Vadim Ermolayev · Heinrich C. Mayr
Mykola Nikitchenko · Aleksander Spivakovsky
Grygoriy Zholtkevych (Eds.)

Communications in Computer and Information Science 469

Information and Communication Technologies in Education, Research, and Industrial Applications

10th International Conference, ICTERI 2014
Kherson, Ukraine, June 9–12, 2014
Revised Selected Papers

Springer

Communications in Computer and Information Science 469

More information about this series at http://www.springer.com/series/7899

Vadim Ermolayev · Heinrich C. Mayr
Mykola Nikitchenko · Aleksander Spivakovsky
Grygoriy Zholtkevych (Eds.)

Information and Communication Technologies in Education, Research, and Industrial Applications

10th International Conference, ICTERI 2014
Kherson, Ukraine, June 9–12, 2014
Revised Selected Papers

Springer

Editors
Vadim Ermolayev
Zaporizhzhya National University
Zaporizhzhya
Ukraine

Heinrich C. Mayr
Institute of Applied Informatics
Alpen-Adria-Universität Klagenfurt
Klagenfurt
Austria

Mykola Nikitchenko
Taras Shevchenko National University of Kyiv
Kyiv
Ukraine

Aleksander Spivakovsky
Kherson State University
Kherson
Ukraine

Grygoriy Zholtkevych
V.N. Karazin Kharkiv National University
Kharkiv
Ukraine

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-319-13205-1 ISBN 978-3-319-13206-8 (eBook)
DOI 10.1007/978-3-319-13206-8

Library of Congress Control Number: 2014953945

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

It is our pleasure to present the proceedings of ICTERI 2014, the 10th International Conference on Information and Communication Technologies (ICT) in Education, Research, and Industrial Applications: Integration, Harmonization, and Knowledge Transfer.

The conference was planned to be held at Kherson, Ukraine, during June 9–12, 2014. But unfortunately, due to the tense situation in the country,[1] we had to cancel the face-to-face gathering of the conference participants. Based on the results of the review process, however, we were convinced that some of the submissions deserved to be published, and are of interest to the scientific community. We therefore decided to invite the authors of the best papers, to revise and extend these and to submit them for publication in this volume.

Thus, the selection was made in two phases, as the resubmitted papers were again reviewed by at least three peers based on the scientific and technical quality, anticipated reader interest, and coverage of the conference scope. Finally, the Program Committee selected the 16 most mature and interesting papers. This led to an acceptance rate of 24.2 % out of the initially 66 submitted papers.

ICTERI as a conference series is concerned with interrelated topics of the development, deployment, and use of ICT that are vibrant for both the academic and industrial communities: education, research, industrial applications, and cooperation in ICT-related aspects. The selected papers reflect that scope and are grouped into three parts: (I) Frameworks and Tools; (II) ICT in Teaching and Learning; and (III) ICT in Research and Industrial Applications.

The volume begins with the written versions of the keynote talks planned for ICTERI 2014. Vladimir Gorodetsky gives a research-oriented overview of the phenomenon of Big Data. Nick Bassiliades presents an approach and a tool to compare university rankings so that safe conclusions about their reliability are drawn; this is done by extracting data from several ranking lists and linking it to the DBpedia linked open dataset.

Part I of the volume presents the state-of-the-art formal and algorithmic frameworks aimed at enabling further ICT development and also software tools based on such formalisms. It starts with a paper contributing a novel soft computing algorithm for multiplying fuzzy sets based on the formalism of universal analytic models. The second paper deals with developing a theoretical framework for formalizing the dynamics of systems at an abstract level and taking into account the continuous nature of time. The third paper formalizes complex event processing based on pre-automata. The fourth paper analyzes the algebraic properties of nominative data and functions. Finally, the

[1] See for example http://research.un.org/en/ukraine.

fifth paper reports on a software framework that uses rewriting rules for the automated adjustment of parallel tasks in a target platform for parallel computations. These papers also indicate the applications for which the presented results may be useful.

Part II focuses on using ICT in or developing ICT for teaching and learning, and their effects on didactics. The first paper presents an original approach of using demo hardware for enabling better and deeper understanding by IT students of the inner workings of a processor. The second paper studies the challenges of learning and using the Git system for distributed version control in development projects. This study bases on the experience gained in several different kinds of computing courses, and results in a set of recommendations. The third paper surveys the practices of IT formation at Ukrainian universities and proposes, as a result, an IT competence model as a methodological core for ramping-up the IT competences of Ukrainian students. The fourth paper focuses on the proposal of a pedagogical framework for improving the efficiency in learning ICT through a collaborative approach. This framework involves school teachers and university students of pedagogy and bases on the outcomes of a series of ICT workshops that had been held with representatives of these groups. The fifth paper addresses the efficiency of an ICT-based learning environment for postgraduate students based on a set of proposed criteria. Finally, the sixth paper deals with accounting for a dominant learning style of IT students for developing more effective targeted course materials.

Part III of the volume is dedicated to the applications of ICT in research and industry. The first paper looks at how ICT may help transform Bio-Design into an industrial strength discipline. It is based on the four core hypotheses which formed the basics for the development of the MENDEL Bio-Design software platform. The second paper deals with the applications in the Airspace industry. It proposes an approach to the analysis of the verification objectives and features of on-board information and control software at development and operation lifecycle stages. The third paper proposes an evolutionary game-theoretic approach for modeling and optimizing Internet connections in telecommunications. The fourth paper proposes a model of an evolutionary stable strategy for the choice of an appropriate social behavior pattern, based on the Cournot competition formalism, for the companies on a homogeneous product market. Finally, the fifth paper analyzes the performance of the off-the-shelf plagiarism detection software and elaborates the recommendations for their effective use in electronic publications domain

This volume would not have materialized without the support of many people. First, we are very grateful to all the authors for their continuous commitment and intensive work. Second, we would like to thank the Program Committee members and additional reviewers for providing timely and thorough assessments, and also for being very cooperative in doing additional review work at short notice. Furthermore, we would like to thank all the people who contributed to the organization of ICTERI 2014.

Without their effort there would have been no substance for this volume. Last but not least we would also like to acknowledge the proactive support of our editorial assistant Olga Tatarintseva.

September 2014

Vadim Ermolayev
Heinrich C. Mayr
Mykola Nikitchenko
Aleksander Spivakovsky
Grygoriy Zholtkevych

ICTERI 2014 Conference Organization

General Chair

Aleksander Spivakovsky	Kherson State University, Ukraine

Steering Committee

Vadim Ermolayev	Zaporizhzhya National University, Ukraine
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria
Mykola Nikitchenko	Taras Shevchenko National University of Kyiv, Ukraine
Aleksander Spivakovsky	Kherson State University, Ukraine
Mikhail Zavileysky	DataArt, Russian Federation
Grygoriy Zholtkevych	V.N. Karazin Kharkiv National University, Ukraine

Program Co-chairs

Vadim Ermolayev	Zaporizhzhya National University, Ukraine
Heinrich C. Mayr	Alpen-Adria-Universität Klagenfurt, Austria

Workshops Chair

Mykola Nikitchenko	Taras Shevchenko National University of Kyiv, Ukraine

Tutorials Chair

Grygoriy Zholtkevych	V.N. Karazin Kharkiv National University, Ukraine

Publicity Chair

Nataliya Kushnir	Kherson State University, Ukraine

Editorial Assistant

Olga Tatarintseva	Zaporizhzhya National University, Ukraine

Program Committee

Eugene Alferov	Kherson State University, Ukraine
Costin Badica	University of Craiova, Romania
David Camacho	Universidad Autónoma de Madrid, Spain
Michael Cochez	University of Jyväskylä, Finland
Tom Coupe	Kyiv School of Economics, Ukraine
Maxim Davidovsky	Zaporizhzhya National University, Ukraine
Anatoliy Doroshenko	National Technical University of Ukraine "Kyiv Polytechnic Institute", Ukraine
David Esteban	TECHFORCE, Spain
Anna Fensel STI	University of Innsbruck, Austria
Louis Feraud IRIT	Paul Sabatier University, France
Giacomo Fiumara	University of Messina, Italy
Vladimir Gorodetsky	St. Petersburg Institute for Informatics and Automation of The Russian Academy of Science, Russian Federation
Brian Hainey	Glasgow Caledonian University, UK
Sungkook Han	Wonkwang University, South Korea
Ville Isomöttönen	University of Jyväskylä, Finland
Mitja Jermol	Jožef Stefan Institute, Slovenia
Jason J. Jung	Yeungnam University, South Korea
Natalya Keberle	Zaporizhzhya National University, Ukraine
Oleksandr Kolgatin	H.S. Skovoroda Kharkiv National Pedagogical University, Ukraine
Christian Kop	Alpen-Adria-Universität Klagenfurt, Austria
Hennadiy Kravtsov	Kherson State University, Ukraine
Vladislav Kruglik	Kherson State University, Ukraine
Sergey Kryukov	Southern Federal University, Russian Federation
Vladimir Kukharenko	National Technical University "Kharkiv Polytechnic Institute", Ukraine
Nataliya Kushnir	Kherson State University, Ukraine
Alexander Lyaletski	Taras Shevchenko National University of Kyiv, Ukraine
Vira Liubchenko	Odessa National Polytechnic University, Ukraine
Frédéric Mallet	Université de Nice Sophia Antipolis, France
Wolf-Ekkehard Matzke	MINRES Technologies GmbH, Germany
Sergey Mazol	Academy of Public Administration, Belarus
Mykola Nikitchenko	Taras Shevchenko National University of Kyiv, Ukraine
Inna Novalija	Jožef Stefan Institute, Slovenia
Olga Ormandjieva	Concordia University, Canada
Vladimir Peschanenko	Kherson State University, Ukraine
Peter Popov	City University, UK
Gary Pratt	Eastern Washington University, USA

Carlos Ruiz	Playence, Spain
Abdel-Badeeh Salem	Ain Shams University, Egypt
Wolfgang Schreiner	Research Institute for Symbolic Computation (RISC), Johannes Kepler University of Linz, Austria
Vladimir A. Shekhovtsov	Alpen-Adria-Universität Klagenfurt, Austria
Mariya Shyshkina	Institute of Information Technologies and Learning Tools of the National Academy of Pedagogical Sciences of Ukraine, Ukraine
Marcus Spies	Ludwig-Maximilians-Universität München, Germany
Martin Strecker	IRIT, Paul Sabatier University, France
Olga Tatarintseva	Zaporizhzhya National University, Ukraine
Vagan Terziyan	University of Jyväskylä, Finland
Ville Tirronen	University of Jyväskylä, Finland
Nikolay Tkachuk	National Technical University “Kharkiv Polytechnic Institute”, Ukraine
Leo van Moergestel	Utrecht University of Applied Sciences, The Netherlands
Paul Warren	Knowledge Media Institute, Open University, UK
Alexander Weissbult	Kherson State University, Ukraine
Yulia Zaporozhchenko	Institute of Information Technologies and Learning Tools of the National Academy of Pedagogical Sciences of Ukraine, Ukraine
Iryna Zaretska	V.N. Karazin Kharkiv National University, Ukraine
Mikhail Zavileysky	DataArt Solutions Inc., Russian Federation
Tatyana Zaytseva	Kherson State University, Ukraine
Grygoriy Zholtkevych	V.N. Karazin Kharkiv National University, Ukraine
Daur Zukhba	Donetsk National Technical University, Ukraine

Additional Reviewers

Cristian Ramírez-Atencia	Universidad Autónoma de Madrid, Spain
Fernando Palero	Universidad Autónoma de Madrid, Spain
Martin Kammerlander	STI, University of Innsbruck, Austria

ICTERI 2014 Sponsors

Kherson State University	www.ksu.ks.ua
DataArt	www.dataart.com

Contents

Invited Contributions

Big Data: Opportunities, Challenges and Solutions

Vladimir Gorodetsky(✉)

St. Petersburg Institute for Informatics and Automation,
Russian Academy of Sciences, 39, 14-th Liniya,
199178 St. Petersburg, Russia
gor@iias.spb.su

Abstract. The problems related to the phenomenon of Big Data are currently among the top 10 hottest topics of information and communication technology. Big Data phenomenon refers to the data explosion observed today. At present, the term is widely used in different communities of many application domains, including researchers and practitioners. Big Data analysis can provide for many new opportunities in many respects motivating and stimulating industrial and commercial take-up of novel emerging technologies. The in-depth analysis of Big Data processing and analytics publications shows that the most of them write about "new opportunities" and "new challenges". However, very few papers present the solutions for predictive analytics that go beyond the limits of OLAP-like processing models and technologies. The goal of this paper is to outline in more detail not only the nature of opportunities and particular challenges but also some original solutions to attack them.

