm for cutting the hyper-sphere with a hyper-plane.

Determining the cutting of hyper-plane by picking ***n*** points inside an n-dimensional hyper sphere.

- Creating a random permutation of n dimensions, e.g. for n = 3, the permutation can be (2, 1, 3).
- Constructing the point displacement vector in respect to the hyper- sphere center for ***n*** dimension:
  - Square coordinates are picked according to dimension permutations, e.g. for three dimensions with the permutation (2, 1, 3):
    y^2 = rand () % r^2, x^2 = rand () % ($r^2 - y^2$), z^2 = rand () % ($r^2 - (y^2 + x^2)$)
    Where: r – hyper-sphere radius, (x, y, z) are the coordinates of the constructed point in a three-dimensional space.
  - The roots of square coordinates are calculated.
  - A coordinate radical sign is picked.
  - The hyper-sphere center coordinates are added to the new point resulting in obtaining a new point inside the n-dimensional hyper-sphere.

The equation of the hyper-plane cutting the hyper-sphere is calculated and the obtained system of linear equations is solved.

After creating new clusters, initiating algorithm of task allocation and initiating algorithm of scheduling are put into motion.

Elimination of Clusters.
At this stage of the algorithm, half individuals are removed from the population. The initial number of individuals is restored. The elimination of individuals is carried out with use of Boltzmann tournament selection strategy [6].

Boltzmann Tournament.
The winner of the tournament is chosen on the base of calculating the following equation f(x) – Function 2.2, where: r1 – ranking of the first solution, r2 – ranking of the second solution, T – global temperature.

The values of this function are numbers of the range <0, 1>. In order to assign the winner of the tournament, we choose the number of the range (0, 1). If it is larger than

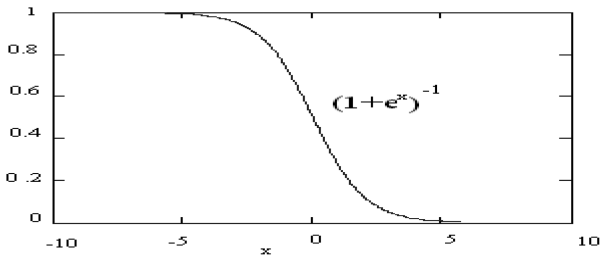


Fig. 4. The chart of probability of winning Boltzmann tournament depending on global temperature

the one enumerated out of the formula, the individual of ranking r1 is the winner. Otherwise, the second individual (of ranking r2) is the winner [6].

The analysis of the results of tournament can be carried out on the basis of function graph Fig. 4.

$$\mathrm{f}(x) = (1 + e^{\mathrm{x}})^{-1} \tag{2}$$

Where:

$$\mathrm{x} = \frac{(\mathrm{r1} - \mathrm{r2})}{\mathrm{T}}$$

If *r1* < *r2*, then *x* is negative, in this case when the temperature is high, there is a larger probability that an individual of rank *r1 will* win the tournament than when the temperature is lower. For low temperatures an individual of rank *r2* most frequently will be the winner.

If *r1* > *r2,* then *x* is positive, in case when the temperature is high, there is a larger probability that an individual *r2 will* win the tournament than for lower temperatures. For low temperatures, an individual of rank *r1* most frequently will be the winner.

2.4 Scheduling of Tasks

The diagram of the algorithm of task scheduling is showed in Fig. 5.

- The crossing operator of task allocation to resources resembles cluster crossing, the difference is that the task hyper-sphere is used for that purpose. This operation results in information exchange between the tables of task allocation to resources for individuals' crossbreeding. Thus, some of the tasks change their allocation to resources.
- Schedule crossing operator acts in the following way: After the allocations have been crossed, a map is created defining from which parent a given feature of an offspring comes. The offspring stores the allocation vector (obtained after crossing task allocations to resources) and an empty vector of lists with schedules of tasks on available resources. The algorithm analyses the tasks by checking their position on

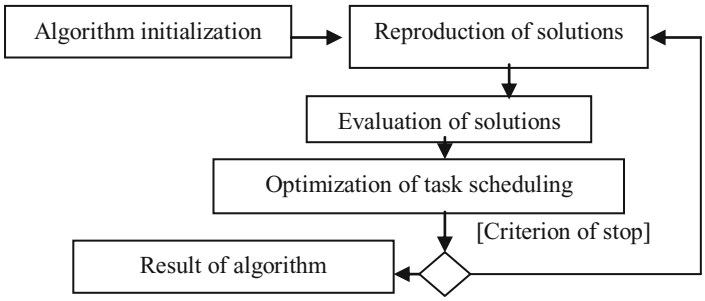


Fig. 5. Algorithm of task scheduling

the graph. For all the tasks on the same level, the resources on which the tasks will be performed (defined by the table of allocation to resources) are put on the list. If there is only one task executed on a given resource on the level, the task is enrolled into the resource schedule; otherwise the tasks are sorted according to the starting time they possessed in the parent and are placed in the schedule in ascending order. The last stage is to calculate (on the basis of sequence constraints and the times of task execution on processors to which they are allocated) the starting and ending times of executing each of them.

2.5 Coherent Resource Partition and Task Scheduling

The diagram of the algorithm of the coherent resources selection and task scheduling according to genetic approach is showed by Fig. 6. The initialization of algorithm resembles the initialization of resource partition algorithm.

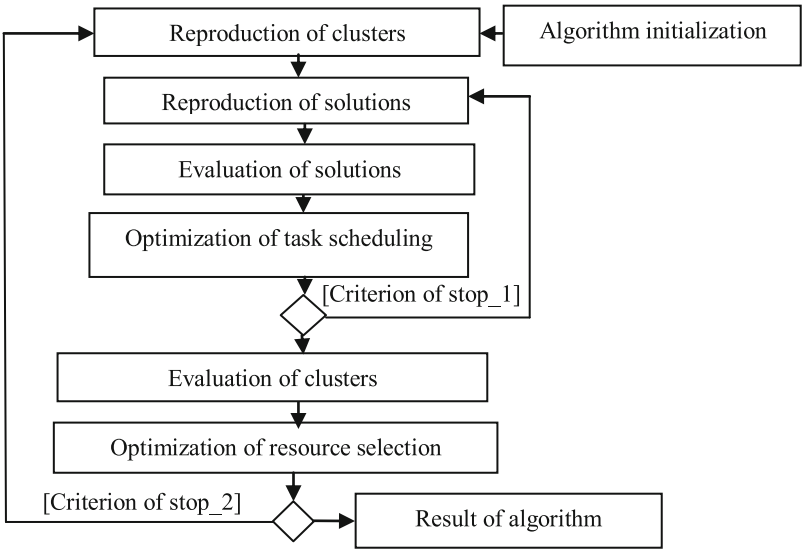


Fig. 6. Coherent synthesis of computer system – genetic approach

The input parameters are the number of clusters in the population and the number of solutions in clusters. Solution clusters represent the structures of system (of resources) sharing the same resource allocation, but with different task allocation to resources and different schedules. All the procedures within algorithms of resources and task scheduling are used in the algorithm of coherent synthesis.

The outer loop of the algorithm (realizes resource partition) is conducted until the number of generations without population improvement is exceeded. This value is defined by the parameter of algorithm.

The input parameters are the number of clusters in the population and the number of solutions in clusters. Solution clusters represent the structures of system (of resources) sharing the same resource allocation, but with different task allocation to resources and different schedules. All the procedures within algorithms of resources and task scheduling are used in the algorithm of coherent synthesis.

The outer loop of the algorithm (realizes resource partition) is conducted until the number of generations without population improvement is exceeded. This value is defined by the parameter of algorithm. The number of iteration of internal loop (realizes task scheduling) of the algorithm is defined by the pattern: The input parameters are the number of clusters in the population and the number of solutions in clusters.

Solution clusters represent the structures of system (of resources) sharing the same resource allocation, but with different task allocation to resources and different schedules. All the procedures within algorithms of resources and task scheduling are used in the algorithm of coherent synthesis. The outer loop of the algorithm (realizes resource partition) is conducted until the number of generations without population improvement is exceeded. This value is defined by the parameter of algorithm. The number of iteration of internal loop (realizes task scheduling) of the algorithm is defined by the pattern:

$$f(x) = -k\ (e^{-ax})^3 + k \tag{3}$$

Where: k – the parameter of algorithm, a – the cooling parameter.

Argument $\boldsymbol{x}$ with <0, n> in every generation is enlarged by the step of temperature. At the beginning of the performance of algorithm, internal loops are scarce – Fig. 7.

Their number grows until it reaches the value of $\boldsymbol{k}$ with the falling of the temperature. At the beginning when system structure is not close enough to optimum, fewer task allocations and scheduling are performed. When the temperature falls

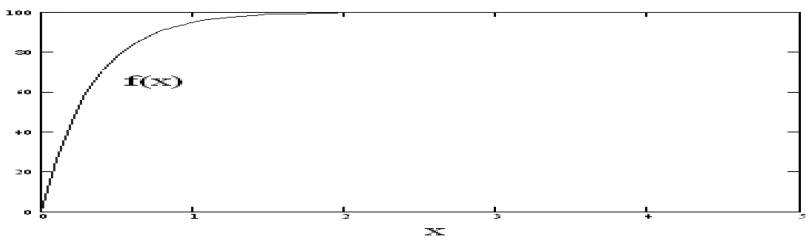


Fig. 7. Graph f(x) for k = 100

sufficiently low, the number of iterations for scheduling algorithm is k. The number of iterations may be regulated with the temperature step parameter. The greater the step the faster the number of inner iterations reaches the k value.

3 Calculative Experiments. Comparing the Results of Coherent and Non Coherent Synthesis

Resources applied in tests are shown in the following table Table 3. The tasks (system functions for specification) used during the tests are generated as digraphs and they are received as: Graphs STG [http://www.kasahara.elec.waseda.ac.jp/schedule] and Graphs TGFF [http://ziyang.eecs.umich.edu/∼dickrp/tgff]. These generators have been worked out in order to standardize random tests for research into common task scheduling and allocating problems, especially for system synthesis and for such applications which need pseudo-random generating acyclic directed graphs. In generators, the sort of graph, number of source and sink nodes, the length of maximal track, the node and edge weight, degree of graph, probability of predecessors and successors' number etc. should be determined. For example, time of tasks might be generated as follows: the average time value for the task (e.g. =5 units) and time of tasks determined by uniform distribution or regular distribution with a fixed standard deviation (e.g. =1). For the tests, maximum number of tasks have been determined =100 (independent tasks) and 50 (dependent tasks), which is the sufficient number for the presentations of all algorithm features, and their comparisons as well (also to other algorithms) and is also the right number of operations for realistic system synthesis; obviously, system functions in its specification are given on the suitable level of granulation.