Keywords: Big data · Hadoop · Data modeling · Intelligent data analysis · Ontology · Dimensionality reduction · Causality analysis

1 Introduction

During recent decades, the role of big data in the information and communication technologies (ICT) was ever increasing, and currently ICT world is recognizing and practically experiencing its great scientific, commercial, economic and social values.

The term "Big Data" is not only about data themselves, although data is the central subject covered by this term. "Big Data" is the term encompassing the use of techniques to capture, process, analyze, and visualize potentially large datasets in a reasonable timeframe not accessible to standard IT technologies. By extension, the platform, tools and software used for this purpose are collectively called "Big Data technologies" [3].

The term "Big Data" introduced in 2008 was accepted, right away, by ICT scientific and business communities as a name for new research and application area aimed at developing of new approaches, algorithms and technologies that would be capable to cope with a number of heavy challenges caused by the novel properties of

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 3–22, 2014.
DOI: 10.1007/978-3-319-13206-8_1

data emerged to that time and named "big data".[1] Let us note that often one says that big data is *VVV*-data, while having in mind ***V**ariety*, ***V**olume* and ***V**elocity* as the main properties of big data understood as follows:

- ***V**ariety*: Diversity and multiplicity of (distributed, in particular) data sources composing the whole data set that can contain structured data measured in various scales (numerical, categorical, Boolean, ordered), semi-structured data (in XML, for example, or in other self-describing format) and unstructured ones (texts, for instance);
- ***V**olume*: Data are specified by enormous number (possibly, hundreds of thousands and more) of attributes determining data dimension and by many millions of instances determining the data set size; in many cases, dimensionality of data set overcomes its size that is not the case for "traditional" data sets;
- ***V**elocity*: Data are accumulated on permanent basis with high speed; in many cases, big data are composed of multidimensional data streams (time series), which volume can be of petabytes and more.

Big Data problem is currently one of 10 hottest ICT problems. For example, it was and is a subject of 81 projects of FP7, while addressing many particular Big Data topics ranging from data storing and content creation to big data predictive analytics and real-time knowledge extraction [3].

Big Data solutions are of great value for both science and industry, including many critical application domains, e.g., government, ecology, life sciences, economy, social security, manufacturing, medicine, social networks, genetics, road traffic management, and many others. Big data analysis provides, for data analyst, with many new opportunities making it possible, e.g. to discover hidden structures in big data samples, to on-line detect and classify anomalies in high dimensional high-speed data streams, to provide, for a person or organization, with multi-source cross-domain context-aware personalized recommendations, to analyze dynamics of social network structure and it's other properties, etc. Currently a number of very "heavy" but practically important high-profit big data applications are implemented. Examples, among others, are car rental customers' feedback on-line management system (Hertz company), real-time data streams analysis to identify traffic patterns (Royal Institute of Technology, Sweden), leveraging key data to provide proactive patient care (University of Ontario Institute of Technology), etc. [14]. Many impressive applications are solved in genomics, biotechnology and other domains.

It is worth to note that most practically interesting success stories and achievement, in big data processing, are achieved due to recent significant progress in powerful computing infrastructures and data warehousing. Indeed, current Big Data technology is usually implemented using *NoSQL* data base model, *Hadoop Distributed File System* integrated with *MapReduce* distributed programming model, IBM computing infrastructure and like. These and some other software and hardware means made it possible

[1] Hereinafter *Big Data* refers to the corresponding problem domain whereas *big data* refers to big data samples.

to implement several impressive big data based-applications [14] thus confirming the value of big data for science, economy, industry, etc.

Nevertheless, the state-of-the-art with Big Data processing technologies is not very optimistic. If to analyze the up-to-date publications devoted to big data processing, one can find a number of commonalities in their titles, keywords and contents. These commonalities are “new opportunities” and “new challenges”. Unfortunately, there are very few results relating to big data analysis that overstep, in a noticeable degree, the limits of routine OLAP-like technologies. In fact, much effort of ICT researchers is needed to make big data processing technology intelligent. However, there are practically no of powerful and efficient approaches, algorithms and software means for big data *intelligent* processing, e.g. *predictive* analytics or knowledge extraction.

The paper objective is to outline current state-of-the-art with the existing software means specifically developed for big data processing, to analyze the natures of particular big data challenges and to present some advanced solutions developed specifically for big data intelligent processing, in particular, to cope with some basic big data challenges.

The paper is organized as follows. Section 2 explains, by several typical examples of big data sets and some big data applications, what the big data is and why big data intelligent processing can be challenging. Section 3 outlines current state-of-the-art with the big data software infrastructures available now and gives several examples of applications successfully implemented via exploiting this infrastructure. Section 4 describes and analyses the basic computational challenges specifically peculiar to big data statistical processing, although there exist many challenges of other nature that are out of the paper mainstream. Section 5 outlines/proposes known/original approaches to resolve some key issues of big data intelligent processing intended to cope with high data dimensionality and instability of classical statistical algorithms as applied to big data. Among others, issues associated with semantic enrichment of source data using big data domain ontology and causal data analysis are paid the main attention. Conclusion summarizes the paper content and evaluates some perspectives.

2 Big Data and Big Data Processing: Typical Examples

It is difficult even to imagine or outline the diversity of currently available big data sets, multiplicity of applications that should deal with big data and types of tasks arisen in practice. Many examples can be found in science (biology, genetics, medicine, chemistry, life sciences, physics, etc.), industry (manufacturing, for instance), military (a variety of decision making/decision support tasks, information fusion, situation assessment and prediction, military logistics, etc.), business (wide multitude of recommender task, for instance), smart city, social networks, electronic government etc. To make clearer the big data-based task and to understand their peculiarities/challenges, let us exemplify three particular types of applications, that are (a) decision making/decision support/recommendation, (2) on-line anomaly detection and classification based on time series of enormous dimensions and (3) some network analysis problems, e.g. viral marketing and like.

Decision support is the most frequent kind of big data driven applications operating, as a rule, in real time. In these applications, the available data set is usually composed of a number of distributed databases storing streaming information of arbitrary types and of several abstraction levels inputting asynchronously from various data sources and in different time instants. Typical decision support/making application areas in military scope, for instance, are situation awareness, situation assessment, prediction, and some others. In such applications, the objectives of the data processing include the development of models of a number of *target attributes/variables*, e.g. categorical attributes corresponding to some predefined classes of situations of various abstraction levels, numerical attributes predicting values of several quantitative indices and criteria estimating future effects of former and current decisions using regression dependencies, etc.

In practice, the aforementioned models of target attributes should be built through data mining and machine learning. In this process, the critical task is selection of a small number of informative attributes approximating target variables with a predefined accuracy. Let us note that usually a few number of decades of attributes has to be filtered from, possibly, several decades of thousands or even hundreds of thousands ones. In fact, this kind of tasks puts great computational challenges falsifying the classical statistical models and algorithms (see Sect. 4). There is a great deal of particular applications in which the similar problem has to be resolved. For example, a large number of context-aware cross-domain personalized recommender systems fall into the same pool of applications. Indeed, modern recommender systems should to be learnt based on any available information sources in which the footprints of particular user activity appear, e.g., home page, Facebook and other social networks, e-mail, Internet shopping history, etc.

The second example that is on-line anomaly detection and classification deals with big data represented by time series of enormous dimensions. Mining this data aims to detect, in historical batch of time series, patterns of abnormal behavior, to cluster them and build the models of clusters' behavior patterns through data mining in terms of time series attributes of a reasonable dimensionality. The best example of such type of big data sets is given by NineSigma Company in its Request #699876 [20]. This Request seeks for a research team to develop algorithm and software "*that uses a natural language query to retrieve matching results from large-scale time-series data sets created from measurements taken at industrial plant facilities*", the document claims. This language is intended to query data patterns corresponding to the abnormal events. It is assumed that data patters under query are presented by keywords specified in natural language and indicating all possible "*root causes for system issues*" [20]. What is exclusive in this Request is the data set itself. It contains the records of 4 millions of sensors produced every 7 seconds during several former years. The total number of instances (records of sensor-generated values of 4 million variables) is more than 20 millions. The task claimed by NineSigma assumes that the time series in questions have already been represented in symbolic form, all types of abnormal patterns associated with "*root causes for system issues*" are known (mined) and interpreted. Of cause, time series symbolization and mining is too heavy task and currently it is under intensive research in data mining community. However, what is actually very specific for this claim is the novel additional problem associated with big data arisen in this application: *efficient real-time retrieval* of very long symbolic sequences of extra-high dimensions.

The third example of big data associated tasks concerns, e.g., with social networks analysis and its further specification in terms of viral marketing.[2] Observations proved that people usually trust more to information received from their friends, familiars and relatives than to information received from anonymous sources like advertisements. Therefore, social network-based advertizing information propagation that takes into account this fact, for instance, might be very productive and cheap. However, practice proved that information propagation speed critically depends on the nodes selected for information injection. This type of advertizing is called "viral marketing".

The task intended to search for a set of (social) network nodes providing the quickest information propagation is an example of network-based big data problem. What is worth to note is that viral marketing task found out a prototype of many other important applications based on processing of big data. One of them is social network-based management of opinion poll. Other one is an operation in cyber space intended to propagate computer worms/network viruses. The dual tasks are important too. The examples of such tasks are optimal strategy of computer network defense against worm/virus propagation through specific distribution of the defensive software over the network nodes, allocations of vaccination points to optimally prevent epidemics propagation, etc. In all these and formally equal network-based tasks, a great deal of VVV–data should be processed.

The presented examples explicitly demonstrate how diverse the big data sets are, how critical big data-driven applications can be, how challenging the big data processing, e.g. data mining and machine learning for predictive analytics, is, and how valuable the big data processing results are. These factors are the driving forces of the current intensive research, developments and expectations in the Big Data approaches, models and algorithms.

3 Big Data Technology State-of-the-Art and Examples of Success Stories

The main objective of big data analysis how the majority of experts formulates it is to *predict* future observations, states of systems, outputs of models, etc. However, at present, these tasks are solved, for big data, in a degree that is far from the best one. Let us analyze the practical strength of the *existing* methods, technologies and computer infrastructures that are used, at present, for big data processing.

It is well recognized that IBM computer infrastructure is the doubtless leader, among available hardware means, as well as *Hadoop* software platform that plays the same role among software infrastructures, for big data processing. Indeed, significant part of successful applications, in the area of big data processing, has been implemented via exploiting these hardware and software.

[2] This application is used only to explain the essence of the big data analysis tasks arisen in social networks.

Apache Hadoop is a Java-based software platform aimed at design and implementation of distributed big data management and processing. It is built based on *MapReduce* program model, uses open source software library and includes own file system, *Hadoop Distributed File System* (*HDFS*). It provides for quick access to large-scale data stored at several computers in redundant mode. The data redundancy is used to provide for data processing systems with fault tolerance and availability of data for parallel processing. Parallel mode of data storage provides also with quick data access though system input-output capacity is limited. Every computer of computing infrastructure, in *HDFS*, is called *DataNode*. The size of data block stored in *HDFS* is big enough – it is of 64 Мб that makes smaller the total size of meta–data. *HDFS* system provides with high speed of reading for data stored in sequential mode on hard disks. The meta–data comprising file names and storage addresses may be modified by multiple clients and therefore needs for permanent synchronization. To provide for quick access to meta–data, the latter are stored on the single computer called *NameNode*. To get access to a file and to process it, a client connects with *NameNode*, retrieves the file list and their addresses on corresponding *DataNodes* and then reads the target file, by blocks, directly from *DataNodes*.

MapReduce is a program model of large scale data parallel processing. It implements traditional *divide and conquer* approach, while dividing the data into parts with the subsequent parallel processing of them on different processors and fusion of the processing results. *MapReduce* software is responsible for correct implementation and execution of these functions.

It is important to emphasize that *Apache Hadoop* is a tool providing for reading of big data composed of components of various structures and for data *preprocessing* to prepare them for the subsequent analysis. It other words, *Apache Hadoop* is not a tool for big data analysis. It is a software tool for big data preprocessing. However, a lot of publications identify big data preprocessing and big data analysis itself.

What concerns with big data analysis software it is produced by traditional players on this market those are IBM, Oracle and like. For example, IBM is the developer of such widespread software tools as *InfoSphere BigInsights*, *BigSheets*, *InfoSphere Streams*, etc. *InfoSphere BigInsights* is a toolkit of analytical means intended to operate with the large scale structured and unstructured data arrays including complex text data. IBM *BigSheets*, the other software tool of IBM for big data analysis, is designed to implement a number of functions intended for increasing the computer performance, providing for security and supporting for data administration. *InfoSphere Streams* software tool is destined for extraction from data the streaming patterns meeting predefined templates. It is capable to work with data streams lasting from several minutes to several hours while providing for real time collection of target patterns. *InfoSphere Streams* can fuse the streams and form new data sets from several data streams. These and some other software tools used at present for big data analysis are described in [14–17] in more details.

To understand the real capabilities and power of the existing software tools available for big data processing, some representatives of big data–driven applications developed recently and evaluated, by IBM, as success stories are described below.

1. *Managing the environmental impact on rivers by processing of streaming information* (Beacon Institute, Clarkson University and IBM [14]). This is real-time systems for rivers and estuaries intended for continuous monitoring of physical, chemical and biological data from many points in New York's Hudson, Mohawk and St. Lawrence rivers by means of an integrated network of sensors and robotics, as well as by means of mobile human monitoring. A networked array of sensors provides for the data necessary to estimate the local spatial values of many attributes, e.g. temperature, pressure, salinity, turbidity, dissolved oxygen and other basic water chemistry parameters. The sensor-based and other information is transmitted in real time to the system storage. The data processing objective is to understand interactions between river ecosystem, local population and environment. The ultimate objective is resource management to respond to chemical, physical and biological alterations of water [14].

The data processing is performed by IBM®*InfoSphere®Stre*ams software system providing for data collection, data analysis as well as visualization of chemical components movement. It also monitors the water quality and provides for fish population protection during its migration. In general words, the system helps to scientists to better understanding of interactions inside heterogeneous environment. Let us note that, although the aforementioned system actually processes big data and solves the important and practically useful task and subtasks, it does not provide for predictive data analytics formulated as basic task of big data analytics.

2. *Analysis of real-time data streams to identify traffic patterns* (KTH-Royal Institute of Technology, Sweden [14]). The system is designed for real-time analysis of road traffic in a big city through on-line gathering of multiple data sources including GPS from large numbers of vehicles, radar sensors on motorways, congestion charging, weather, etc. Its objective is to better manage traffic through integration and analysis of the available data mentioned above. The system supports for identification of the current road traffic conditions, estimation of time needed to travel from one point of the city to the other one and is capable to propose alternative routes for road traffic. The basic analytical software used in this system includes *IBM® InfoSphere™ Streams*, *IBM System Storage® DS3400* and is deployed on an *IBM Blade Center* infrastructure.
3. *TerraEchos@IB*M [14]. It is the surveillance sensor system intended to detect, classify, locate and track potential threats (both mechanical and biological; above and below ground) to secure perimeters of the border areas. The application is deployed to protect Engineering national laboratory of nuclear and energy research of the U.S. Department of Energy from natural disaster and terrorism. It continuously measures and analyzes real-time information associated with motion of humans, animals and the atmosphere (wind, for instance). Gathering and processing of the information is performed simultaneously. The system is developed and implemented on the basis of *TerraEchos Adelos S4 System* (formerly designed for US Navy purposes to analyze the streaming audio information generated by a lot of acoustic sensors), *IBM InfoSphere Streams* software and *IBM BladeCenter* computer infrastructure.

Information about several other successful big data – driven applications developed based on *Apache Hadoop* and IBM software and deployed using IBM computer infrastructures can be found in [14].

However, all described successful stories of big data-driven applications as well as most of others developed to the present days possess limited capabilities. In what concerns with the data analytics they "reinvent wheels" developed in previous decade and, in fact, implement limited data analytics, e.g. analytics corresponding to the OLAP technology developed more than decade ago. It is too few successes in big data predictive analytics, so far. The matured achievements in the area of intelligent data analytics for the moderately–sized data scale are much more powerful. The main obstacles are posed by several problems of algorithmic nature caused by some negative effects peculiar to statistical processing of big data analyzed in the next section.

4 Big Data Problems and Challenges: Statistical View

The practical limitations of intelligent big data analytics are caused by several experimentally discovered effects appearing with classical statistical methods and algorithms intended for association and correlation analysis. These effects make it impossible to use the classical statistical dependency discovery methods, in big data case. Of cause, the main reason of appearance of these effects is the huge dimensionality of big data attribute space that can achieve decades of thousands and even millions. One of the peculiarities of big data is that their dimensionality can be much more than data size (the number of data set instances) and this data property implies the most difficult situations for data dependency analysis exploiting statistical methods and algorithms. Let us recall that search for dependencies among attributes of big data to specify a model of object or process is the key task of intelligent data analysis.

In was stated earlier, in this paper that, among the attributes presenting data, a specific role belongs to the so-called *target attributes* (variables). These attributes can correspond to quality factors (quality indices) to be optimized, to labels of classes of an object states or system situations, or to some other types of variables of high interests. Assessment or prediction of such variable values is usually a focus problem of big data processing. In general, these assessment/prediction needs are realized via building of approximate formal models of the target variables. In this modeling, the most difficult problem is to select the model variables from huge set of them. Theoretically, this task is of exponential complexity, and if the total dimension of big data attributes corresponds to hundreds of thousands, than it is difficult to solve it not only due to computational complexity. However, even if the unlimited computational resources were available there would be no guarantee that the task could be successfully solved using well-developed statistical methods and algorithms.[3] Classical statistical methods and algorithms, as a rule, fail for such data dimensionalities due to some specific computational effects peculiar to high dimensional statistical data processing.

[3] Unfortunately, there are many industrial practitioners who still trust that unlimited computing resources are capable to cope with any big data-related problem.

A comprehensive analysis of these effects is done in [6]. Let us outline some critical of them while following the latter publication.

Indeed, attribute values results from sensor-based measurements and, therefore, inevitably contain errors. If the total number of model variables involved into statistical processing is huge the computational errors are accumulated and begin to remarkably dominate over the desired signal thus leading to invalid results. Especially catastrophic consequences occur if computations include matrix inversion, correlation analysis, search for eigenvalues and eigenvectors, etc. What is important that traditional regularization methods do not work, in such cases, as a rule.

This very negative effect called *error accumulation* is impressively demonstrated in [6] using classification task as an example. It considers binary classification problem using two training data samples of size $n = 100$ and dimension $d = 1000$ having Gaussian distributions, i.e. $X_1, \ldots, X_n \sim N_d(\mu_1, I_d)$ *and* $Y_1, \ldots, Y_n \sim N_d(\mu_2, I_d)$, where I_d – the unity covariance matrix of dimension d selected as the same for both data sets. In the first class, all the feature vector mathematical expectation components are set zero, i.e. $\mu_1 = 0$. In contrast, the first 10 components of the same feature vector μ_2 of the second class are set nonzero with value 3 whereas the rest of them are also set zero. It was assumed that classification rule is determined, in some way, in two-dimensional space, which axes correspond to first two principal components of the best m-dimensional feature space. In other words, each data set instance of the first and the second classes is presented by a point in this two dimensional space and the coordinates of every point are equal to the projections of the instance features vector onto the above mentioned principal components.

The authors of [6] investigated the *error accumulation effect* as a function of the feature vector dimension, i.e. as a function of m for $m = 2$, $m = 40$, $m = 200$ and $m = 1000$, whereas the total number n of the instances in both data sets is equal to 100. The results of this investigation showed that for $m = 2$ the instances of different classes are discerned well and linear classification rule should provide for high quality performance. If $m = 40$ the classes discernibility is much worse, and what is important is that the larger m, the more severe the noise accumulation influence is. For example, if $m = 200$ and $m = 1000$ the error accumulated remarkably dominates the useful signal and the classes cannot be discriminated within the selected two-dimensional space. The authors of [6] indicate that low discriminative power results from the existence of many weak features, which increase noise accumulation while not reducing the classification error. In particular, the authors discovered that first 10 features contribute to the classifications power whereas the rest of them do not.

This fact reflects a common property of big data-driven modeling of target variables. For example, [6] refers to [8] and [13] discovered that, due to noise accumulation, "*conventional classification rules using all features perform no better than random gue*ss".

However, variable selection that is challenging in any data-driven modeling becomes much more challenging for big data case also due to several other negative effects specifically peculiar to big data statistical processing. These effects are spurious correlation, incidental endorgeneity, heterogeneity, and measurement errors [6]. Let us outline the natures of spurious correlation and incidental endorgeneity while referring to the source [6] for understanding of the other effects.

Spurious correlation is an effect of appearance of spurious sample-based correlation between theoretically independent variables. In [6], this effect is illustrated using computation of sample-based correlation coefficients of independent variables $X = [x_1, \ldots, x_d]^T$ having Gaussian distribution $N(0,\ I_d)$, where I_d is unity covariance matrix of dimension d. Experimental analysis of distributions (histograms) for maximal values of sample-based correlation coefficient $\hat{r}$

$$\hat{r} = \max_{j>1}\{corr(x_1, x_j), \tag{1}$$

of variable x_1 with all other variables of vector X exhibits existence of dependencies between theoretically independent variables [6]. The experiments were provided for the data sample of size $n = 60$ for two values of dimensionality d that are 800 и 6400. The correlation coefficient distributions (histograms) were built based on a constant size that is 1000 instances. These experiments proved that the most part of them exhibits strong sample correlations. It is worth to mention that the relative part of such variables as well as the module of the averaged correlation coefficient values increase if the data dimension increases, according to the experiments for dimensionalities 800 и 6400.[4]

Spurious correlation effect is dangerous since it misleads knowledge engineer who builds the target variable models since it results in false scientific conclusions leading to selection of unrelated variables as potential causes of these or those target variables. A simple illustration is done by traditional liner observation model [6]:

$$Y = \boldsymbol{X}^T A + \beta, \text{ where:} \tag{2}$$

- $Y \in R^n$ – the vector of the observable values of target variable y in n experiments; $Y^T = [y_1, \ldots, y_n]$ – vector of observation;
- $X = [x_1, \ldots, x_d]^T \in R^d$ – the vector of the model variables, which values in n experiments are known (assigned or measured);
- $\boldsymbol{X} \in R^{n \times d}$– the matrix composed of n instances $X^{(k)} = [x_1^{(k)}, \ldots, x_d^{(k)}]^T \in R^d$ of X;
- A – the vector of coefficients of the linear model of the target variable y.
- β – the vector of additive noise with independent components having Gaussian distribution of the components and correlation matrix $Var(\beta) = \sigma^2 I_n$.

In practice, the model dimension d (the number of model variables and the number of coefficients under search) is to be minimized in order to reduce the linear estimation task, to make the resulting model semantically interpretable and, what is the most important, to decrease the influence of error accumulation effect. However, model dimension reduction is computationally hard, and it is subjected to spurious correlation effect, in big data case. One of the approaches to avoid selection of unrelated variables is use of causal analysis that is accentuated in many papers, e.g., in [1, 6].

[4] See [6] for impressive graphical illustrations of the error accumulation and the spurious correlation effects.

In addition to false modeling, spurious correlation can lead to false statistical inference. In particular, it is shown in [7] that if a model contains too many unrelated variables erroneously selected due to spurious correlation effect then estimations of the model coefficient dispersions σ^2 find out too reduced, as a rule. This can lead to the false conclusions with regard to the statistical significance of the model coefficients selected.

One more specific effect resulting from big data dimensionality called *incidental endorgeneity* is caused by the arising of statistical dependency between noise and model variables. If to refer to the model

$$y = \sum\nolimits_{j=1}^{d} a_j x_j + \beta \tag{3}$$

built through resolving of the linear observation model (2) then incidental endorgeneity corresponds to the existence of the correlations between noise β and components of attribute vector X. The majority of existing statistical estimation models assumes the independence of the noise β and the variables composing the vector X. It is important to note that, in contrast with the spurious correlations, incidental endorgeneity concerns with really existing correlations. [6] refers to the two causes of this effect, in big data case:

1. If the total number of variables is numerous then the probability of existence of real dependencies between noise and some model variables is high.
2. Data can be obtained from many sources with various accuracy and biases and, therefore, some dependencies in question can result from data fusion procedures.

However, this problem is weakly investigated so far [6].

Other big data-specific challenging effects exist too. They require reconsidering approaches to and algorithms for solution of the tasks of big data statistical analysis. According to [6], the main attention should be paid to adaptive and more robust data processing algorithms using various types of heuristics, e.g.:

– use of any reasonable measures to reduce the dimensionality of data under processing;
– use of grounded approaches to filter the set of data attributes forming the data-driven target variable modeling.