Table 3. Resources applied in tests

No.	Type of processor	ID	Speed	Cost	Power consumption
1	General	P1	1	1.00	0.01
2	General	P2	2	1.60	0.02
3	General	P5	5	2.20	0.05
4	General	P10	10	3.70	0.1
5	Dedicated	ASIC1	1	0.50	0.01
6	Dedicated	ASIC2	2	0.75	0.02
7	Dedicated	ASIC3	3	1.00	0.03
8	Dedicated	ASIC4	4	1.25	0.04
9	Dedicated	ASIC5	5	1.50	0.05
10	Dedicated	ASIC10	10	2.75	0.11
11	Memory modules	PAO	1	0.2	0.001

This point shows computational results of research obtained during testing the presented algorithms. The results are gathered in the tables and charts for comparing the results obtained by coherent algorithm with the results obtained by non-coherent algorithm. The aim of the tests is to check whether the solution generated by coherent

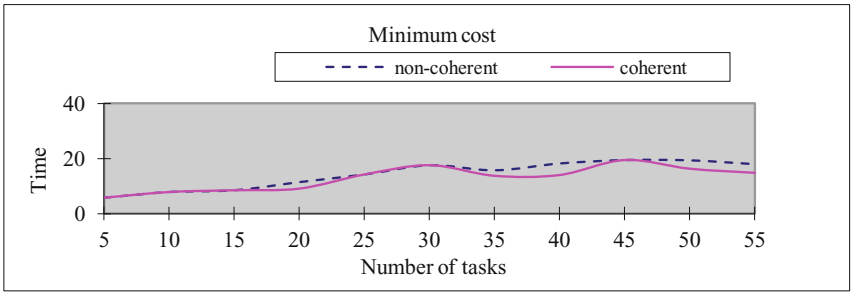


Fig. 8. Cost minimization – time and number of tasks

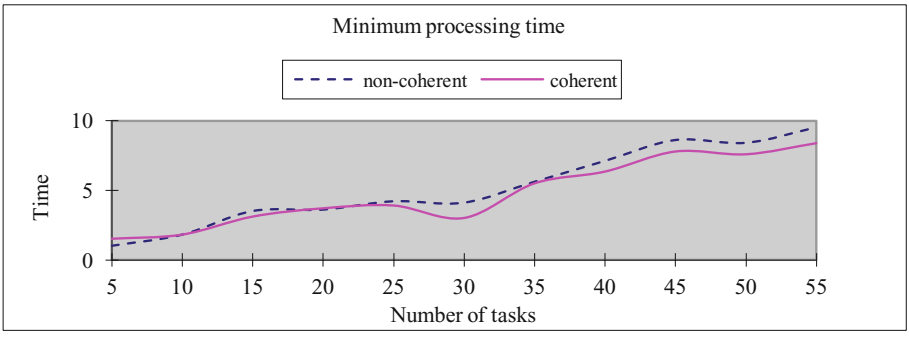


Fig. 9. Time minimization – time and number of tasks

algorithm is better. The research was conducted into dependent tasks. Additionally, power consumption was calculated by coherent algorithm during completing all the tasks. During cost optimization, both algorithms acquired similar cost values for all tested task sets. However, the coherent algorithm improved time for graphs exceeding 30 tasks. For 45 and 50 tasks it achieved a 15% improvement of the task completion time Fig. 8. When the flowchart reflecting the dependence of time on the numbers of system tasks is considered Fig. 9, time minimization is comparable for both algorithms. Nevertheless, once the chart showing the interdependence of cost and the number of tasks is analyzed Fig. 10, it is clear that the solutions acquired by the coherent algorithm are far less expensive than those acquired by the non-coherent algorithm (e.g. for number 45 tasks and in generally for number greater than 20). The coherent algorithm achieves similar task completion times in solutions much cheaper from those found by the non-coherent algorithm.

The above presented charts show multi-criteria optimization for coherent synthesis of computer systems. As a result of algorithm performance the designer receives a set of optimal solutions in sense of Pareto. The designer has to decide which resource fulfills the requirements of the solution. Depending on the system requirements, it is possible to rely on one of the obtained results. To learn the specification of the solution space for the given problem instance, it is important to provide a sufficiently long list of

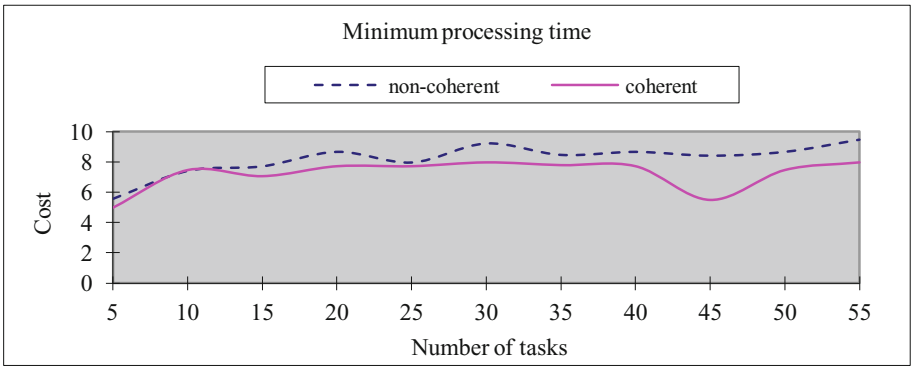


Fig. 10. Time minimization – cost and number of tasks

the best saved solutions (e.g. In tests of algorithm parameters 'the best solution number" has been fixed as the value of 50). It is also important to define slow cooling of the algorithm (parameters "temperature step" and "cooling coefficient" – depending on the numbers of tasks in the system). Thanks to this, we prevent the population from too big convergence. Algorithm searches for the lower temperature, a bigger area in the space of solutions. It has also been noticed that a bigger probability of mutation helps to look for a better system structure, whereas a bigger probability of crossing improves optimization of time criterion.

4 Conclusion

This paper is about the problems of coherent synthesis of computer systems. Such a synthesis is carried out on a high level of abstraction in which system specification consists of a set of functions i.e. required operations, which should be implemented by a sets of resources and these are listed in the database (in container, pool, or a catalogue) and are available i.e. do exist or can be created. Resources possess certain constraints and characteristics, including speed, power, cost and reliability parameters. Thus such a synthesis concerns systems of the following type: resource and operation complex and the problems of resource partitioning and their selection as well as scheduling and sequencing of operations performed on these resources are determined on this level. Optimization of afore mentioned synthesis actions occurs on the same level. When one possesses operations with system specifications and one possesses selected resources, software and hardware which can perform these operations, as well as defined control which allocates tasks and resources and schedules operations, then one possesses general knowledge necessary for the elaboration of design details – to define standard and with specific parameters, physical processors and software modules, to apply available components and, when it is necessary to prepare special hardware subcomponents and software modules for the implementation of the whole system of greater efficiency i.e. cheaper, faster, consuming less power.

Problems of high-level synthesis in coherent – our – approach are solved simultaneously and globally and as it is confirmed by calculation experiments, the solutions of these problems are more efficient than non-coherent ones, which were in previous approaches. Of course these problems, as mentioned earlier are computationally NP-complete [9], as a result there is a lack of effective and accurate algorithms, thus one has to use heuristics in solutions. In the paper one proposed the so called hybrid implementation methods genetic with simulated annealing. Obviously these methods were chosen out of many and proposed adaptations of these methods can be different. However, for method and adaptations presented herein calculations clearly point out the advantage of coherent synthesis, i.e. joint optimization of resource partitioning and, then on selected resources, task scheduling over non-coherent synthesis, where resource partitions and task scheduling are optimized separately.

Algorithms presented in this work can be utilized to prepare tools which automatically implement such a synthesis.

The solution suggested in the paper may be applied in supporting computer system prototyping, for dependable and fault-tolerant multiprocessors systems and grid system, too. The model presented for coherent synthesis and the experimental results allow a further research in this area. For example, other heuristics may be applied. One may also specify additional optimality criteria.

References

1. Drabowski, M.: Modification of concurrent design of hardware and software for embedded systems—a synergistic approach. In: Grzech, A., Świątek J., Wilimowska, Z., Borzemski, L. (eds.) Information Systems Architecture and Technology, Proceedings of 37th International Conference on Information Systems Architecture and Technology-ISAT 2016, Advances in Intelligent Systems and Computing, vol. 522, pp. 3–13. Springer, Heidelberg (2017)
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
3. Drabowski, M.: Tabu search and genetic algorithms in par-synthesis of multiprocessors systems. In: Proceedings of the IASTED International Conference on Artificial Intelligence Applications, Innsbruck, pp. 146–151. ACTA Press, Anaheim (2009)
4. Yen, T.Y., Wolf, W.H.: Performance estimation for real-time distributed embedded systems. IEEE Trans. Parallel Distrib. Syst. **9**, 1125–1136 (1998)
5. Coffman Jr., E.G.: Computer and Job-Shop Scheduling Theory. Wiley, New York (1976)
6. Aarts, E.H.L., Korst, J.: Simulated Annealing and Boltzmann Machines. Wiley, Chichester (1989)
7. Dick, R. P., Jha, N. K: MOCSYN: Multiobjective core-based single-chip system synthesis. In: Proceedings of Design Automation and Test in Europe Conference, pp. 263–270 (1999)
8. Dick, R.P., Jha, N.K.: MOGAC: a multiobjective genetic algorithm for hardware–software cosynthesis of hierarchical heterogeneous distributed embedded systems. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **17**(10), 920–935 (1998)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)

Model Transformation Method for Hybrid Approach

Paweł Sitek(✉), Jarosław Wikarek, and Tadeusz Stefański

Institute of Management and Control Systems, Kielce University of Technology,
Al. 1000-lecia PP 7, 25-314 Kielce, Poland
{sitek, j.wikarek, t.stefanski}@tu.kielce.pl

Abstract. This paper presents a universal model transformation method for the problem which is represented in the form of facts. This method is the key element of the hybrid approach. In this approach, two environments of constraint logic programming (CLP) and mathematical programming (MP) were integrated and hybridized. Hybrid approach is very effective method for modeling and solving different kinds of decision and optimization problems in manufacturing, transportation, logistics etc. These problems are characterized by large numbers of integer variables and constraints, which build a large space of possible solutions. The model transformation method makes it possible to significantly reduce this space even before looking for a solution.

The paper also presents an author's illustrative model for variant of CVRP (Capacity Vehicle Routing Problem) and many numerical experiments to test the model transformation method and evaluate its effectiveness when using a hybrid approach.

Keywords: Constraint logic programming · Mathematical programming · Hybrid methods · CVRP

1 Introduction

Decision and optimization problems are ubiquitous in our everyday life and other fields, such as industry, finance, logistics, business, trade and transportation, to name only a few. Many important or necessary decision problems can be formulated as feasibility or optimization problems, in which the question is to decide the values of various decision variables representing a quantity, time, allocation, selection, location etc. Most often numerous sets of constraints are imposed on these problems – then we talk about constrained decision problems. Often, one has to find a solution to a situation by searching in a finite, but generally huge collection of alternatives.

In practice, the above-mentioned areas, most problem models are mathematical programming (MP) [1, 2] models based on linear programming-based approach, in particular on mixed integer linear programming (MILP) or integer linear programming (ILP). The primary drawbacks of such problem models based on mathematical programming (MILP, IP, ILP etc.) include linearity of constraints and limited efficiency of combinatorial optimization for problems of larger size. Declarative environments are not only free of these shortcomings but also make use of CSPs (constraint satisfaction

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_31

problems) and CLP (constraint logic programming), which contributes to their high potential in the modeling and solving of such problems [3, 4]. Declarative environments show some limitations when numerous decision variables are added up in multiple constraints. Such constraints are common in constrained decision problems in industry, logistics, and transportation.