Some approaches intended to cope with the big data challenges concerning with variable choice, data and task dimensionality reduction are described in the next section.

5 Big Data Analytics: Selected Approaches and Algorithms

5.1 Data Model and Ontology Role

The same big data set can be exploited for solving multiple big data-driven applications aiming, as a rule, at prediction of a number of target variables specific for each particular application. Thus, one of the typical and most important tasks of the big data analysis is reduced to the data-driven modeling of several target variables, and, therefore, big data representation should support multiple modeling of big data.

In this respect, an important role, for data representation, belongs to ontology, according to the modern point of view. Ontology-based big data model provides, for data-driven target variables modeling, with clear semantic interpretation of the model variables, enriches available data with expert knowledge and simplify solution of data and information fusion if data are distributed over multiple heterogeneous sources. Other important advantages of using ontology-based approach to big data (predictive) analysis exist too.

However, manual design of ontology for big data is a labor–intensive and error–prone problem: it is unreal to develop interval ontology for data set of huge dimensionality containing thousand of concepts while preserving feasibility of time spent, ontology correctness and consistency. It is especially hard to discover and specify ontology concept categorization, in manual mode. Fortunately, recently a number of approaches and means for supporting for automated ontology design have been proposed [2]. Among them, Wikipedia damp–based categorization or use of Linked Data web [4] can significantly reduce the manual efforts needed for hierarchical structuring of the ontology concepts. These approaches to ontology design automation attract the increasing interest of the knowledge engineers and look as most promising.

The author's experience proved [9] that, in many cases, enriching data only with ontology concept hierarchy allows for simpler detection of the redundant attributes (as a rule they correspond to the lower level concepts) thus decreasing the dimensionality of target variable modeling task. Ontology concept hierarchy positively contributes to reduction of the data attribute aggregation task (see Sect. 5.3). Currently, the ontology design automation is a problem of intensive research, in intelligent information technology scope and Big data problems is one of the serious motivations, for this.

5.2 Big Data Size and Dimensionality Reduction

The basic paradigm of big data processing is minimization of dimensionality of data involved into processing at each step of general processing technology. The methods realizing this paradigm can be divided into four groups:

1. Approaches based on decomposition of the whole set of data instances into blocks of reasonable sizes, processing these groups separately and then fusion/assembling the results, in some way. Sampling, e.g. random sampling, is another approach of processed data size reduction. Instead of processing the whole data set instances, one can select, at random, for example, a subset of data instances and infer the target statistics based on the latter. The natural requirement to sample-based data size reduction is preserving statistical significance of the target sample-based estimations. Let us note that Apache Hadoop software implements this strategy of processed data size reduction.
2. Data dimensionality reduction approaches that divide the data attribute space into several subspaces of reasonable dimensions, perform the required processing of the data of less dimensionality in parallel or sequentially and fuse or assemble the

results, in some way. In general case, these approaches assume to involve in the processing the whole attributes. Again, Apache Hadoop is a software tool supporting for approaches of this type.

3. Filtering the whole set of attributes in order to select an attribute subspace to solve the target task in the subspace of reduced dimensionality. To decrease the processing space dimensionality, some rules, heuristics or on other intuition are used. One of such approaches specifically developed for big data case is considered below.
4. Transformation of data attributes to more aggregated granularity. This type of approaches intends to reduce the cardinalities of domains of all attributes or its subset, thus decreasing the total search space cardinality. One of such aggregation-based dimensionality reduction approach is described below. It substitutes the source domains of the attributes with several statements about its properties. An additional advantage of such approach is that heterogeneous space of attributes (source variables) is transformed to the homogeneous attribute space, and what is not less important is that that final homogeneous space of the big data attributes is Boolean.

Two approaches implementing the third and the fourth types of the aforementioned approaches are considered below in Subsects. 5.3 and 5.4, respectively.

5.3 Data Attribute Aggregation

Transformation of data attributes to more aggregated granularity is an approach aimed to reduce big data dimensionality In this section, a simple but productive and efficient algorithm to big data feature (attribute) aggregation is described. Its additional advantage is that it provides, for final data space representation, with the homogeneity independently of the measurement scales of the source data dimensions. It was proposed in [11] and, to present days, it is experimentally validated on the basis of various applications.

Let us assume that data model is represented it terms of domain ontology and every variable mentioned below is presented in terms of ontology notions and/or notion attributes and, therefore, well semantically interpretable in domain context. These features can be heterogeneous with regard to their measurement scales.

The aggregation algorithm idea is explained below by the example of classification task. Let $X = \{x_1, x_2, \ldots x_n\}$ be a set of data variables and n be the variable set dimensionality that can be huge, in case of big data. Let $\boldsymbol{X}_{\mathrm{i}}$ be the domain of the attribute x_i, i.e. $x_i \in \boldsymbol{X}_{\mathrm{i}}$, and $\boldsymbol{X} = \boldsymbol{X_1} \times \boldsymbol{X}_2 \times \ldots \times \boldsymbol{X}_n$. It is also assumed that each domain $\boldsymbol{X_i}$ is discrete and of finite cardinality. Additionally, every instance of data sample corresponds to an object presented $X \in \boldsymbol{X}$ that is mapped a class label $\omega_k \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$.

The basic idea of aggregation is that every attribute x_i is considered as a one–feature classifier as follows, for example:

$$if\ F(x_i \in \boldsymbol{X}_i^{(k)})\ \ then\ \ \omega_k \tag{4}$$

where $F(x_i \in \boldsymbol{X}_i^{(k)})$ is an unary predicate and $\boldsymbol{X}_i^{(k)} \subseteq \boldsymbol{X}_k$. On the other hand, every classifier can be interpreted as a feature, e.g. $F(x_i \in \boldsymbol{X}_i^{(k)})$ can be interpreted as an aggregated feature of a new feature space. Of course, the discrimination power of such feature, in general case, is questionable. It depends on the selected feature x_i and on the selected domain $\boldsymbol{X}_i^{(k)}$, for given ω_k. To search for useful aggregates, a metrics of aggregate usefulness should be correctly selected. One of such metrics is described below.

Thus, feature aggregation task can be formulated as follows:

For all $i \in \{1, 2..., n\}$ and $\omega_k \in \Omega$ to find $\boldsymbol{X}_i^{(k)}$ such that

$$P(\omega_k / x_i \in \boldsymbol{X}_i^{(k)}) > P(\omega_v / x_i \in) + \Delta, \text{ for all } \omega_v \neq \omega_k, \tag{5}$$

where Δ is a positive threshold $0 < \Delta < 1$, $P(\omega_k / x_i \in \boldsymbol{X}_i^{(k)})$ and $P(\omega_v / x_i \in \boldsymbol{X}_i^{(k)})$ are the values of conditional probabilities of the class ω_k and ω_v if the predicates $F(x_i \in \boldsymbol{X}_i^{(k)})$ and $F(x_i \in \boldsymbol{X}_i^{(v)})$ take the value true, respectively. Of course, some of $\boldsymbol{X}_i^{(k)}$, $i \in \{1, 2, \ldots, n\}, k \in \{1, 2, \ldots, m\}$, can by empty, i.e. $X_i^{(k)} \in \{\emptyset\}$.

Thus, each domain $\boldsymbol{X}_i^{(k)}$ is such that it contains those values of the variable (feature) $x_i \subseteq X_i$ that occurs more frequently with the object instances of the class ω_k than with the object instances of any other classes $\omega_v \neq \omega_k$. In other words, for each particular feature x_i the rule "*if* $\boldsymbol{X}_i^{(k)} \subseteq \boldsymbol{X}_i$ *then* ω_k", $k \in 1, \ldots, m$, is capable to recognize the objects of the class ω_k with the probability $P(\omega_k / x_i \in \boldsymbol{X}_i^{(k)}) > \delta_k$,) and this probability is larger than the corresponding probabilities $P(\omega_v / x_i \in \boldsymbol{X}_i^{(k)})$ for any other classes $\omega_v \neq \omega_k$. Since each $\boldsymbol{X}_i$ is discrete and of finite cardinality, the search for domain $\boldsymbol{X}_i^{(k)}$ can be done through enumeration of the domain $\boldsymbol{X}_i$ for $i \in 1, \ldots, n$.

The result of the aggregation procedure is the set of predicates $F_i^{(k)} = F(x_i \in \boldsymbol{X}_i^{(k)})$ with the given truth domains $\boldsymbol{X}_i^{(k)}$ marked by the index of the class ω_k and assigned sample-based conditional probabilities.

The next step of feature space reduction is *feature filtering* procedure. It consists in detection of the so called *good aggregates*. Slightly re-formulating the Condorcet theorem [5], it is said that an aggregate $F_i^{(k)}$ is "good" if accuracy of the classification rule (4) is strictly more than *0.5*. Otherwise an aggregate is *bad*. Let us remind that classification rule accuracy is computed as follows:

$$Acc[\textit{if } F(x_i \in \boldsymbol{X}_i^{(k)}) \textit{ then } \omega_k] = P(\omega_k / F_i^{(k)}) + P(\bar{\omega}_k / \neg F_i^{(k)}) \tag{6}$$

Thus, the filtering procedure aims at detection of good aggregates and deletion of the bad ones. The latter is done in two steps. At the first filtering step, each aggregate found according to condition (5) is used to build one-feature Naïve Bayesian classifier (4) and its data sample-based testing in order to compute contingency matrices diagonal probabilities $P(\omega_k/F_i^{(k)})$ and $P(\bar{\omega}_k/\neg F_i^{(k)})$[5] with the subsequent filtering them according to the following rule:

$$\begin{aligned} &\mathit{if}\ P(\omega_k/F_i^{(k)}) + P(\bar{\omega}_k/\neg F_i^{(k)}) > 0.5\ \mathit{then\ the\ aggregate}\ F_i^{(k)}\ \mathit{is\ preserved}, \\ &\quad \mathit{otherwise\ it\ is\ filtered} \end{aligned} \tag{7}$$

The result of this step is the set of "*good*" aggregates that can also be considered as "*good*" classifiers, according to the Condorcet theorem.

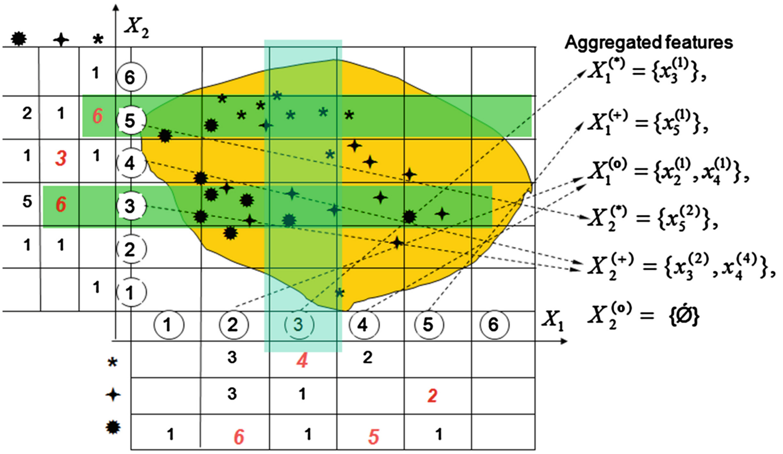


Fig. 1. Aggregation in two-dimensional feature space for data instances of classes marked as ✹, ✦, and *. Aggregated feature list: $X_1^{(*)} = \{x_3^{(1)}\}$, $X_1^{(+)} = \{x_3^{(1)}\}$, $X_1^{(o)} = \{x_2^{(1)}, x_4^{(1)}\}$, $X_2^{(*)} = \{x_5^{(2)}\}$, $X_2^{(+)} = \{x_3^{(2)}, x_4^{(4)}\}$, $X_2^{(o)} = \{\emptyset\}$.

Figure 1 illustrates the essence of the search for aggregates $F_i^{(k)} = F(x_i \in \boldsymbol{X}_i^{(k)})$.

Let us summarize the properties of the described feature aggregation and filtering approach, its advantages and applicability to big data case.

1. It uses natural measure for estimation of the discriminative power of a single-feature subspace: conditional probability of class ω_k subject to $x_i \in \boldsymbol{X}_i^{(k)}$.
2. The aggregated feature space possesses very valuable property: It is homogeneous independently of the heterogeneity of the source feature space. Moreover, all the aggregated features are Boolean and presented in the form of unary predicates $F(x_i \in \boldsymbol{X}_i^{(k)})$. Let us note that frequently used variants of reducing heterogeneous feature space to Boolean one consider propositions but not predicates. It can be shown that this variant of aggregated feature space is capable to prevent error accumulation and spurious correlation effects peculiar to the classical statistical algorithms as applied to big data. The reason of this advantage is that reduction of

[5] At this step, the testing procedure has to be applied only for data subset assigned label $\bar{\omega}_k$.

feature space does not use operations with correlation matrices and operates with integers.

3. For each class $\omega_k \in \Omega$, specific feature space is built, i.e. class – specific classification rules use different features. This property is novel that distinguishes the proposed approach from other existing ones.
4. The aggregated features are presented in terms of application ontology and therefore are well interpretable.
5. The procedure computing the aggregates $\boldsymbol{X}_i^{(k)}$ presented in the formula (5) is of complexity $O(n_i \times m)$ where n_i is the cardinality of the domain $\boldsymbol{X}_\mathrm{i}$ and m is the total number of classes. It is worth to note that the described $\boldsymbol{X}_i^{(k)}$ domain search algorithm is well suited for Hadoop–based parallel technology.

Let us also note that any other classification model could be exploited instead of Naïve Bayes to estimate the discriminative power of the aggregated features and therefore the approach is potentially improvable, through this possibility. One more potential improvement is search for binary predicates $F(x_i \in \boldsymbol{X}_i^{(k)}, x_j \in \boldsymbol{X}_j^{(k)})$ instead of unary ones. An algorithm computing two dimensional informative subspaces was, e.g., proposed in [24].

Figure 1 illustrates the aggregation algorithm by an artificial toy example. In fact, this figure is self-explanatory. It graphically demonstrates how $\boldsymbol{X}_i^{(k)}$ are aggregated, where k={*, +, ✹} are the class labels and the feature set is $\{X_1, X_2\}$.

5.4 Causal Analysis of Aggregates

The rules (4) built based on the results of feature aggregation and filtering are association classification rules. It is theoretically proved. in e.g. [1, 23], that among associations (4) the most interesting are such of them which present causal dependencies [1]. The researchers in Big Data area also state that causal analysis of big data is a way to improve design of data–driven target variable models [6].

However, in general case, efficient detection of causal dependencies among association ones using formal procedures is a challenging problem. Classical approaches based on Bayesian network [21] and Causal Bayesian networks [22] define causal dependencies in terms of Markov blanket. Markov blanket for a variable (a node of Bayesian Network) is the set of the node "parents", "children" and "other parents" of the node "children". To discover causal dependencies between variables using data sample, it is necessary to learn for Causal Bayesian Network, but this task is computationally hard. The existing software developed for Bayesian network learning is capable to cope with the data sample of small dimensions, e.g. for a couple of decades or close. Therefore, Bayesian network model cannot be considered as a means to detect causal dependencies in big data which dimensionalities are of more that decades of thousands.[6] Since the middle of 1990-th, this challenge has motivated the search for

[6] Practically, causal analysis based on Bayesian network model can be used for data dimensionalities of no more than 20.

alternative approaches to causality analysis. The basic idea is to build a numerical metrics measuring the "strength" of causal dependency between a pair of variables, see [23, 19], for instance. Such metrics are called *causal–associative* ones.

One of such metrics was proposed in [11] and then in [10]. It assumes that each variable of data sample is Boolean and modeled as a random event. The proposed metrics estimates the strength of causality between random events A and B in the following form:

$$R(A,B) = P(B/A) - P(B/\bar{A}) \tag{8}$$

The metrics $R(A,B)$ computed for a pair of random events A and B is known, in probability theory, as *regression coefficient of the random event A on random event B*. In formula (6) the first term corresponds to the conditional probability of the random event B if the random event A has occurred, *w*hereas the second term corresponds to the conditional probability of the same event if the random event A has not occurred. Conceptually, this metrics reflects adequately what causal dependency, from human viewpoint, is. It also meets the formal requirements to a causal metrics: it has to be (a) non–commutative, $R(A,B) \neq R(B,A)$, in general case, (b) $R(A,B) \in [-1,+1]$, (c) $R(A,B) = 0$ for causally independent random events and (d) it achieves maximum of its module when the random events are functionally dependent.

The value of causal metrics $R(A,B)$ can also be computed in terms of unconditional sample-based probabilities:

$$R(A,\ B) = \frac{P(A,B) - P(A)P(B)}{P(A)[1 - P(A)]}. \tag{9}$$

If to return to the classification task exemplifying the attribute aggregation procedure (Sect. 5.3) and to use the denotations introduced there then metrics of causal dependency between an aggregated feature $F(x_i \in \boldsymbol{X}_i^{(k)})$ and class label $\omega_k \in \Omega$ can be computed as follows:

$$R(F(x_i \in \boldsymbol{X}_i^{(k)}), \omega_k) = P(\omega_k / F(x_i \in \boldsymbol{X}_i^{(k)})) - P(\omega_k / \bar{F}(x_i \in \boldsymbol{X}_i^{(k)})) \text{ for all } i \text{ and } k, \tag{10}$$

or, in terms of unconditional probabilities,

$$R(F(x_i \in \boldsymbol{X}_i^{(k)}), \omega_k) = \frac{P(F(x_i \in \boldsymbol{X}_i^{(k)}), \omega_k) - P(F(x_i \in \boldsymbol{X}_i^{(k)})) \times P(\omega_k)}{P(F(x_i \in \boldsymbol{X}_i^{(k)})) \times [1 - P(F(x_i \in \boldsymbol{X}_i^{(k)}))]} \tag{11}$$
$$\text{for all } i \text{ and } k.$$

The causality metrics considered above determines the strength of the causal dependency between an aggregated data concept/attribute $F(x_i \in \boldsymbol{X}_i^{(k)})$ and the target variable ω_k . This metrics can be used for second step threshold-based filtering of the aggregates found at previous step:

$$\left| R(F(x_i \in \boldsymbol{X}_i^{(k)}), \omega_k\ ) \right| \geq \delta_{min}^{(k)} \text{ for all } i \text{ and } k \tag{12}$$

The aggregates meeting the condition (11) can be used further as the premises of the classification rules in the form

$$\mathit{If}\ R(F(x_i \in \boldsymbol{X}_i^{(k)}), \omega_k\ ) \geq \delta_{min}^{(k)}\ \mathit{then}\ F(x_i \in \boldsymbol{X}_i^{(k)}) \rightarrow \omega_k\ , \tag{13}$$

or as the arguments of the model of the target variables of other types.

The results of experimental validation of the metrics (7) in the form (10) or (11) can be found in [10, 11], for example.

5.5 Further Minimization of Target Variable Model Dimensionality

Causal analysis described above is aimed to discover a set of one-literal causes of target variables. The selected causes that are data ontology attributes/concepts can be dependent, in their turn, and such dependencies can be, in some cases, strong enough. On the one hand, this fact argues that the set of causes can be redundant and, therefore, further reduced. On the other hand, the existence of strong dependencies among causes can mislead the designer of the target variable model. An example below demonstrates such mislead.

Let us again consider the case of classification when the target variable corresponds to a class label, and the set of the target variable causes can by clustered into subsets of strongly dependent ones. Let the classifier designer intends to use voting model. In such case, he/she cannot use traditional schemes of voting, instead, he/she can use cluster-based voting, i.e. assign one vote to each cluster, but not to each member of the clusters, as in usual scheme. Indeed, it is most probable that all the rules like (12) of each cluster of strongly correlated causes will vote evenly. Therefore, it is reasonable to take into account one vote of each cluster.

In the multiple classifier scope, this problem is well known as classifier diversity one [12, 18]. In the existing literature, several heuristic approaches are proposed to cope with this problem. The aforementioned approach based on clustering of the causes into the subsets of strongly dependent ones with the subsequent cluster-based voting can be considered as a theoretically grounded approach. It is currently in progress of the development and validation. It is an additional source for big data dimensionality reduction that is highly necessary to cope with big data predictive analytics.

6 Conclusion

The paper considers various aspects of the current value of, challenges and achievements in the area of intelligent processing of big data. Analysis of the current state-of-the-art in this area showed that the existing approaches, mathematical methods, algorithms and software tools are basically not capable to support for rigorous big data analytics so far, e.g. to support for predictive analytics. A conclusion should rather be that big data challenges prevail over the productive solutions. The existing literature, in this area, is mostly tell about numerous challenges and much less about efficient and effective solutions for fundamental tasks of big data predictive analytics. A well founded justification of this statement can be found in [6] that analyzes the fundamental challenges associated with the use of the existing statistical algorithms for big data of huge dimensionality.

The paper outlines some basic directions and particular approaches and algorithms that care capable to achieve some progress in resolving the basic challenges of the big data predictive analytics.

Acknowledgment. This research is supported by the Project No. 1.12 of the Research Program entitled "Information Technologies and Methods for Complex System Analysis" supervised by Nano- and Information Technology Branch of the Russian Academy of Sciences.

References

1. Aliferis, C.F., Statnikov, A., Tsamardinos, I., Xenofon, S.M., Koutsoukos, D.: Local causal and markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation. J. Mach. Learn. Res. **11**, 171–234 (2010)
2. Bedini, I., Nguyen, B.: Automatic Ontology Generation: State of the Art. http://bivan.free.fr/Janus/Docs/Automatic_Ontology_Generation_State_of_Art.pdf
3. Big Data: A New World of Opportunities. NESSI White Paper, December 2012. http://www.nessi-europe.com/Files/Private/NESSI_WhitePaper_BigData.pdf
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. Int. J Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
5. Condorcet's Theorem. http://en.wikipedia.org/wiki/Condorcet's_jury_theorem
6. Fan, J., Han, F., Liu, H.: Challenges of Big Data Analysis. Princeton University, Johns Hopkins University (2013). http://arxiv.org/pdf/1308.1479.pdf
7. Fan, J., Guo, S., Hao, N.: Variance estimation using refitted cross-validation in ultrahigh dimensional regression. J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) **74**(1), 37–65 (2012)
8. Fan, J., Fan, Y.: High dimensional classification using features annealed independence rules. Ann. Stat. **36**(6), 2605–2637 (2008)
9. Gorodetsky, V., Samoylov, V., Tushkanova, O.: Agent-based customer profile learning in 3G recommending systems. In: Proceedings of 9-th International Workshop on Agent and Data Mining Interaction (ADMI -2014) Associated with International Conference on Autonomous Agents and Multi-agent Systems (AAMAS -2014), Paris (2014)
10. Gorodetsky, V., Samoylov, V., Serebryakov, S.: Context–Driven Data and Information Fusion. In: Proceedings of International Conference on Information Fusion (Fusion 2012), pp. 1830–1837, Singapore (2012)
11. Gorodetsky, V., Samoylov, V., Serebryakov, S.: Ontology–based context–dependent personalization technology. In: Proceedings of WI/IAT/ACM International Conference, Associated Workshop "Web Personalization and Recommender Systems", Toronto (2010)
12. Gorodetsky, V., Serebryakov, S.: Methods and algorithms of collective recognition. Autom. Remote Control. **69**(11), 1821–1851 (2008)
13. Hall, P., Pittelkow, Y., Ghosh, M.: Theoretical measures of relative performance of classifiers for high dimensional data with small sample sizes. J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) **70**(1), 159–173 (2008)

14. IBM Big Data Success Stories. http://public.dhe.ibm.com/software/data/sw-library/big-data/ibm-big-data-success.pdf
15. InfoSphere BigInsights Enterprise Edition. http://www-03.ibm.com/software/products/ru/infobigienteedit/
16. IBM Business Analytics for Big Data – Overview. http://www-01.ibm.com/software/analytics/solutions/big-data/
17. InfoSphere Streams Technical Overview – Use Cases Big Data. http://www.slideshare.net/IBMInfoSphereUGFR/infosphere-streams-technical-overview-use-cases-big-data-jerome-chailloux
18. Kuncheva, L., Whitaker, C.: Measures of diversity in classifier ensembles. Mach. Learn. **51**, 181–207 (2003)
19. Li, J., Le, T.D., Liu, L., Liu, J., Jin, Z., Sun, B.: Mining causal association rules. In: Proceedings of International ICDM-2013 Workshop on Causal Discovery, Dallas, USA (2013)
20. NineSigma REQUEST #69987. https://www.ninesights.com/docs/DOC-8380
21. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1991)
22. Pearl, J. and Verma, T.S.: A Theory of Inferred Causation. In: Proc. Second International Conference on the Principles of Knowledge Representation and Reasoning, pp. 441–452 (1991)
23. Silverstein, C., Brin, S., Motwani, R.: Scalable techniques for mining causal structures. In: Proceedings of 24th VLDB Conference, New York, USA, pp 594–605 (1998)
24. Skormin, V.A., Gorodetski, V.I., Popyack, L.J.: Data mining technology for failure prognostic of avionics. IEEE Trans. Aerosp. Electron. Syst. **38**(2), 388–403 (2002)

Collecting University Rankings for Comparison Using Web Extraction and Entity Linking Techniques

Nick Bassiliades(✉)

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
nbasssili@csd.auth.gr

Abstract. University rankings are rankings of institutions in higher education, ordered by combinations of factors. Rankings are conducted by various organizations, such as news media, websites, governments, academics and private corporations. Due to huge financial and other interests, the rankings of universities worldwide recently received increasing attention. The rankings are based on different criteria and collect data in various ways. As a result, there is a large divergence in the specific rankings of different institutions. In order to compare rankings so that safe conclusions about their reliability are drawn, data from the sites of different such ranking lists must be collected. In this paper we present this first step for university ranking comparison, namely we discuss in detail how we have developed a Prolog application, called URank, that collects the data, by (a) extracting them from the various ranking list web sites using web data extraction techniques, (b) uniquely identifying the University entities within the above lists by linking them to the DBpedia linked open data set, and (c) constructing a combined data set by merging the individual ranking list data sets using their DBpedia URI as a primary key.