Therefore, based on the experience, research and use of both environments in modeling and solving such problems [5–7] a hybrid approach has been proposed [8]. This approach is unique and innovative in that it involves original integration of two mutually reinforcing methods/paradigms, MP and CLP. The key element of the hybrid approach is model transformation as shown in [8]. This paper presents a universal model transformation method based on representation of a problem in the form of facts. Developing effective algorithm for model transformation, that significantly reduces the number of variables and constraints of the modeled problem is main contribution of this paper.

The rest of this paper is organized as follows. Section 2 conveys the fundamentals of hybrid approach. Section 3 shows model transformation method. An example illustrating the model transformation methodology is provided in Sect. 4. Section 5 gives a summary and future work.

2 Hybrid Approach for Modeling and Solving Decision Problems

Based on literature [3–7, 9] and our previous studies [8, 10–12] was observed some advantages and disadvantages of both CLP-based and MP-based approaches. Taking into account these studies and experiences with both environments, a hybrid approach has been proposed for modeling and solving decision problems.

Main assumptions of the proposed hybrid approach were as follows: (a) integration of CLP and MP environments, (b) use of strong points and compensation of weak points in terms of problem modeling and optimization revealed in both environments, (c) problem data representation in the form of sets of facts with a suitable structure based on the relational database, (d) introduction of model transformation as a presolving method, (e) substantial reduction of the feasible solution space for the post-transformation models, and (f) automatic generation of implementing models and their translation into the MILP/ILP form.

Figure 1 presents the general concept of the hybrid approach implementation as an implementation platform. The hybrid approach comprises several phases: modeling (CLP model is formed), presolving (after transformation CLP^T model is formed), generating ($MILP^T$ is generated) and solving. It has two inputs and uses the set of facts. Inputs are the set of constraints and the objectives to the reference model of a given problem.

One of the characteristic features of the hybrid approach is that the problem is represented in the form of facts (facts-based representation). Decision and optimization problems are typically written as algebraic expressions with coefficient matrices and vectors of decision variables, as in mathematical programming. The new representation (facts-based) reduces the size of the problem. This reduction can be referred to as

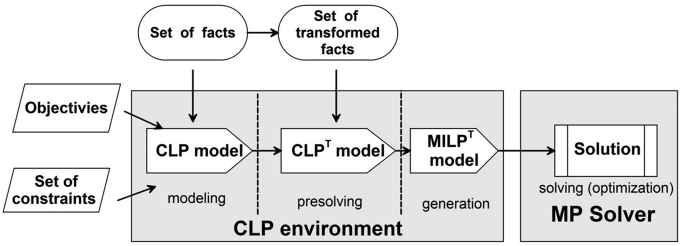


Fig. 1. The schema of hybrid approach implementation

"hidden" transformation resulting only from the change in problem representation. With algebraic notation, the status is usually shown as a matrix of coefficients and the decisions as a vector of decision variables. If a status is unacceptable, its corresponding parameter is "0". Then the corresponding decision parameters are zeroed but still present in the model representation. This increases the size of the model, thereby increasing the efforts necessary for solving it, and in some cases, leading to the situation in which the model is too big to be solved by a given MP solver. The change of representation (from algebraic to facts-based) provides the same effect "for free", with no need to use advanced methods or algorithms. The change of representation is shown symbolically in Fig. 2.

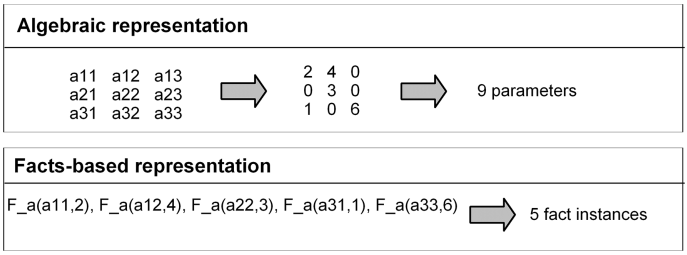


Fig. 2. Algebraic representation versus fact-based representation

3 Model Transformation Method – Assumptions, Algorithm and Implementation

The fact-based representation (Fig. 2) was used to propose another, "non-hidden", transformation of the model. For this purpose was developed the algorithm for its transformation. The transformation reduced the number of decision variables and constraints of the modeled problem, which results in the reduction and transformation of solution space relative to the model prior to transformation. It is based on the analysis of the instances of the facts that represent the problem. In short, it involves the removal of unacceptable points form solution space. Let us use the simple example. There are facts: *products(id_product, name), order(order_id, id_product, quantity)*,

fpp(fpp_id), *production(fpp_id, id_product, cost)*, *delivery(delivery_id, id_product)*, etc. which describe the simple problem of fulfill customer orders in food industry. The ordered products are produced in food processing plant (*fpp*) and delivered to customers in a certain way. Table 1 shows instances of the facts for this example. Analysis of data instances in Table 1 indicates that: (a) not all products are ordered (only 5 of 10), (b) not all products can be produced in all food processing plant, and (c) not all products can be delivered by any method.

Table 1. Facts and instances for a simple example

Fact name	Instances
products(#p, ...)	*products(p1, pork), products(p2, beef), products(p3 eggs), products(p4, chicken), products(p5, turkey), products(p6, oil), products(p7, bacon), products(p8 duck), products(p9, goose), products(p10, olive oil)*
order(#z, #p, ...)	*order(z1, p2,10), order(z2, p2,20), order(z3, p2,40), order(z4, p1,100), order(z5, p6,150), order(z6, p10,200), order(z7, p3,1200),*
fpp(#f, ...)	*fpp(f1), fpp(f2), fpp(f3)*
production(#f, #p, ...)	*production(f1, p1,2), production(f1, p2,5), production(f1, p7,2), production(f2, p3,1), production(f2, p1,2), production(f2, p5,3), production(f2, p8,4), production(f2, p9,4), production(f3, p6,3), production(f3, p10,4)*
delivery(#d, #p)	*delivery(d1, p1), delivery(d1, p2), delivery(d1, p4), delivery(d1, p5), delivery(d1, p7), delivery(d1, p8), delivery(d1, p9), delivery(d2, p3), delivery(d2, p6), delivery(d2, p10)*

- the key attribute of the fact (*#p* - id_product, *#z* - order_id, *#f* - fpp_id, *#d* - delivery_id)

These are the basic inadmissibility criteria R for the example analyzed in Table 1, emerging from the fact-based representation.

In the algebraic representation of the same example, the decision variable $X_{p,f,d}$ denoting the quantity of the product p produced in the factory f and delivered by the method d determines the space of $10 \times 3 \times 2 = 60$ potential points, whereas in the facts-based representation and after the analysis of instances (Table 1), only $(2 + 1 + 1 + 1 + 1) = 6$ points form the solution space ($X_{p1,f1,d1}$, $X_{p1,f2,d1}$, $X_{p2,f1,d1}$, $X_{p6,f3,d2}$, $X_{p10,f3,d2}$, $X_{p3,f2,d2}$). In this way, a tenfold reduction in the solution space is achieved (60/6 = 10).

The main assumptions of the proposed model transformation method are as follows: (a) the problem is represented in the form of facts, (b) the problem, in the form of a CSP, is modeled using constraints, variables and their domains, and (c) the transformation algorithm should have at most polynomial computational complexity

The model transformation algorithm consists of three procedures: *proc_1*, *proc_2*, and *proc_3* called sequentially. The procedure *proc_1*, based on the analysis of fact instances, determines whether the model transformation is viable and determines the fact/facts of the transformation. This viability is expressed as a measure of the magnitude of the reduction in the model solution space before and after transformation. The procedure *proc_2*, based on the fact/facts of transformation defined by *proc_1* and the set of R criteria of inadmissibility reduces the solution space i.e. removes those points

where finding a solution is pointless. Criteria R are determined on the basis of the analysis of fact instances and selected model constraints. The procedure *proc_3* is crucial to the whole algorithm as it allows the facts, decision variables, parameters, and constraints of the model to be transformed. The outcome of *proc_3* is a transformed model. The execution of the procedure *proc_3* for the transformation of facts for the illustrative example is shown in Fig. 3.

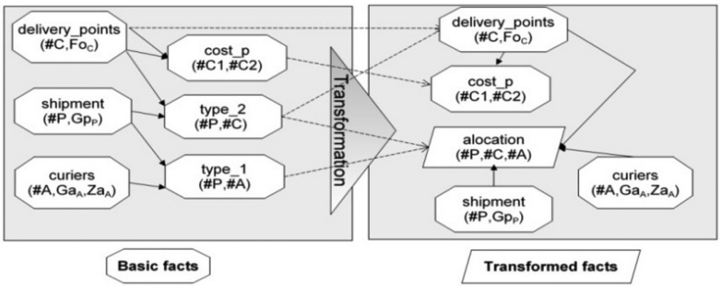


Fig. 3. The set of facts for illustrative example

4 Illustrative Example – The Variant of CVRP

As an illustrative example of the application of the model transformation method, a significantly modified version of CVRP (Capacity Vehicle Routing Problem) was chosen [13–16]. In this variant distributor/depot d delivers shipments p ($p \in P$, P – the set of all shipments) to customers' c ($c \in C$, C – the set of all customers). Shipment is a palette/basket of different products. Distributor can send several shipments to the customer. These shipments are delivered by couriers/suppliers a ($a \in A$, A – the set of all couriers/suppliers). Courier/supplier can take the shipment to several customers at once, can also do several courses from the distributor d. Shipments have dimensions. During the course courier/supplier can only take so many shipments, do not to exceed its capacity.

Shipments and couriers are different types. Therefore, the factor G_{ap} is introduced. ($G_{ap} = 1$ means that the courier a can carry the shipment p and $G_{ap} = 0$ otherwise).

In addition, it is assumed that:

- There are known times of travel between distributor and customer as well as the time it takes to load/unload the shipment;
- there is no possibility of reloading shipments and that not every courier can reach every customer;
- the total time/cost of all deliveries should be as short/low as possible (time is proportional to delivery costs);

- the working time of any courier/supplier was no higher than the given *Tw* (optional);
- some deliveries to selected customers must be in a specific time window–time interval (optional).

The illustrative problem was formulated in the form of constraints (2)–(14) with objective function (1) – minimization of travel time. Essential parameters and decision variables are listed in Table 2. A description of constraints is provided in Table 3.

$$\min \sum_{a\in A} \sum_{j\in C\cup d} \sum_{i\in C\cup d} \sum_{l=1\ldots Za_a} X_{a,j,i,l} \cdot B_{i,j} \tag{1}$$

$$\sum_{j\in C\cup d} X_{a,i,j,l} = \sum_{j\in C\cup d} X_{a,j,i,l} \; \forall a \in A, i \in C\cup d, l = 1\ldots Za_a \tag{2}$$

$$X_{a,i,j,l} \le \sum_{p\in P} Y_{a,i,j,p,l} \; \forall a \in A, i \in C\cup d, j \in C, l = 1\ldots Za_a \tag{3}$$

$$Y_{a,i,j,p,l} \le X_{a,i,j,l} \; \forall a \in A, i \in C\cup d, j \in O\cup d, p \in P, l = 1\ldots Za_a \tag{4}$$

$$\sum_{p\in P} Gp_p \cdot Y_{a,i,j,p,l} \le Ga_a \cdot X_{a,i,j,l} \; \forall a \in A, i \in C\cup d, j \in C\cup d, l = 1\ldots Za_a \tag{5}$$

$$\sum_{a\in A} \sum_{j\in C} \sum_{l=1\ldots Za_a} Y_{a,d,j,p,l} = 1 \; \forall p \in P \tag{6}$$

$$\sum_{j\in C} \sum_{l=1\ldots LK_k} X_{a,d,j,l} \le Za_a \; \forall a \in A \tag{7}$$

$$\sum_{a\in A} \sum_{j\in C\cup d} \sum_{l=1\ldots Za_a} G_{a,p} \cdot Y_{a,j,c,p,l} - \sum_{a\in A} \sum_{j\in c} \sum_{l=1\ldots Za_a} G_{a,p} \cdot Y_{a,c,j,p,l} = U_{p,c} \; \forall p \in P, c \in C \tag{8}$$

$$\sum_{i\in C\cup d} \sum_{j\in C\cup d} \sum_{l=1\ldots Za_a} Y_{a,d,i,p,l} \le WX \cdot S_{a,p} \; \forall a \in A, p \in P \tag{9}$$

$$\sum_{a\in A} S_{a,p} = 1 \; \forall p \in P \tag{10}$$

$$TX_{a,i,l} - WX \cdot (1 - X_{a,i,j,l}) + B_{i,j} \cdot X_{a,i,j,l} \le TX_{a,j,l} \; \forall a \in A, i \in C\cup d, j \in C, l = 1\ldots Za_a \tag{11}$$

$$Y_{a,i,j,p} = \{0,1\} \; \forall a \in A, i \in C\cup d, j \in C\cup d, p \in P \tag{12}$$

$$X_{a,i,j} = \{0,1\} \; \forall a \in A, i \in C\cup d, j \in C\cup d \tag{13}$$

$$S_{a,p} = \{0,1\} \; \forall a \in A, p \in P \tag{14}$$

Table 2. Parameters and decision variables for illustrative model

Symbol	Description
Parameters	
Gp_p	Shipments volume (volumetric weight) p $(p \in P)$
Ga_a	Courier's vehicle volume a $(a \in A)$
Za_a	Maximum number of courses courier a can take $(a \in A)$
Fo_c	Time of unloading shipments at customer c $(c \in C)$
$G_{p,a}$	If courier a $(a \in A)$ may carry shipment p $(p \in P)$ then $G_{p,a} = 1$ otherwise, $G_{p,a} = 0$
$B_{i,j}$	Travel time from the point i $(i \in d \cup C)$ to point j $(j \in d \cup C)$
$U_{p,c}$	If shipment p $(p \in P)$ is destined to the customer c $(c \in C)$ then $U_{p,c} = 1$ otherwise $U_{p,c} = 0$
WX	Large constant
Decision variables	
$X_{a,i,j,l}$	If courier a travels from point i to point j by course l than $X_{a,i,j,l} = 1$, otherwise $X_{a,i,j,l} = 0$, $(a \in A,\ i \in C \cup d,\ j \in C \cup d,\ l = 1 \ldots Za_a)$
$Y_{a,i,j,p,l}$	If courier a travels from point i to point j by course l carrying item p than $Y_{a,i,j,p,l} = 1$, otherwise $Y_{a,i,j,p,l} = 0$, $(a \in A,\ i \in C \cup d,\ j \in C \cup d,\ p \in P,\ l = 1 \ldots Za_a)$
$S_{a,p}$	If shipment p is delivered by courier a then $S_{a,p} = 1$, otherwise $S_{a,p} = 0$ $(a \in A,\ p \in P)$
$TX_{a,i,l}$	The time when courier arrives to customer or distributor, $(a \in A,\ i \in C \cup d,\ l = 1 \ldots Za_a)$

Table 3. Description of constraints for illustrative model

Constraint	Description
(2)	Arrival and departure courier from the delivery point (distributor, customer)
(3)	If no shipments are to be carried on the route, a courier does not travel that route
(4)	If a courier does not travel along a route, no shipments are to be carried on that route
(5)	At no route segment courier carries more items than allowable courier's vehicle volume
(6)	The shipment must be downloaded from the distributor
(7)	Specifies the number of courier courses
(8)	Shipments picked up/delivered to a delivery point
(9)	Which courier delivers which shipment
(10)	One shipment delivered by only one courier
(11)	Specifying the moment of the time when the courier arrives at the point
(12–14)	Binarity

5 Implementation and Computational Experiments

The illustrative example (Sect. 4) was implemented both in the hybrid approach with model transformation and in the classical MP approach. The set of facts describing the problem is shown in Table 7 and in Fig. 3. The transformation of facts was realized using the procedure *proc_3* (Fig. 4) of the algorithm and inadmissibility criteria *R1⋯R4*, shown in Table 6. As a result of the transformation, new fact was created *allocation()*, on the basis of which only admissible delivery routes for couriers were generated.

Following the analysis of instances of the fact *type_2(#P,#C),* the set *C* of the customers to which the shipments were sent is determined.

Following the analysis of instances of the fact *delivery_points (#C),* instances for which no deliveries are removed.
Following the analysis of instances of the fact *type_2(#P,#C)*, the set *P* of the all shipment is determined.
Following the analysis of instances of the fact *couriers(#A),* the set *A* of the all couriers is determined.
For $c \in C$; For $p \in P$; For $a \in A$

For existing instances of facts *type_1(#P,#A)* and *type_2(#P,#C)*

Create fact *allocation(#P,#C,#A)*

From the fact *cost_p(#C1,#C2),* instances for which no deliveries are removed

Fig. 4. Transformation facts for illustrative example (*proc_3*)

Multiple computational experiments were performed after the implementation (1)–(14). In the first stage, the problem was optimized for the data instances varying in the number of parcels (from $P = 10$ to $P = 50$); the other parameters, $C = 20$, $A = 6$ were left unchanged. In the second stage, time constraints on the route length *Tw* were introduced. The results are summarized in Tables 4 and 5, respectively. In addition, for the data instances containing $P = 30$, the solutions found were represented as the schemes of transport networks and routes for individual couriers (Fig. 5).

Table 4. The results of numerical experiments for both approaches

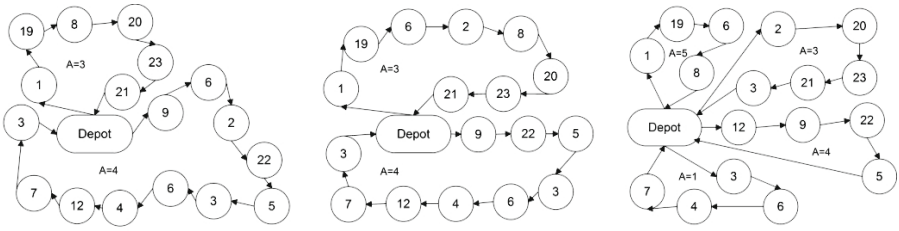
N	P	MP				Hybrid with transformation			
		V	C	Fc	Tc	V	C	Fc	Tc
1	10	29 420	37 452	118	732	6 834	39 562	118	31
2	20	55 940	64 192	147[b]	1200[a]	10 894	68 412	123	47
3	30	82 460	90 932	274[b]	1200[a]	14 954	97 262	243	81
4	40	108 980	117 672	NSF	1200[a]	19 014	126 112	253	123
5	50	135 500	144 412	NSF	7200[a]	23 074	154 962	265	351

V - the number of (integer) decision variables, C - the number of constraints, Fc - objective function,
Tc - calculation time, P - the number of shipments, N - example number, NSF - no solution found,
[a] - Interrupt the process of finding a solution after a given time 1200/7200 s,
[b] - Feasible solution (not found optimality)

Table 5. The results of numerical experiments for examples with time windows

N	P	Tw ≤ 100		Tw ≤ 200		Tw ≤ 400	
		Fc	Tc	Fc	Tc	Fc	T
1	10	190	45	118	31	118	31
2	20	210	56	123	47	123	47
3	30	294	96	260	96	243	94
4	40	310	56	310	141	253	134
5	50	NSF	45	265	351	265	351

Tw - time window



(a) (b) (c)

Fig. 5. Schemes of transport networks and routes for: (a) Tw ≤ 400, (b) Tw ≤ 200 (c) Tw ≤ 100

6 Conclusion

Compared to the MP approach, the hybrid approach with the model transformation method offers (a) three to six times reduction in the number of decision variables, (b) more than 35 times acceleration of optimization time, and (c) optimization of larger size examples. The data obtained result from a considerable reduction in the solution space. The reduction was possible owing to the transformation algorithm, in particular by dint of fact instances reduction and their transformation (*proc_3*). In the proposed hybridized framework, additional constraints can be introduced without changing the model as illustrated with the constraints on the couriers' working hours introduced at the fact transformation stage (only the instances that meet the time criterion remain).

In future research, the hybrid approach is planned to be complemented with fuzzy logic [17] and complex systems [18]. It is also considering development of models to take account product demand interdependencies [19]. It is planned to integrate proposed model with ERP and APS systems [20] and as a cloud internet application [21].

Appendix A. Criteria *R* and Description of Facts

See (Tables 6 and 7).

Table 6. Description of inadmissibility criteria *R* for the illustrative example

Criteria	Description
R1	Following the analysis of instances of the *shipment()* fact, only the corresponding instances of the *type_1(), type_2()* facts are retained
R2	Following the analysis of instances of the *type_2()* fact, only the corresponding instances of the *delivery_points()* fact are retained
R3	Following the analysis of instances of the *type_1()* fact, only the corresponding instances of the *couriers()* fact are retained
R4	On the basis of the facts *type_1(), type_2()* is created fact *allocation()*

Table 7. Description of facts for illustrative example

Name	Description
delivery_points(#C, Fo_c)	A fact that describes customers
shipment(#P, Gp_p)	A fact that describes shipments
couriers(#A, Ga_a, Za_a)	A fact that describes the couriers and their properties
type_1(#P, #C)	A fact describes the relationship between the shipment and the customer
type_2(#P, #A)	A fact is that the selected courier can deliver the selected shipment
cost_p(#C, #C $B_{c,c}$)	A fact describes distance (cost) between customers
allocation(#P, #C#, #A)	The fact after transformation determines which shipment is delivered by which courier to which recipient

References

1. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, Hoboken (1998). ISBN: 0-471-98232-6
2. Sinha, S.M.: Mathematical Programming. Elsevier Science, Amsterdam (2005). ISBN: 9780080535937
3. Apt, K., Wallace, M.: Constraint Logic Programming using Eclipse. Cambridge University Press, Cambridge (2006)
4. Rossi, F., Van Beek, P., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York (2006)
5. Bockmayr, A., Kasper, T.: Branch-and-infer: a framework for combining CP and IP. Constraint Integer Program. Oper. Res. **27**, 59–87 (2004)
6. Milano, M., Wallace, M.: Integrating operations research in constraint programming. Ann. Oper. Res. **175**, 37–76 (2010). doi:10.1007/s10479-009-0654-9
7. Hooker, J.: Logic optimization, and constraint programming. J. Comput. **14**(4), 295–321 (2002)