Keywords: University rankings · Web data extraction · Entity linking · Linked open data · Semantic web

1 Introduction and Problem Definition

University/College/Higher Education rankings are rankings of institutions in higher education, ordered by combinations of factors, such as measures of wealth, research excellence and/or influence, student choices, eventual success and/or demographics, on surveys, and others. Rankings are conducted by various organizations, such as news media, websites, governments, academics and private corporations. Rankings can evaluate institutions within a single country and/or region, or worldwide. In this paper we consider worldwide/global university rankings.

Due to huge financial and sometimes political interests, the rankings of universities worldwide recently received increasing attention. The rankings are based on different criteria and collect data in various ways. As a result, there is a large divergence in the specific rankings of different institutions. Therefore, rankings have produced much debate about their usefulness and accuracy. The expanding diversity in rating

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 23–46, 2014.
DOI: 10.1007/978-3-319-13206-8_2

methodologies and accompanying criticisms of each indicate the lack of consensus in the field. In order to compare rankings so that safe conclusions about their reliability are drawn, data from the sites of different such ranking lists must be collected and then statistically tested [1–4, 9, 10, 19, 21, 22].

In this paper we present the first step needed in order to compare university rankings, which is data collection. Actually, we have developed a Prolog application, called URank after "University Ranking", using SWI-Prolog [24], that (a) extracts data from the various ranking list web sites, (b) uniquely identifies the University entities within the above lists, and (c) constructs a combined data set that can be fed to the statistical comparison test. The actual comparison of rankings is beyond the scope of this paper; an initial report of a statistical comparative analysis of rankings (based on the data collection methodology described in this paper) can be found at [2].

Table 1 contains the University ranking lists we have used in this study, performed during academic year 2012–2013. In order to collect data from all those different ranking lists/sites several technical challenges exist. First of all is the acquisition of data, which are published in heterogeneous ways and formats. Usually, there are no downloadable and/or structured data, which in most cases must be extracted (scrapped) from HTML pages. To this end, web data extraction tools must be employed [7]. In our case, we have used DEiXTo [12], a powerful web data extraction tool based on the W3C DOM. It allows users to create highly accurate "extraction rules" (wrappers) that describe what pieces of data to scrape from a website. Actually, we have used only the GUI of DEiXTo, a friendly graphical user interface that is used to manage extraction rules (build, test, fine-tune, save and modify). Then, we have used the extraction rules built with DEiXTo GUI for the wrapper component of URank to extract data at run-time.

Data acquisition also "suffers" from the heterogeneity of the schemata of the data extracted from the various sites. In order to resolve this we have developed a small OWL ontology that describes ranked universities homogeneously and we have customized extraction rules (using Prolog) in order to map the extracted data (sometimes

Table 1. University Ranking Lists used in the paper.

Acronym	Full name	URL	Collected Universities
ARWU	Academic Ranking of World Universities	www.shanghairanking.com	500/500
Leiden	CWTS Leiden Ranking	www.leidenranking.com	500/750
QS	Quacquarelli Symonds	www.topuniversities.com	600/800
THE	Times Higher Education	www.timeshighereducation.co.uk	400/400
URAP	University Ranking by Academic Performance	www.urapcenter.org	750/2000
Webometrics	Ranking Web of Universities	www.webometrics.info	600/~12000

using tailored transformations) into this common schema. Actually, as a byproduct of our project, each extracted data set takes the form of RDF data that can be published into the Linked Open Data (LOD) cloud individually from the rest of the datasets.

The second and third challenges depend on each other. In order to merge different ranking lists into a single table (third challenge) one has to find a unique identification key for the Universities along the different ranking lists (second challenge). This is not a trivial task, since the names used in the different ranking lists are not always the same. For example, in the ARWU list (Table 1) the Imperial College[1] is mentioned as "The Imperial College of Science, Technology and Medicine", whereas in the QS list it is mentioned as "Imperial College London". In order to find a unique primary key for each list that can be safely used across datasets in order to merge them together, we should consider finding a unique immutable identifier for each University entity. We decided to consider DBpedia,[2] a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia offers the ability to ask sophisticated queries against Wikipedia and to link the different data sets on the Web to Wikipedia data. So, linking the entities extracted from the different ranking datasets to DBpedia could serve two goals: (a) linking the data extracted in the first step with a very well-known and rich linked open dataset, and (b) using the DBpedia ID (actually a URI) as a unique primary key across datasets to enable dataset merging.

Linking entities to DBpedia is not a trivial task either. DBpedia (and Wikipedia) contain crowd-sourced data, so they not always accurate or complete. For example, there might cases where a DBpedia entity that represents a University is not classified correctly under the University or Educational Institution class, but to a class higher in the hierarchy of the DBpedia ontology (e.g. *owl*:*Thing*). Furthermore, there might be synonym Universities in different places (e.g. Newcastle University[3] in the UK, University of Newcastle[4] in Australia) or there might be University mergers or splits along history, whose names still appear for historical reasons in Wikipedia and DBpedia (e.g. University of Paris[5] which split in 1970 into 13 Universities named very similarly some times as "University of Paris I, II, …").

In order to resolve all the above issues, general purpose entity linking software, such as DBpedia Spotlight [14] or SILK [23], cannot possibly have a 100 % accuracy, simply because domain-specific knowledge on University naming, geographical reasoning and temporal reasoning (to name a few), must be used additionally to disambiguate University entities in DBpedia. Even using domain-specific knowledge, sometimes the official DBpedia dataset does not contain up-to-date information because Wikipedia articles are constantly being revised, so when some pieces of information cannot be found at DBPedia, DBpedia Live[6] is used. Finally, when neither DBpedia

1 http://www3.imperial.ac.uk/
2 http://dbpedia.org/
3 http://www.ncl.ac.uk/
4 http://www.newcastle.edu.au/
5 http://en.wikipedia.org/wiki/University_of_Paris
6 http://wiki.dbpedia.org/DBpediaLive

nor DBpedia Live can provide a satisfactory disambiguation for an entity, URank uses Wikipedia text search (which proved to be better than DBpedia's text search) and web extraction techniques to find better candidate entities.

In the rest of the paper, we present the architecture and functionality of the URank system in Sect. 2, we report on the extensive evaluations we have performed on URank, and finally we conclude with a critical discussion on the ability to extend URank to become a general purpose tool, some thought for future work and a small comparison to relevant systems.

2 URank Architecture and Functionality

The architecture of the URank application is shown in Fig. 1. The main components of our system are: (a) the Web data extractor or *Entity Extractor*, that extracts the University entities from the ranking sites, (b) the *Entity Linker*, that links the extracted University entities with DBpedia entities, and (c) the *Entity Merger*, that generates a single entity for each University by merging the different datasets, using the DBpedia entity URI as a primary key. In the following subsections we present in detail each of these components.

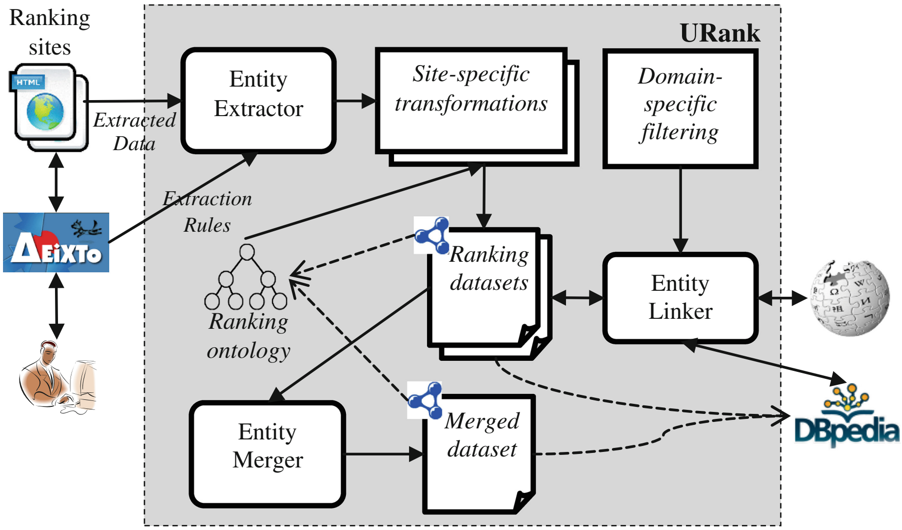


Fig. 1. URank architecture

2.1 Entity Extractor

The Entity Extractor is the component of URank that acquires needed data from University ranking sites. The Entity Extractor is driven by users who use the DeiXTo GUI in order to define site-specific extraction rules for each ranking list web site. Figure 2 shows an example of using DeiXTo for defining an extraction rule for the

ARWU site. More details about using DeiXTo are beyond the scope of this paper and can be found at DeiXTo site[7] and at reference [12]. What is important to notice is that the extraction rule (or pattern) defined using DeiXTo is exported in an XML file (Fig. 3). The contents of this file are fed to the Web data extractor component of URank which uses the XML, XPath and http libraries of SWI-Prolog to extract the data from the ranking sites. Although the interpretation of the XML DeiXTo extraction rules by our wrapper component involves a rather sophisticated algorithm, its detailed presentation is beyond the scope of this paper. For each site the name of the University, its global rank and its country is collected. Notice that countries are needed for name disambiguation purposes later, as already discussed in the introduction. Furthermore, the URL that contains details about the specific University is also extracted, in case the data transformation component needs to access it for disambiguation purposes.

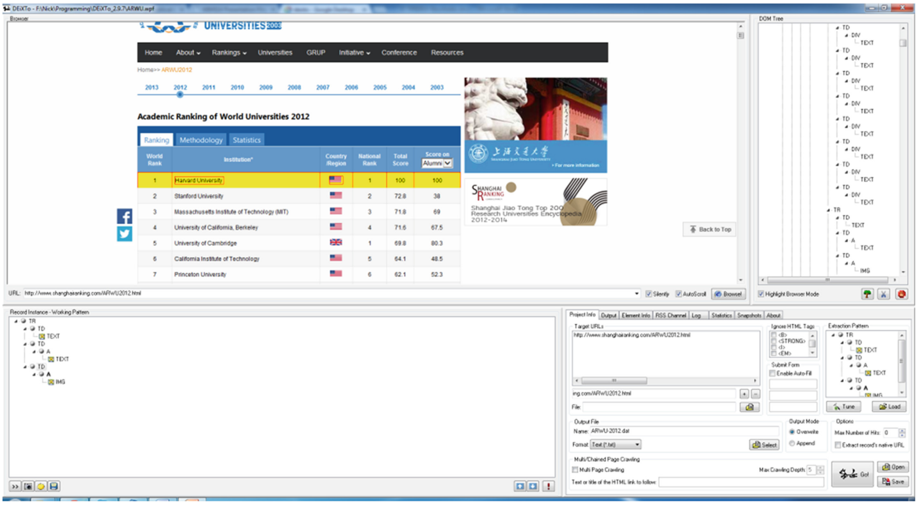


Fig. 2. DeiXTo GUI screenshot for defining extraction rule for the ARWU site

The extraction rules differ a lot, depending on the site. In the simplest case, such as ARWU (Table 1), all data are found in a single page. However, there are cases where data are found in several linked pages, such as Webometrics; therefore, web extraction must load and scrape several consecutive pages, navigating through a “next”-type link. In the case of THE list there is no “next” link, so the pages of all the ranges of ranking must be manually collected and fed to the extractor.