8. Sitek, P., Wikarek, J.: A hybrid programming framework for modeling and solving constraint satisfaction and optimization problems. Sci. Program. **2016**, 13 (2016). doi:10.1155/2016/5102616
9. Bocewicz, G., Nielsen, I., Banaszak, Z.: Production flows scheduling subject to fuzzy processing time constraints. Int. J. Comput. Integr. Manuf. **29**(10), 1105–1127 (2016). doi:10.1080/0951192X.2016.1145739. Article ID 5102616
10. Sitek, P.: A hybrid approach to the two-echelon capacitated vehicle routing problem (2E-CVRP). Adv. Intell. Syst. Comput. **267**, 251–263 (2014). doi:10.1007/978-3-319-05353-0_25
11. Sitek, P., Nielsen, I.E., Wikarek, J.: A hybrid multi-agent approach to the solving supply chain problems. Procedia Comput. Sci. **35**, 1557–1566 (2014). doi:10.1016/j.procs.2014.08.239
12. Sitek, P., Wikarek, J.: A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. Int. J. Prod. Res. **53**(21), 6611–6628 (2015). doi:10.1080/00207543.2015.1005762
13. Jairo, R., Montoya, T., Francob, J.L., Isazac, S.N., Jiménezd, H.F., Herazo-Padillae, N.: A literature review on the vehicle routing problem with multiple depots. Comput. Ind. Eng. **79**, 115–129 (2015)
14. Kumar, S.N., Panneerselvam, R.: A survey on the vehicle routing problem and its variants. Intell. Inf. Manag. **4**, 66–74 (2012)
15. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. Eur. J. Oper. Res. **202**(3), 756–763 (2010). doi:10.1016/j.ejor.2009.06.034
16. Perboli, G., Tadei, R., Vigo, D.: The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transp. Sci. **45**, 364–380 (2012)
17. Kłosowski, G., Gola, A., Świć, A.: Application of fuzzy logic in assigning workers to production tasks. In: 13th International Conference on Distributed Computing and Artificial Intelligence, AISC, vol. 474, pp. 505–513 (2016). doi:10.1007/978-3-319-40162-1_54
18. Grzybowska, K., Łupicka, A.: Knowledge acquisition in complex systems. In: Yue, X.-G., Duarte, N.J.R. (eds.) Proceedings of the 2016 International Conference on Economics and Management Innovations, vol. 57, pp. 262–266 (2016). doi:10.2991/icemi-16.2016.5
19. Nielsen, P., Nielsen, I., Steger-Jensen, K.: Analyzing and evaluating product demand interdependencies. Comput. Ind. **61**(9), 869–876 (2010). doi:10.1016/j.compind.2010
20. Krenczyk, D., Jagodzinski, J.: ERP, APS and simulation systems integration to support production planning and scheduling. In: Advances in Intelligent Systems and Computing, vol. 368, pp. 451–46. Springer International Publishing, New York (2015)
21. Bak, S., Czarnecki, R., Deniziak, S.: Synthesis of real-time cloud applications for internet of things. Turk. J. Electr. Eng. Comput. Sci. (2013). doi:10.3906/elk-1302-178

Hurst Exponent as a Risk Measurement on the Capital Market

Anna Czarnecka[1(✉)] and Zofia Wilimowska[2]

[1] Wrocław University of Science and Technology, Wrocław, Poland
anna.kilyk@pwr.wroc.pl
[2] Państwowa Wyższa Szkoła Zawodowa Nysa, Nysa, Poland
zofia.wilimowska@pwr.wroc.pl

Abstract. There are many methods, which can be used to analyze risk on the capital market. This paper describes several approaches to risk analysis and then attempts to create a risk prediction model. In the conclusion one can see that it's possible to minimize the investment risk by using Hurst exponent.

Keywords: Hurst exponent · Trend · Returns

1 Introduction

For several years economists worked intensively with risk and ways of managing it. However, their interests aren't distributed uniformly, that's why the methods which they created mostly can be used only to analyze specific cases. A good example of this situation is the document "Standard Risk Management" created by Ferna (Federation of European Risk Management Association) [1]. The paper's topic indicates a rather general approach but one can apply it only to risk management in organization.

A completely different situation can be observed in finance management, where risk management skills are important but often underrated elements. That underestimation happens for small investors and big corporations alike. This situation can be seen clearly while investing on the stock exchange, where two groups of methods are used. The first one, which won't be described in this paper, is a subjective set of methods called qualitative methods. They are based on the conclusions, which an investor can have from (for example) an analysis of the political and economical background of a set of markets.

The second one, which is based only on the analysis of historical data, is the set of quantitative methods. Using these methods investors analyze available data in some specific ways, for example: calculating standard statistical quantities (standard deviation, variation, subvariation, etc.), using technical analysis (trends) or applying one of the VaR (Value at Risk) methods.

1.1 Technical Analysis - Trend Analysis

The first way of investigating risk is the technical analysis, which is often used to analyze prices or returns at stock exchange by the short term investors. One of the

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_32

reasons for choosing it, is that it allows an easy interpretation of the final solution – e.g. plots of the trend. Apart from that, there is a lot of websites, which offer already prepared plots or create new one for a given time period on the fly, and give the most important information for these plots.

The main element of this model is the trend. It clearly shows the direction of changes in analyzed data series. To calculate it one can use many different functions (linear function, exponential function, polynomials, etc.) and look for the best fit. The easiest function amongst them is the line function which is described by the Eq. (1):

$$y = at + b \tag{1}$$

where:

t - time step,

y - analyzed parameter (price, return),

a, b - fit parameter (sign of "a" indicates the trend direction).

There are two possible ways to use this method. The first one tries to analyze trends by looking for characteristic behavior of the index. The second analyses two type of trends simultaneously: short (for 5 days) and long (for 21 days). Thanks to this one can determine two kinds of point: death (when short trend crossed long from the top - and it is an indicator to sell assets) and gold (when long trend crossed short from the top - and it says to buy assets).

To summarize, the trend analysis can be a useful tool. However, these methods are unable to detect small changes, which are related to price fluctuation. This lack of sensitivity is associated with the fact that trends are a way of indicating the mean of prices. That's why some fluctuations are extinguished by different fluctuations of a different sign.

1.2 VaR (Value at Risk) [2]

Value at risk is a method, which calculates the amount which can be lost during an investment, with some assumed initial conditions. Initial parameters include total investment time and the investor's confidence level [8].

The easiest method from the VaR group of methods is called variation–covariance approach. To assess potential losses and gains one doesn't have to use a specific initial investment. That's why using Eq. (2) one can consider possible percentage changes [6].

$$VaR_N = k\delta_N - \mu_N \tag{2}$$

where:

δ_N - standard deviation,

μ_N - mean,

k - constant depended of confidence level.

The biggest disadvantage of this method is the assumption that returns have normal distribution. This assumption is a big generalization, which can distort the solution. During the analysis of financial markets it can be seen, that the distribution of returns is

similar to the normal distribution. However, the distribution plot has characteristic "fat tails", which are typical for distribution with positive kurtosis.

1.3 Hurst Exponent

The last of the described approaches is DMA (detrending moving average) [7], from which one can obtain an important parameter: the Hurst exponent. The basic definition of this parameter says that it can show a probable change of direction [4]. The easiest way to calculate it is shown in Eq. (3):

$$D = 2 - H \tag{3}$$

where:

D - fractal dimension,

H - Hurst exponent.

The most important information, which can be obtained from this parameter (H) is its position in relation to point 0.5. Obtained value of the Hurst exponent allows to put it in two groups of solutions (Eq. 4) [5]:

$$H \in \begin{cases} 0 \leq H \leq 0.5 \\ 0.5 \leq H \leq 1 \end{cases} \tag{4}$$

The first one indicates an antipersistent signal, which means that there is higher probability that in next step the price will be in the opposite direction then in the same direction. On the other hand, the second group – persistent signal – indicates that the opposite situation is more likely.

This way of risk analysis is a new approach and came from fractal geometry. Although this parameter can be helpful in risk management, it's application introduces a lot of problems and is quite difficult.

1.4 Detrended Moving Average (DMA)

One of the methods, which can be used to calculate Hurst exponent is Detrended Moving Average (DMA) [3]. This method has three steps. The first step is to calculate a moving average:

$$\check{y}_n(i) = \frac{1}{n}\sum_{k=0}^{N} y(i-k) \tag{5}$$

where:

$\check{y}_n(i)$ - moving average of length n,

$y(i)$ - time series,

n - length of moving average,

N - length of time series.

In the next step (using the results from Eq. 2) a standard deviation is calculated:

$$\delta_{DMA}(n) = \sqrt{\frac{1}{N-n}\sum_{i=n}^{N}[y(i) - \check{y}_n(i)]^2} \tag{6}$$

where:

$\delta_{DMA}(n)$ - modified standard deviation.

The last step is to find relation between standard deviation (calculated by Eq. 6) and the value of the Hurst exponent (H):

$$\delta_{DMA}(n) \sim n^H \tag{7}$$

2 Construction of Proposed Model

This section will explain each step of the model creation. Before that, some information about the analyzed data is needed. This paper is focusing on analyzing returns from the WIG20 index which is noted on Warsaw Stock Exchange.

The first step, of the presented model, is to create groups of returns. Returns are split into 11 sections of equal width, which can be easily calculated as the difference between maximal and minimal value of the returns divided by number of the sections (Eq. 8):

$$W = \frac{r_{max} - r_{min}}{11} \tag{8}$$

where:

W - section width,

r_{max} - maximum value of returns,

r_{min} - minimum value of returns.

When all data will be assigned to appropriate sections it will be possible to create the probability of finding a point from each group. To count it (Eq. 9) one need only two numbers: how many points are in all section and how many points are in the current section:

$$P = \frac{d}{g} \tag{9}$$

where:

P - probability,

d - number of points in one section,

g - number of points in all section.

This probability distribution will then be used to mimic historical returns while generating new points of the artificial series.

During this research one can analyze eight combinations of the H value and trend (four for each of the two possible trends). Same approach is used for both positive and negative trends.

- $H \leq 0.25$ - it means that the trend is weak (prices have large fluctuation) and there is very high probability that the next step will bring a change in the current trend.
- $0.25 < H \leq 0.50$ - here the fluctuation of prices are weaker (the closer to 0.5 the weaker), the trend is still weak but less likely to changes in next step.
- $0.50 < H \leq 0.75$ - here the trend start to be stronger, hence there is a chance that in the next step the trend will be the same.
- $H > 0.75$ - the price fluctuations are weak (for $H = 1$ there aren't any fluctuation) that means the trend is strong and it will probably be the same in next steps.

This approach is aimed at a better replication of real data since division into only persistent and antipersistent series would mean ignoring the effects of extremely high and extremely low H values.

In Table 1 one can find all possible outcomes of the distribution modifications based on the H value and trend direction. To understand its meaning let's take an example of growing trend and $H > 0.25$. In this case the price has a very weak growing trend (due to the low value of H), knowing that it's quite probable that in next step the trend will change to a negative one. Of course in second case $(0.25 < H \leq 0.50)$ the probability of trend change is weaker but still greater than the probability of maintaining the trend.

Table 1. Possible modifications of the distribution fluctuation due to H value and trend direction. δ is the standard deviation of the analyzed series.

	$0 < H \leq 0.25$	$0.25 < H \leq 0.5$	$0.5 < H \leq 0.75$	$0.75 < H \leq 1$
Positive trend	-3δ	-2δ	2δ	3δ
Negative trend	3δ	2δ	-2δ	-3δ

Knowing the influence of the H exponent on the trend, one should modify the possible price changes accordingly (Table 1). Using data from the table, one creates four different probability distributions. For example, to modify the distributions for the '-3δ' (first case), probability of selecting a section with negative returns will be multiplied by 3δ while the other section will stay the same. For the opposite situation: growing trend with H$(0.75 < H \leq 1)$, the probability of selecting a positive section should be increased by 3δ .

The final stage of preparation is to generate a random number from the modified distribution function that mimics the current situation. After adding that new point to the available data (and removing a point from the beginning to maintain a constant length) the calculations are repeated and a new point can be generated. Therefore, to obtain predictions for 10 days the whole process should be repeated 10 times.

3 Results Obtained from Proposed Model

To have a good starting point 200 returns from the WIG20 index were analyzed (period between 28.02.2011 and 13.12.2011).

This period (Fig. 1) is characterized by a noticeable fall of price, which has been particularly visible in the vicinity of trading day number 109 (early August 2011).

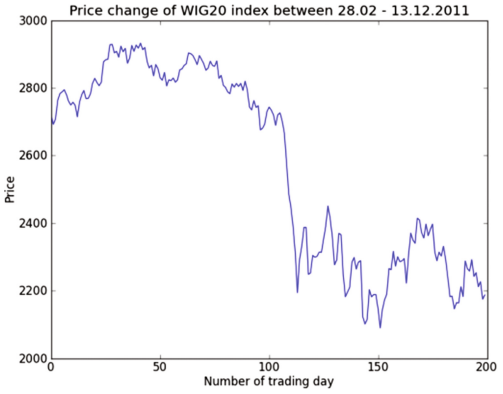


Fig. 1. Plot of the WIG20 index for the time period between 28.02.2011 and 13.12.2011.

Analyzing the values obtained from the calculations for the initial data set (Table 2) one can note that the sample period is characterized by:

Table 2. Statistical results for the real daily returns (calculated for the time interval from 28.02.2011 to 13.12.2011).

	Value
Return	−0.172
Trend	−0.011
Hurst exponent	0.510

- Decline in price, indicated by the negative value of the trend and return,
- And big probability of future price changes in next steps – as shown by low value of Hurst exponent.

Created histogram for the analyzed returns (Fig. 2) shows that the most of the points are in the range between −0.018 and 0.015 (about 77% of all data). Moreover, it's worth noting that the range of negative values is bigger, which also indicates a downward trend.

It is clearly visible for Fig. 3 that at the beginning of the period the fluctuations were small and starting from about 109 trading day they started to strongly fluctuate. Combining this information with the obtained values of the calculated parameters (Table 2) it can be assumed that created model will continue the downward trend.

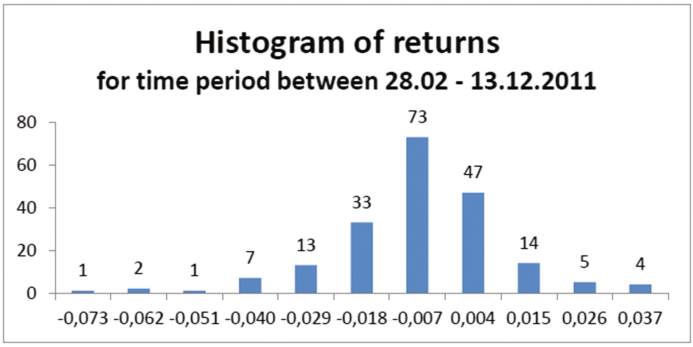


Fig. 2. Histogram of the WIG20 index daily returns for the time period between 28.02.2011 and 13.12.2011.

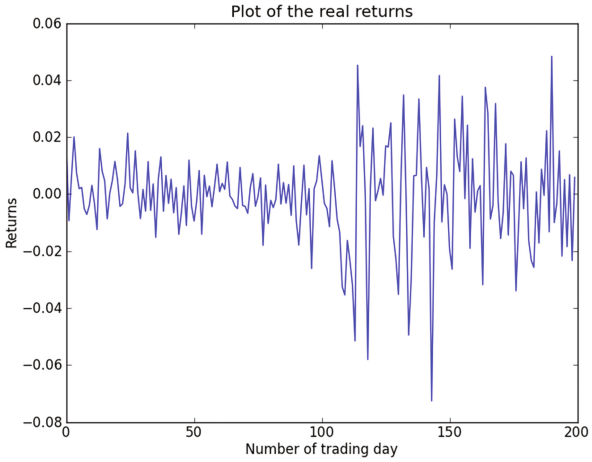


Fig. 3. Plot of real daily returns for WIG20 for the time period between 28.02.2011 and 13.12.2011.

In the next step, using the original data, one will create a simulation of the future returns. As mentioned earlier, each added data point automatically discards the oldest point from the original data to maintain a constant length.

Analyzing the obtained results (Fig. 4) it can be noted that the beginning of the forecasted returns (approximately 150–160 steps) is fairly stable. However, after this, comes a short period (about 10 steps) where the daily returns start to change. That decline in returns is the reason for the significant negative global return. Moreover it's noteworthy that the values of analyzed parameters (trend, Hurst exponent, average returns – Table 3) are growing and after the simulation they are significantly better (from the investors point of view) then the original ones.

Figure 4 contains last 150 points of real data and artificially generated data. One can compare it with an analogous plot done for real data (previous 150 points with the

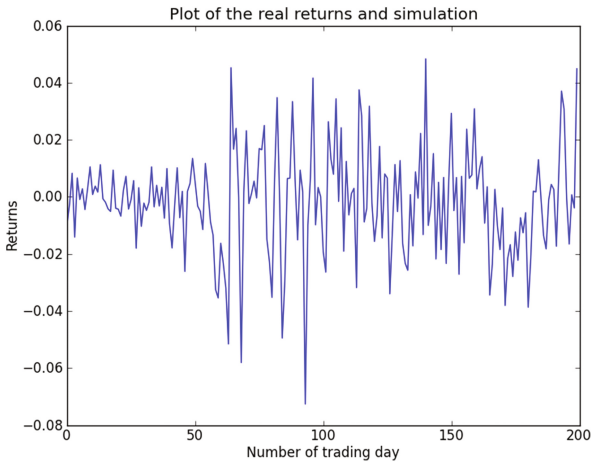


Fig. 4. Plot of daily returns for the time period between 12.05.2011 and 24.04.2012 (with added 50 artificial (generated) points).

Table 3. Statistical results of the WIG20 index simulation for the time period between 12.05.2011 and 24.04.2012.

	Value
Return	−0.154
Trend	−0.004
Hurst exponent	0.596

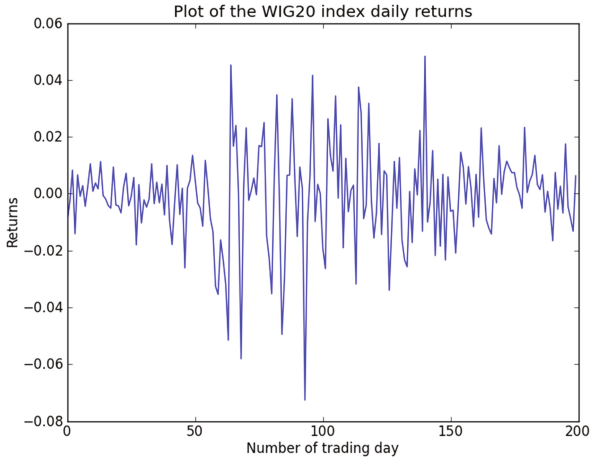


Fig. 5. Plot of the WIG20 index daily returns for time period between 12.05.2011 and 24.04.2012.

addition of next 50 points of real data). This "hybrid series" was also analyzed in the same way as the original data (statistical measures are available in Table 3) and shows similar values of return and trend to those from the original series.

Comparing Figs. 4 and 5 one can notice a similar visual appearance, although the extreme values are more visible on the simulated data. By comparing values of the real return (0.008) with the simulated one (−0.154) one can see that the artificial data mimics the history well, but is significantly differs from the real values.

4 Conclusion

The proposed model still has some drawbacks as it repeats historical data too accurately. The rather poor pre-diction of future returns, might be a result of a too global approach. Focusing on replicating the historical data in a shorter time span might give more accurate results. Future work will go in this direction.

References

1. http://www.ferma.eu/
2. Best, P.: Value at Risk, calculation and implementation of the VaR model, Oficyna Ekonomiczna (2000)
3. Bunde, A., Havlin, S., Kantelhardt, J.W., Penzel, T., Peter, J., Vooigt, K.: Correlated and uncorrelated regions in heart-rate fluctuations during sleep. Phys. Rev. Lett. **85**, 3736 (2000)
4. Czarnecki, Ł., Grech, D., Pamuła, G.: Comparison study of global and local approaches describing critical phenomena on the Polish stock exchange market. Phys. A Stat. Mech. Appl. **387**, 6801–6811 (2008)
5. Karpo, K., Orłowski, A.J., Łukasiewicz, P.: Stock indices for emerging markets. Acta Phys. Pol. Ser. A Gen. Phys. **117**, 619 (2010)
6. Kiłyk, A.M., Wilimowska, Z.: VaR dynamics for investment portfolios. In: ISAT 2013 (2013)
7. Kiłyk, A.M., Wilimowska, Z.: Application of Hurst exponent on the stock exchange. Naukowa Szkoła Letnia, Jachranka 05 (2012)
8. Stambaugh, F.: Risk and value-at-risk. Eur. Manag. J. **14**(6), 612 (1996)

Modeling the Customer's Contextual Expectations Based on Latent Semantic Analysis Algorithms

Nina Rizun[1(✉)], Katarzyna Ossowska[1], and Yurii Taranenko[2]

[1] Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland
{nina.rizun, katarzyna.ossowska}@zie.pg.gda.pl
[2] Alfred Nobel University, Naberezhna Lenina Street, 18,
Dnipro 49000, Ukraine
taranen@rambler.ru

Abstract. This paper presents the approach of modeling the system of customer's contextual expectations. The novelty consists in systematic usage of the different technic. The first technic – the theory of Benefits Language as an instrument of forming Concept of Benefits for studied product. The second one – the combination of advantages the probabilistic Latent Dirichlet Allocation (LDA) and Linear Algebra based Latent Semantic Analysis (LSA) methods as an instrument of textual data retrieval. The verification of the proposed approach for specifies type of product – films – was conducted. The main research plan was realized: Contextual Summary using the LDA-based algorithm was formed; the Contextual Frameworks using LSA-based approach were performed; the Manually Created Contextual Expectations Dictionary was built. The results of case study, based on the Polish-language film reviews corpora analysis, allowed to make the conclusions about the possibility to use proposed approach for building the system of Customer's Contextual Expectations.