After the extractor completes the task of retrieving every piece of raw data that can be retrieved from the ranking web sites, site-specific transformations clear and homogenize the data in order to create the site-specific datasets in RDF. These transformations mostly deal with converting the retrieved country-related data into a proper

[7] http://deixto.com/

```
<!DOCTYPE Project SYSTEM "wpf.dtd">
<Project>
    <TargetUrls>
       <URL Address="http://www.shanghairanking.com/ARWU2012.html"/>
    </TargetUrls>
    <MultiplePage Enabled="false" ContainsText="" MaxCrawlDepth="5"/>
    <ExtractionPattern>
       <Node tag="TR" stateIndex="grayed" IsRoot="true">
          <Node tag="TD" stateIndex="grayed">
             <Node tag="TEXT" stateIndex="checked"/>
          </Node>
          <Node tag="TD" stateIndex="grayed">
             <Node tag="A" stateIndex="grayed">
                    <Node tag="TEXT" stateIndex="checked"/>
             </Node>
          </Node>
          <Node tag="TD" stateIndex="grayed">
             <Node tag="A" stateIndex="grayed">
                <Node tag="IMG" stateIndex="checked"/>
             </Node>
          </Node>
       </Node>
    </ExtractionPattern>
    <OutputFile Filename="ARWU-2012.dat" Format="TabDelimited"/>
</Project>
```

Fig. 3. DeiXTo extraction rule for the ARWU site

country name, common across the different ranking sites. For example, in ARWU the country information is retrieved as a URL that contains all the Universities of this specific country contained in the ARWU list.[8] In this case, specific string processing rules retrieve the name of the country. Other sites, such as Webometrics, do not have a link to country pages/profiles, but they just show the flag of the country, using a short country code in the image URL.[9] In this case, string processing isolates the country code and a transformation table derived from the ISO 3166 Country Codes standard[10] transforms it into a proper country name.

The rest of the site-specific transformations deal with clearance of the University names, such as removing extra spaces, transforming names from URL to ASCII encoding, removing trailing numbers from Webometrics entries when Universities maintain multiple web domains,[11] etc. Finally, in the case of Leiden the main ranking page used to contain abbreviated University names only, while full names could be found in the detailed pages of the Universities. Therefore, data transformation included additional web data extraction activities. In the current version of the Leiden ranking site full University names are included in the main ranking page as tooltips.

After extracted data are cleared and transformed the individual datasets for each ranking site can be constructed. These datasets are in RDF and can be published in the

[8] E.g. http://www.shanghairanking.com/World-University-Rankings-2012/USA.html

[9] E.g. http://www.webometrics.info/sites/default/files/logos/us.png

[10] http://www.iso.org/iso/country_codes.htm

[11] http://www.webometrics.info/en/node/36

LOD cloud. In order to have a common schema for all sites, we have developed a lightweight University ranking ontology which consists of two classes (Fig. 4): *RankingOrganization* and *RankedInstitution*. The former has six instances, representing the six ranking list/sites of Table 1 included in this study. The latter will have as many instances as per University entities extracted from each ranking site. Table 2 presents the properties for the two classes, while Fig. 5 shows the instance of the *RankingInstitution* class for the ARWU list. Notice the use of the *dc:title* property for the name of the ranking institution and *owl:sameAs* property for linking our datasets to the LOD cloud, i.e. the DBpedia entry for the ranking list. Instances for the *RankedInstitution* class will be shown later, after the entity linking with DBpedia entries is discussed.

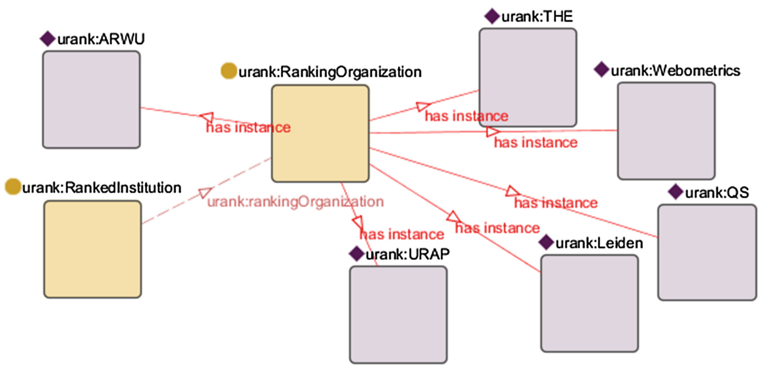


Fig. 4. The University Ranking Ontology and the 6 ranking list instances

Table 2. Properties of the University Ranking Ontology

Property	Domain	Range
hasURL	RankingOrganization	xs:anyURI
rankingOrganization	RankedInstitution	RankingOrganization
rankURL	RankedInstitution	xs:anyURI
institutionRegion	RankedInstitution	xs:string
rank	RankedInstitution	xs:int

```
Individual: urank:ARWU
Types: urank:RankingOrganization
Facts:
 dc:title "Academic Ranking of World Universities"^^xs:string
 urank:hasURL  "http://www.shanghairanking.com/"^^xs:anyURI
SameAs:
 dbpedia:Academic_Ranking_of_World_Universities
```

Fig. 5. The ARWU instance of *RankingOrganization* class

2.2 Entity Linker

Linking our dataset entities to DBpedia is performed as indicated by Algorithm 1, which is a non-formal high-level description of the main DBpedia matching algorithm. The algorithm consists of two main loops, for each ranking list and for each University entry retrieved from each list. Inside the second loop, there are a number of steps to retrieve matching DBpedia entries using 3 different approaches, explained later. At each step, if a satisfactory match is found the algorithm terminates immediately and returns that match for each University. Otherwise, candidate DBpedia matching entries are collected into a candidate set, scored according to our own scoring function and then the best scored candidate is returned as a match.

Algorithm 1. The basic matching algorithm.

1. **For each** Ranking list *R*
2.   **For each** University *U* **in** *R*
3.   Candidate Universities *Cand* = $\emptyset$
4.   Find *Top-N1* instances of *EducationalInstitution* class using DBpedia lookup service with keyword *U.name*. Each instance must adhere to domain-specific spatiotemporal restrictions.
5.   **If** *CheckMatchFound*(*Top-N1*) **then**
      Result = *CheckMatchFound*(*Top-N1*); **break**;
    else *Cand* = *Cand* $\cup$ *Top-N1*
6.   Find *Top-N2* instances of *EducationalInstitution* class using DBpedia SPARQL endpoint and a query template with list of words from *U.name*. Each instance must adhere to domain-specific spatiotemporal restrictions.
7.   **If** *CheckMatchFound*(*Top-N2*) **then**
      Result = *CheckMatchFound*(*Top-N2*); **break**;
    else *Cand* = *Cand* $\cup$ *Top-N2*
8.   Find *Top-N3* entries in Wikipedia using keyword search (keyword *U.name*)
9.   *Top-N3'* = $\emptyset$
10.   **For each** alternative University name *A* **in** *Top-N3* find the corresponding DBpedia entry *A'* and form *Top-N3'*
11.   Check if Wikipedia entry *A* corresponds to a DBPedia entry *A'* with URI transformation. *A'* must adhere to domain-specific spatiotemporal restrictions.
12.   **If** the above is true **then**
        Top-N3' = *Top-N3'* $\cup$ {*A'*}
      Else
        Repeat steps 4, 6 using alternative name *A*. Give priority to *EducationalInstitution* instances. In case of failure relax this requirement. For each alternative name *A*, a set of *Top-N1'* and *Top-N2'* entries will be returned.
        Top-N3' = *Top-N3'* $\cup$ *Top-N1'* $\cup$ *Top-N2'*
13.   **end for**
14.   **If** *CheckMatchFound*(*Top-N3'*) **then**
      Result = *CheckMatchFound*(*Top- N3'*); **break**;
    else *Cand* = *Cand* $\cup$ *Top-N3'*
15.   Score each *C* **in** *Cand* using the string distance between *U.name* and *C.name*, source of *C* and the method that *C* was obtained
16.   Sort *Cand* in descending score order
17.   *Result* = *Cand*[1]
18.   **end for**
19. **end for**

Table 3. Thresholds for satisfactory maching.

Steps	Threshold
4, 6	0.98
14 (*EducationalInstitution* instances)	0.97
14 (*owl*:*Thing* instances)	0.99
14 (search Wikipedia via Google)	1.00

The "satisfactory match" (*CheckMatchFound* function in Algorithm 1) for a DBpedia entry depends on the string distance between the name of the University extracted from the ranking list and the name of the matching DBpedia University entry. The distance threshold depends on the step of the algorithm. More specifically, in order to measure string distance we use the *isub/4* built-in function of SWI-Prolog, which is based on a string metric for ontology alignment [20]. This metric is more appropriate in our case than the Levenshtein distance metric, since it mainly concerns substring matching, which is appropriate for matching names of Universities. For example, recall the case for "Imperial College" from the introductory section. Table 3 shows these thresholds, which have been experimentally found and are very high to ensure termination only for almost absolutely certain matches.

The two main methods for retrieving DBpedia entities by matching extracted University names is (a) DBpedia lookup service[12] and (b) OpenLink Virtuoso built-in SPARQL endpoint, on the DBpedia host instance,[13] using a template query derived from the Faceted Browser, and Search & Find Service.[14] For the DBpedia lookup service the query template is:

```
http://lookup.dbpedia.org/api/search.asmx/KeywordSearch?
      QueryClass=<Class>&MaxHits=<Top-N1>&QueryString=<U.Name>
```

Notice that the above query returns results in an XML file, which is parsed using the same SWI-Prolog libraries as for extracting Universities from HTML files, above.

The query template for DBpedia SPARQL endpoint is as follows:

```
select ?u, ?n where {
     ?u rdf:type <Class> .
     ?u ?p ?v .
     ?v bif:contains <U.Name.Words> option (score ?sc) .
     ?u rdfs:label ?n .
     FILTER (lang(?n) = "en") }
order by desc (?sc*0.3+sql: rnk_scale(<LONG::IRI_RANK> (?u)))
limit <Top-N2>
```

In the above query, `?u` is the URI of the matched DBpedia entry and `?n` its name. The query retrieves the values `?v` of all properties `?p` of the University and searches

12 http://wiki.dbpedia.org/lookup/
13 http://dbpedia.org/sparql
14 http://dbpedia.org/fct/

them for words contained within the extracted University name (<*U.Name.Words*>) using Virtuoso's built-in *bif*:*contains* predicate.

In the above searches, the query class <*Class*> is *EducationalInstitution* for steps 4, 6 and *owl*:*Thing* for the relaxed search in step 12. Furthermore, the maximum number of hits <*Top-N1*> and <*Top-N2*> are 2 for steps 4, 6 and 4 for the relaxed search in step 12 and they have been established experimentally. Furthermore, in the case of step 12, at line 3 in the SPARQL query template property `?p` becomes *rdfs*:*label*; therefore, search concentrates only on the property that contains the name of the University. Notice that in step 12, DBpedia is searched using as a keyword the name of a Wikipedia-retrieved University, not the name of the originally-retrieved University.

During all DBpedia searches (steps 4, 6, 12), the retrieved instances are filtered according to spatiotemporal domain-specific constraints. Namely, the retrieved DBpedia University must be located in the same country as the University extracted from the site and it must also be still operating. The check for the latter is performed by checking if the property *dbpprop*:*closed* exists. Of course, this is not always the case for all closed/suspended Universities, such as the *dbpedia*:*University_of_Paris*, for example. When such information does not exist, then URank is susceptible to errors, unless a better match is found.

The check for location/country compatibility is not always easy, since DBpedia entries stem from Wikipedia articles and sometimes the infoboxes of these articles are not complete. For example, University DBpedia entries may not have a country-related property, but only City- or State-related information (USA and Spanish Universities, mainly). Therefore, spatial inclusion reasoning must be employed in this case, with additional SPARQL queries to find out in which country a city or State is located, etc. Furthermore, sometimes there are multiple entries for the same entity in DBpedia, similarly to Wikipedia. In this case, the original DBpedia search may not retrieve the entry with the country-related information. In this case redirection links are followed and the country-related search is repeated. Table 4 summarizes the properties used for retrieving the country of the University DBpedia entry.

To make things even more difficult, the retrieved country information may not be exactly the same as the country data extracted from the raking site. For example, the country information associated with *dbpedia*:*Harvard_University* is "U.S.", while the country data for this University from the ARWU list is "USA". So, there is a need

Table 4. DBpedia properties related to Location.