Keywords: Benefits Language · Latent Semantic Analysis · Latent Dirichlet Allocation · Customer · Contextual expectations

1 Introduction

Nowadays, in the age of Internet, access to open data detects the huge possibilities for information retrieval. More and more often we hear about the concept of open data which is unrestricted access, in addition to reuse and analysis by external institutions, organizations and people. It's such information that can be freely processed, add another data (so-called remix) and then published. More and more data are available in text format (such as reviews on books, movies, etc.). Algorithms of Latent Semantic Relations Analysis are one of the important tools for extraction and recognition of significant facts from textual data sets. Another aspect of research is to find ways and means of using the information obtained by Semantic tools applying in order to maximize the expected benefit. In this area, one of the modern tools for formulating the concept of benefits is the Benefits Language [1–5]. One of the conditions for the formation of the concept of benefits for studied product is the collection and processing

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8_33

of information about the client's expectations. This process often requires additional time and financial costs. Therefore, one of the ways to obtain the "maximum benefit" of using the Benefits language is to develop a methodology of building the system of customer's contextual expectations via systematic usage of benefit's theory and algorithms of semantic analysis of textual opinions of open Internet pages'.

2 Related Research

Benefits language is a very well-known technique in door-to-door selling. It's a technique that present product in a way that client would like to buy it. Benefits language is based on F-A-B rule (Feature-Advantage-Benefit). It means, that first are shown features of a product (colour, material, shape etc.) then advantages which are result of those features and in the end – benefits. Benefits are profits which the customers will reach when they will buy and then will be using the product [1, 2].

The purpose of the Semantic Relationships Analysis is to extract "semantic structure" of the collection of information flow and automatically expands term into the underlying topic. Significant progress on the problem of presenting and analysing the data have been made by researchers in the field of information retrieval (IR) [6–8]. The basic methodology proposed by IR researchers for text collection – reduces each document in the corpus to a vector of real numbers, each of which represents ratios of counts.

The vector model [9–13] of text representation is one of the first methods used to solve latent semantic relations revealing the topic modeling problems. Initially, this model was used in topic detection tasks by extracting events from the information flow [10, 14]. The representation of the corpora in this case realized with the help of vectors models form, in which each word is weighted according to the chosen weight function [15, 16]. For solving the problem of finding the similarity of documents (terms) from the point of view of the relation to the same topic. The most appropriate metric is cosine measure of the edge between the vectors [9, 10].

Latent Semantic Analysis (LSA) is a theory and method for extracting context-dependent word meanings by statistical processing of large sets of text data [18]. The LSA also works with a vector representation of the "bag of words" text units. The text corpora are represented as a numeric matrix Word-Document, the rows of which correspond to words, and columns to text units – documents.

The revealing the Latent Semantic Relationships via LSA between words/documents usually applying the singular decomposition [15, 16]. According to the theorem on singular decomposition (SVD), any real rectangular matrix can be decomposed into a product of three matrices [9–11].

The significant *shortcomings* of LSA method are: probabilities for each topic and the document distributed uniformly, which does not correspond to the actual characteristics of the collections of documents [13, 15, 16]; increasing the size of the analyzed documents significantly reduces the quality of recognition of hidden relations [14, 17].

Another group Latent Semantic Relationships studying is the Probabilistic thematic modeling. It is a set of algorithms that allow analyzing words in textual corpuses and extract from them topics, links between topics [13–19]. Latent Dirichlet Allocation

(LDA) is a generative model that explains the results of observations using implicit groups, which allows one to explain why some parts of the data are similar. It was proposed by David Blei [20, 21].

The main *drawback* of the LDA is the lack of convincing linguistic justifications. From the point of view of texts, the assumption of the Dirichlet distribution is not justified. Additionally, in the process of the assigning the topics to documents usually LDA use the maximal form possible (not always very high) level of probability of a documents belonging to the topic.

The main *purpose* of this paper to contribute the new methodological approach to the theory of modeling the system of customer's contextual expectations via systematic usage of:

- technics of Benefits Language as an instrument of forming Concept of Benefits for studied product,
- advantages of the algorithms of Latent Semantic Relations Analysis of textual customer's opinions of open Internet pages' as a data retrieval instrument for building the Concept of Benefits.

For demonstration the of basic workability of the author's approach realization as a sample for *case study* the Polish-language film reviews corpora (FRCS) from the filmweb.pl is used.

3 Methodology

In this paper the following author's definitions will be used:

1. *Corpora* (films reviews corpora sample, FRCS) is a collection of the textual Documents.
2. *Term* is a word after preprocessing.
3. *Latent Semantic/Probabilistic topics* is a basic unit of Latent Semantic Relations, received by *LSA/LDA* approach.
4. *Subjectively Positive* (*CFSP*) and *Subjectively Negative (CFSN) Corpora Samples* is a result of the classification of the FRCS on the basis of information on the subjective assessment of films by the reviewers (measured by 10-point scale). For this purpose the following heuristic was adopted: to consider the CFSP, if the subjective review's assessment is more than 7 points, and CFSN – if it is equal or less 4 points.
5. *Contextual Summary* (CS) is a set of Latent Probabilistic Topics (LPT), described the main context of the *Corpora.*
6. *Contextual Framework* (CF) is a set of main Latent Semantic Topics (LST) with keywords, considered in the *Contextual Summary.*
7. *Contextual Bigram* (CB) is a combination of keyword (noun) and its contextually close term (adjective), which describe the particular topic.
8. *Contextual Expectations Dictionary* (CED) is a set of CB with the numeric indicators of the level of their positivity/negativity with respect to particular topic.
9. *Hierarchical Semantic Corpora* (HSC) is a structure of the clustered paragraphs of the FRCS, which relate to a particular Topic form CF.

3.1 Novelty and Research Plan

With *aim* to develop the system of customer's contextual expectations the following scientific research question was raised: *Is it possible to build the system of customer's contextual expectations* via *systematic usage of Benefit's Theory and Algorithms of Semantic Analysis of Textual Opinions?*

For finding the answers for this question the following main concepts were formulated:

Concept 1. Taking into account the specificity of chosen case study and due to the peculiarity of the requirements of film's review writing [11], each paragraph of such document could be identified as an indivisible contextual unit, which characterized by a particular Latent topic and should be analyzed separately.

Concept 2. The quality of Contextual Frameworks for the *CFSP* and *CFSN* building, could be improved due to the systematic usage: LDA-method recognizing of the Latent Probabilistic Topics within the Samples; LSA-method of Latent Semantic Relations recognition within the set of Latent topics.

Concept 3: The system of customer's contextual expectations can be interpreted as a CED after transformation using the special expectation's labels "*Should have*, "*Could have*" and "*Won't have*".

On the bases of this concepts, the following research plan was developed: *to demonstrate the possibility of building the system of customer's contextual expectations via the realizing the following steps*:

1. Formation of the *Contextual Summary* for CFSP and CFSN using the LDA-based algorithm and Paragraph-oriented Approach Concept.
2. Building the *Contextual Frameworks* for CFSP and CFSN using LSA-based approach.
3. Creating a *HSC* on the bases of Contextual Frameworks for CFSP and CFSN.
4. Manually Creating the *CED*.
5. Formation the system of *Customer's Contextual Expectations* using the Benefits Language concept.

The size of *case study* sample for research plan realization is 3000 films reviews (1500 *Subjectively Positive* and 1500 *Subjectively Negative*). All texts were presented in .txt format files and processed using the Python-based programing tools.

3.2 Latent Semantic Analysis Results

Step 1. Formation of the List of the Latent Probabilistic Topics for CFSP and CFSN using the LDA-based algorithm and Paragraph-oriented Approach Concept

This step presupposes the realizing the following stages: text preprocessing [11]; text preparation; LDA-modeling.

The stage of *LDA-modeling* gives the possibility to receive the *Contextual Summary* of Latent Probabilistic Topics with information about most probable (significant) words and assign these topics with maximal probability to particular paragraphs. For obtaining the optimal combination – Number of topics/terms in topic main – the values

of Perplexity [19] were analyzed. The optimum value of the Perplexity achieved in the point, when further changes in the parameters do not lead to its significant decrease.

The structure of the *Contextual Summary,* as a *CS results*, is presented in Table 1. The recall rate as the ratio of the number of topically recognized paragraphs (probability of belonging the paragraph of topic >0.7) to the total number of paragraphs is within 90–95%.

Table 1. The structure of the contextual summary

CFSP		CFSN	
Number of paragraphs	9900	Number of paragraphs	10650
Number of topics	26730	Number of topics in paragraph	33015
Average number of topics in paragraph	2.7	Average number of topics in paragraph	3.1
Average number of terms in topic	6.8	Average number of terms in topic	7.3
Average perplexity value	102	Average perplexity value	99

Step 2. Building the CF for CFSP and CFSN using LSA-based approach

For recognition of Latent Semantic Relations within the *CS* and recognition *CF* via semantic clustering of the CS elements, the following stages are presupposed: text preprocessing; creating the Term-Document Matrix; SVD process; identifying the hidden semantic connection within the CS; LSA clustering of CS elements/terms in the semantic dimension [10].

After realization of first three stages, based on the matrices of cosine distances between the vectors of CS elements (topics) and terms, the LSA clustering step have been realized. As a method the k-means clustering [12, 21] had been chosen. For finding the optimal number of clusters, the following *combination* of two factors was used [9]: the degree of the boundaries fuzziness between the clusters; the closeness of the similarity measures values within a cluster of the CS elements/terms similarity within the cluster

The optimal combination for these two parameters, as a *CS results*, was reached for 5 clusters for *CFSP* and for 4 clusters for *CFSN*. On the bases of this information, for each cluster of both CS the *Contextual Frameworks* were obtained (Tables 2 and 3). For this: the list of keywords was identified (KW); the weights for keywords were formed (W); the Contextual Labels of each of the CF were specified.

Table 2. Contextual frameworks for CFSP

Hero		Director		Script		Plot		Spectator	
KW	W	KW	W	KW	W	KW	W	KW	W
Hero	1.0	Director	1.0	Script	1.0	Plot	1.0	Spectator	1.0
Playing	0.8	Creator	0.8	History	0.8	Hero	0.8	Fan	0.6
Person	0.6	Stage	0.6	Writer	0.6	Action	0.6	Watch	0.8
Actor	0.4	Drama	0.4	Picture	0.4	Film	0.4	Interest	0.4
Main	0.2	Effect	0.2	Layer	0.2	History	0.2	Film	0.2

Table 3. Contextual frameworks for CFSN

Hero		Actor		Creator		Plot	
KW	W	KW	W	KW	W	KW	W
Hero	1.0	Actor	1.0	Writer	1.0	Plot	1.0
Spectator	0.8	Character	0.8	Director	0.8	History	0.8
Climate	0.6	Picture	0.6	Film	0.01	Stage	0.6
Person	0.4	Role	0.4	Spectator	0.6	Script	0.4
Fan	0.2	History	0.2	Action	0.2	Scenarist	0.2

Step 3. Creating a HSC on the bases of Contextual Frameworks for CFSP and CFSN

After receiving the Contextual Frameworks for CFSP and CFSN the basic algorithm for *HSC* building aw developed. The main idea of this algorithm – the process of paragraph selection, semantically close to particular CF (as a set of keywords).

Step 4. Manually Creating of Film Reviews' Corpora-based Contextual Expectations Dictionary

The algorithm of the Manually Creating CED presupposed the retaliation of the following stages:

1. Forming a set CB (is implemented using Python NLTK functions `nltk.bigrams(…)`).
2. Estimation of weights of CB. Assumes the definition of the absolute weight of bigrams, estimated by the frequency of occurrence of this bigram on the CFSP and CFSN.
3. Correction of Weights of CB using Relevance Frequency. In this paper, in addition to the frequency of use of the C of positive or negative tonality, the parameter is used to reverse the frequency of using this term in negative or positive utterances – RF (Relevance Frequency) [24]. The basic meaning of the RF measure is that the weight of the word is calculated on the basis of information about the distribution of this bigrams in the texts of the collection and takes into account the belonging of the collection texts to certain classes (positive, negative):

$$RF_S = \log_2\left(2 + \frac{a}{\max(1, b)}\right) \tag{1}$$

   where a – the number of paragraphs related to category S (CFSP or CFSN) and containing this bigram; b – the number of paragraphs not related to category S and containing this bigram as well.
4. Forming the rules for Polarity Scores building: formulating the Polarity limits for scaling the Weights and there interpreting as an expectation's labels "*Should have*", "*Could have*" and "*Won't have*".
5. Analysis of the structure of the received elements of the CED for the positive and negative parts of the Corpora.
6. Based on this algorithm, as an CS result, the *Corpora-based Hierarchical* CED was obtained. Generally, 797 CB for CFSP and 429 for CFSN were formed.

3.3 Defining the System of Contextual Expectations

Corpora-based Hierarchical CED is a main source for realizing the step of Contextual Expectation System formulation. As identified in step 4, CED contains the positive, negative and neutral features of opinions about specific product "films" and concerns a separate topic. To create a demo example, the authors modeled the situation of having information on only four (identical) topics for each (CFSP and CFSN) group of CED: *Hero*, *Actor*, *Director* and *Script*. The algorithm of Contextual Expectation System formulation presented in Fig. 1.

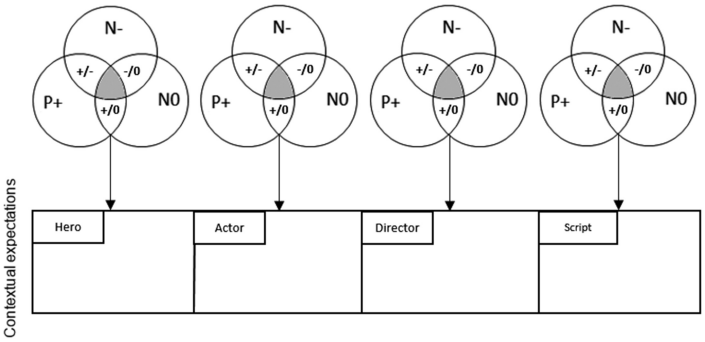


Fig. 1. Algorithm of contextual customer's expectation system forming

Legend: **P +** – positive, **N−** – negative; **N0** – neutral; **±** – positive and negative; **−/0** – negative and neutral; **+/0** – positive and neutral; ▒ – positive and negative and neutral

For demonstration this algorithm realization, the data of CED for keyword "*Hero*" was used (Tables 4 and 5). The process of the Contextual Expectation System builds on the following concept: Based on MoSCoW technique [1–4], for prioritization customer requirements, authors grouped features in three categories: "*Should have*" – represents a high priority item that should be included in the solution, if possible, "*Could have*" – describes a requirement that is perceived as desirable, but not necessarily (it will be included if time and resources allow), "*Won't have*" – represents a requirement that, with the consent of stakeholders, will not be implemented in a given release.

Features from "*Should have*" group are those which were indicated as positive, that why movie should have hero with this features to increase number of positive opinions. "*Could h*ave" features are neutral one. In authors opinion, those are the feature than movie can have but is not necessary and it would not have influence on opinion while negative will have bad influence and those features should not occur. In our *CS* example, we didn't have features that was, at the same time, positive and negative, positive and neutral, negative and neutral, positive and negative and neutral that's why common space between those kind of features is empty (Fig. 2).

Table 4. The fragment of polarity of hero's positive' features

Birgams	Polarity scores	Polarity
Hero amazing	1.00	Positive
Hero beautiful	0.83	Positive
…		
Hero solid	0.33	Neutral
Hero wakes	0.33	Neutral
…		
It is not a hero	−1.00	Negative

Table 5. The fragment of polarity of hero's negative' features

Birgams	Polarity scores	Polarity
Lack of distinctive	0.83	Negative
Is not valid	0.83	Negative
…		
Angry hero	0.33	Neutral
Demonic hero	0.17	Neutral
…		
Beautiful hero	−1.00	Positive

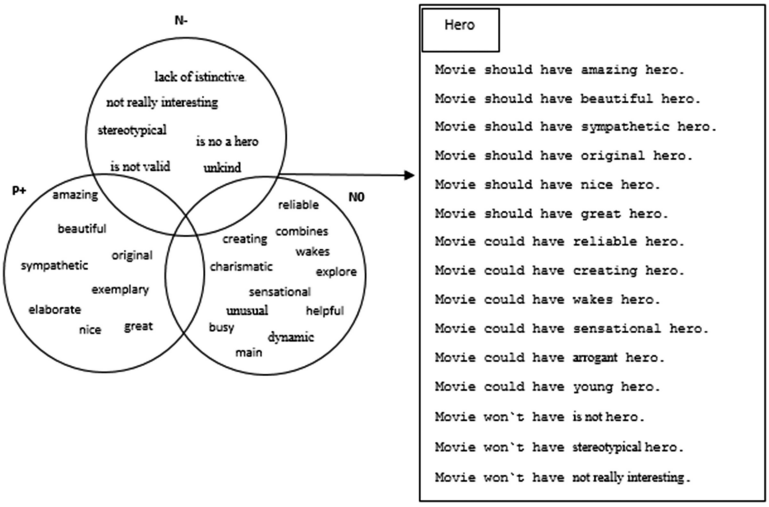


Fig. 2. The example of the algorithm of contextual expectation system forming realization

The Last step of Contextual Expectation System Forming – creating advantages and benefits from features. According to the benefits language and rule F-A-B first we create advantages which are results of features, then, in the end, benefits. In the Table 8

we propose the example of Features for Benefit: "*The spectator is more likely to watch the movie*". But for real situation the benefits should be different (Table 6).

Table 6. The example of features-advantages-benefits about "Hero"

Hero	
Feature	Advantage
Amazing	Arouses admiration
Beautiful	Slightly
Sympathetic	It arouses positive feelings
Original	Arouses curiosity
Elaborate	Professional approach to the viewer
Nice	It arouses positive feelings
Great	Arouses admiration

4 Conclusions and Future Research

In this paper authors contribute the new scientific approach for the system of customer's contextual expectations building. Verifying the approach on the case study of Polish-language Film Reviews Corpus analyzing helps to authors to find the answer to the main research question: the systematic usage of Benefit's Theory and combination of the Algorithms of Semantic Analysis of Textual Opinions is the real scientific instrument of building the system of customer's contextual expectations on the bases of textual opinions of open Internet pages'.

Prospects for future research in terms of the use of Semantic Analysis tools consist in the possibility of using the Concept of Benefits for studied product to create a Methodology for: measuring the degree of positivity/negativity of the user's opinions on the basis of an assessment of the measure of closeness of their opinions to ideal; recognizing the structure of the Customer's expectations expressions formulations.

Acknowledgments. The research results, presented in the paper, are supported by the Polish National Centre for Research and Development (NCBiR) under Grant No. PBS3/B3/35/2015, the project "Structuring and classification of Internet contents with the prediction of its dynamics".

References

1. Ossowska, K., Szewc, L., Weichbroth, P., Garnik, I., Sikorski, M.: Exploring ontological approach for user requirements elicitation in design of online virtual agents. In: Wrycza, S. (ed.) Information Systems: Development, Research, Applications, Education, pp. 40–55. Springer, Cham (2016)
2. Ossowska. K., Szewc. L., Orłowski. C.: The principles of model building concepts which are applied to the design patterns for smart cities, intelligent information and database systems (2017). doi:10.1007/978-3-319-54430-4

3. Ossowska, K.: Design modern technologies for older people by using expert systems containing benefits language, Zeszyty Naukowe Politechniki Poznańskiej, Organizacja i Zarządzanie (2017)
4. Ossowska, K., Orłowski, C.: Designing IT systems using benefits language, design and implementation of management information systems. Selected aspects (2016)
5. Ossowska, K., Czaja, A.: Model of personalization website using benefits language (2017)
6. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, 2nd edn. Addison-Wesley, Wokingham (2011)
7. Furnas, G.W., Deerwester, S., Dumais, S.T., Landauer, T.K., Harshman, R.A., Streeter, L.A., Lochbaum, K.E.: Information retrieval using a singular value decomposition model of latent semantic structure. In: Proceedings of ACM SIGIR Conference, pp. 465–480. ACM, New York (1998)
8. Salton, G., Michael, J.: McGill Introduction to Modern Information Retrieval. McGraw-Hill Computer Science Series, vol. XV, p. 448. McGraw-Hill, New York (1983)
9. Rizun, N., Kapłanski, P., Taranenko, Y.: Development and research of the text messages semantic clustering methodology. In: 2016 Third European Network Intelligence Conference, # 33, pp. 180–187. ENIC (2016)
10. Rizun, N., Kapłanski, P., Taranenko, Y.: Method of a two-level text-meaning similarity approximation of the customers' opinions. Economic Studies – Scientific Papers. University of Economics in Katowice, Nr. 296/2016, pp. 64–85 (2016)
11. Rizun, N., Taranenko, Y.: Development of the algorithm of polish language film reviews preprocessing. In: Proceeding of the 2nd International Conference on Information Technologies in Management. Rocznik Naukowy Wydziału Zarządzania WSM (2017, in print). http://www.wsmciechanow.edu.pl/rocznik-naukowy/
12. Kapłanski, P., Rizun, N., Taranenko, Y., Seganti, A.: Text-mining similarity approximation operators for opinion mining in BI tools. In: Proceeding of the 11th Scientific Conference "Internet in the Information Society-2016", pp. 121–141. University of Dąbrowa Górnicza (2016)
13. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM. **18**(11), 613–620 (1975)
14. Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S.: Using latent semantic analysis to improve information retrieval. In: Proceedings of CHI 1988: Conference on Human Factors in Computing, pp. 281–285. ACM, New York (1988)
15. Deerwester, S., Dumais, S.T., Harshman, R.: Indexing by Latent Semantic Analysis (1990). http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf
16. Eden, L.: Matrix Methods in Data Mining and Pattern Recognition. SIAM, New Delhi (2007)
17. Aggarwal, C., Zhai, C.X. (eds.): Mining Text Data. Springer, New York (2012)
18. Bahl, L., Baker, J., Jelinek, E., Mercer, R.: Perplexity – a measure of the difficulty of speech recognition tasks. In: Program, 94th Meeting of the Acoustical Society of America, vol. 62, p. S63 (1977)
19. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
20. Blei, D.M.: Introduction to probabilistic topic models. Commun. ACM. **55**(4), 77–84 (2012)
21. Daud, A., Li, J., Zhou, L., Muhammad, F.: Knowledge discovery through directed probabilistic topic models: a survey. In: Proceedings of Frontiers of Computer Science in China, pp. 280–301 (2010)
22. Blei, D.M.: Topic modeling. http://www.cs.princeton.edu/∼blei/topicmodeling.html

Author Index

J. Świątek et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*, Advances in Intelligent Systems and Computing 656, DOI 10.1007/978-3-319-67229-8