Location Information	DBpedia properties
Country	dbpedia-owl:country, dbpprop:country
State	dbpedia-owl:state, dbpprop:state
City	dbpedia-owl:city, dbpprop:city
Location	dbpedia-owl:location, dbpprop:location
{*Redirection to another instance*}	owl:sameAs, dbpedia-owl: wikiPageRedirects

to create a compatibility matrix for country names. This can be done only empirically/experimentally by collecting compatible names for some countries. The majority of country names though do not have such a synonymy problem. We do not include this matrix in the paper because some synonyms we came up with are not "politically correct" and may raise conflicts.

There are a few more subtle domain-dependent filtering criteria that must be taken into account, such as using Roman or Arabic numbers in University names (e.g. "University of Montpellier II" vs. "Montpellier 2 University") or using synonyms for the word University in other languages (e.g. "University of Freiburg" vs. "Universität Freiburg"). These are also crucial for achieving a 100 % precision and recall, but are too detailed heuristics to be presented here.

In case steps 4 and 6 of Algorithm 1 do not retrieve a high match, step 8 uses the keyword search engine of Wikipedia to retrieve Wikipedia articles as candidates for alternative (and possibly better) names for the retrieved Universities. For example, the ARWU list contains the entry "University of Paris Sud (Paris 11)". This does not return any result at DBpedia lookup service. Even if the string in parenthesis is stripped, because for the ARWU list it is considered a synonym (therefore, redundant), the DBpedia lookup service returns the entry *dbpedia*:*Paris-Sud_11_University*, with label "*Paris-Sud 11 University*". The string distance between the two names is 0.93, which is lower than the thresholds in Table 3. A query to Wikipedia returns as the best result the page with title "University of Paris-Sud"[15], which corresponds to the DBpedia entry *dbpedia*:*University_of_Paris-Sud*, with *rdfs*:*label* "*University of Paris-Sud*". The string distance now between the Wikipedia article title and the DBpedia entry label is exactly 1, therefore above the threshold of Table 3.

Of course, things are not simple here either. Wikipedia is asked to return <*Top-N3*> articles with *Top-N3* found experimentally to be 3 with the following query:

```
http://en.wikipedia.org/w/index.php?search=<U.name>&
        limit=<Top-N3>&go=Go
```

Sometimes Wikipedia just returns the most probable result, when its score exceeds some threshold. When this happens, the returned page is scraped to extract the article title and to check whether it involves indeed a University, located in the same country as the University extracted from the ranking site and still operational, namely using the same domain-dependent filters as in steps 4, 6. However, in the case of Wikipedia this is done by scraping the HTML of the returned page, and most specifically, the infobox and the categories box. For example, see Fig. 6 for an active public University located in Australia, the University of Sydney[16], and Fig. 7 for a suspended University. Furthermore, general pages such as "*Template*:…", "*List of Universities in* …", "*Higher education in* …", and similar ones, must be excluded, along with disambiguation pages. When a single result page does not exist, Wikipedia returns a list of results and the above checks are performed for the *Top-N3* results.

[15] http://en.wikipedia.org/wiki/University_of_Paris-Sud

[16] http://sydney.edu.au/

Finally, if the Wikipedia keyword search does not generate any alternative names due to all the above restrictions, Google search restricted in the Wikipedia domain is used as a last resort, using the query below and concentrating on the first result:

```
http://www.google.com/search?as_q=<U.name.words>&
      as_sitesearch=en.wikipedia.org
```

After step 8, list *Top-N3* contains alternative names/entries for the original University retrieved from the ranking site. These Wikipedia entries should lead to DBpedia entries, possibly giving better results than the original University name. This is the task of the loop in steps 10–13. There are 2 ways to map Wikipedia entries to DBpedia entries. The first one is direct and rewrites the Wikipedia URL to a DBpedia URI:

```
http://en.wikipedia.org/wiki/<Univ> →
           http://dbpedia.org/resource/<Univ>
```

However, before the DBpedia URI is considered a final match it must be verified for the same domain-specific restrictions already discussed above for steps 4 and 6. If the verification step succeeds then this DBpedia entry is added to list *Top-N3'* which contains candidate matching DBpedia entries. If the verification fails, then at step 12, which is the second way to map a Wikipedia entry to a DBpedia entry, the Wikipedia article titles are used as alternative University names that lead to new DBpedia searches with these alternative names, as in steps 4, 6. From this search, new candidate matching DBpedia entries are retrieved, which are added to list *Top-N3'*.

Notice that actually step 12 performs two searches (using both search methods): one strict with *EducationalInstitution* as the target class and one relaxed with *owl*:*Thing*

Fig. 6. Infobox and categories box for the University of Sydney Wikipedia entry (http://en.wikipedia.org/wiki/University_of_Sydney)

Fig. 7. Infobox for the University of Paris Wikipedia entry (http://en.wikipedia.org/wiki/University_of_Paris)

as the target class. The results for the two searches are scored differently, giving higher score to the stricter search, as it will be discussed later. The reason for this is that step 12 is the last chance of the algorithm to discover a match, so in case the strict match does not retrieve any DBpedia instances, the result of the relaxed search will cover for it. Outside the loop, at step 14, list *Top-N3'* is checked for immediate results, i.e. results that give a string distance above the threshold of Table 3.

The last important step of the matching algorithm is the scoring function for the candidate DBpedia entities collected into set *Cand*. Recall that entities in this list have string distances less than the thresholds of Table 3; otherwise, the algorithm would have stopped and returned a confirmed match. So, the purpose of step 15 is to score each candidate match using the string distance between the University name of the DBpedia entry and the name of the University exported from the ranking site or the alternative name retrieved from Wikipedia articles (in step 8). Furthermore, the scoring function takes into account (a) the source of the candidate match (original or alternative name from Wikipedia), (b) the method that the candidate match was obtained (DBpedia lookup service, SPARQL endpoint, and direct transformation of Wikipedia URL to DBpedia URI), and (c) if the search was strict or relaxed, concerning the target class. Table 5 summarizes the score additions that each of the above dimensions adds to the string distance metric, which is in the range between 0 and 1. For example, when a candidate match was obtained from the original University name retrieved from the ranking site using a strict search at the DBpedia lookup service (step 4) and the string distance of the candidate match from the University name is 0.92, the total score is 1000 + 200 + 10 + 0.92 = 1210.92. On the other hand, if a candidate match is coming from step 11 (Wikipedia search, direct URL/URI transformation) with a 0.95 string distance, the total score is 2000 + 200 + 20 + 0.95 = 2220.95. Notice that direct searches of step 11 and DBpedia searches in steps 4 and 6 are always strict.

Table 5. Score additions along various dimensions.

Dimension	Value	Score addition
Source	Original (ranking site)	1000
	Wikipedia/Google search	2000
Target class	Strict	200
	Relaxed	100
Query method	DBpedia lookup service	10
	SPARQL endpoint	10
	Direct URL/URI transformation	20

From Table 5 it is obvious that when steps 4, 6 fail to produce a confirmed match, then priority is given to candidate matches coming from Wikipedia retrieved alternative University names, since Wikipedia keyword search engine is better than DBpedia's free text search engine. Furthermore, strict searches are preferred to relaxed searches, for obvious reasons. Finally, direct URL/URI transformations (valid only in step 11) are preferred to DBpedia searches (step 13), since the latter may introduce more noise

```
<urank:RankedInstitution
     rdf:about="&urank;Imperial%20College%20London"
     dcterms:title="Imperial College London">
  <urank:institutionRegion rdf:datatype="&xsd;string">United Kingdom
     </urank:institutionRegion>
  <urank:rank rdf:datatype="&xsd;int">41</urank:rank>
  <urank:rankingOrganization rdf:resource="&urank;Leiden"/>
  <owl:sameAs rdf:resource="&dbpedia;Imperial_College_London"/>
</urank:RankedInstitution>
```

Fig. 8. Leiden dataset RDF entry for the "Imperial College London"

due to free text search. Notice that all these preferences have been experimentally established and evaluated.

Finally, after Algorithm 1 terminates, the RDF datasets for each ranking site are generated and saved permanently. Figure 8 shows the RDF code for the "Imperial College London" entry of the Leiden ranking site dataset. In the future, these datasets will be uploaded into an RDF triplestore with a public SPARQL endpoint.

2.3 Entity Linker

The Entity Merger component of URank takes as input the datasets of the 6 ranking sites and produces a single dataset that contains all the Universities with all the rankings from every ranking site contained in a single entity. For example, the merged dataset entry for the "Imperial College London" is shown in Fig. 9. The properties for the merged dataset are slightly different from the individual datasets. Specifically, there is no country-related information, since the purpose of the merged dataset is to statistically compare rank positions, and there is no direct link to the *RankingOrganization* instance, since each entry is ranked by multiple ranking organizations. Furthermore, there are 6 new properties, which hold the ranks of the individual ranking sites. All these are sub-properties of the *urank*:*rank* property and are added to the ontology. For example, in Fig. 9 the Imperial College entity has a *urank*:*rankTHE* property for the THE ranking list, a *urank*:*rankQS* property for the QS list, etc. The following piece of OWL code shows how the *urank*:*rankeTHE* property is defined:

```
<owl:DatatypeProperty rdf:ID="rankTHE">
    <rdfs:supbPropertyOf rdf:resource="#rank"/>
</owl:DatatypeProperty>
```

The merge of the datasets into a single one is performed with Algorithm 2. For each RDF graph *R* that holds the corresponding ranking dataset (step 1) and for each University instance in this dataset (step 2), a new University instance is created in the merged dataset graph *M* and the appropriate property values are copied (step 5). From the second iteration of the outer loop and onwards, it might be the case that the

```
<urank:RankedInstitution
                 rdf:about="&urank;Imperial%20College%20London"
                 dcterms:title="Imperial College London">
  <urank:rankARWU rdf:datatype="&xsd;int">24</urank:rankARWU>
  <urank:rankLeiden rdf:datatype="&xsd;int">54</urank:rankLeiden>
  <urank:rankQS rdf:datatype="&xsd;int">6</urank:rankQS>
  <urank:rankTHE rdf:datatype="&xsd;int">8</urank:rankTHE>
  <urank:rankURAP rdf:datatype="&xsd;int">14</urank:rankURAP>
  <urank:rankWebometrics rdf:datatype="&xsd;int">261
  </urank:rankWebometrics>
   <owl:sameAs rdf:resource="&dbpedia;Imperial_College_London"/>
</urank:RankedInstitution>
```

Fig. 9. Merged dataset RDF entry for the "Imperial College London"

University instance already exists in the merged dataset from previous iterations (step 3). In that case, only the corresponding rank property is copied (step 4).

Algorithm 2. The algorithm form merging the individual ranking datasets into one.

1. **For each** individual RDF graph *R* (or the corresponding ranking site)
2.   **For each** instance of *urank*:*RankedInstitution* *U* **in** *R*
3.   Check if there is an instance *U'* of *urank*:*RankedInstitution* in the merged dataset graph *M*, such that *U'*.*owl*:*sameAs* = *U*. *owl*:*sameAs*
4.   **If** yes, **then** *U'*.*urank*:*rank*<*R*> = *U*.*urank*:*rank*
5.   **If** no, **then**
            Create a new *U'* instance of *urank*:*RankedInstitution* with *U'*.*ID* = *U*.*ID*;
            U'.*dc*:*title* = *U*.*dc*:*title*;
            U'.*owl*:*sameAs* = *U*.*owl*:*sameAs*;
            U'.*urank*:*rank*<*R*> = *U*.*urank*:*rank*;
         end if
6.   **end for**
7. **end for**

Notice that the existence check is based on the value of the *owl*:*sameAs* property which is a link to the DBpedia entry for the University, discovered by Algorithm 1. So, it is important to ensure that all DBpedia URIs for the same University are exactly the same. Although this seems expected, it is not always the case. Sometimes there exist many Wikipedia articles for the same topic, which are redirected to a single Wikipedia page. This is also reflected to the corresponding DBpedia instances. For the same real-world entity there might be several DBpedia instances that re-direct to (possibly) a single DBpedia entry through the *dbpedia-owl*:*wikiPageRedirects* property. Thus, the entity linking process (of the previous sub-section) ensures that when a DBpedia URI is matched to a University name, a pointer-chasing algorithm ends up to the instance at the end of the chain of the re-direction links.

Sometimes there are more-than-one instances with the above property, i.e. they are at the end of different paths of the re-direction link sub-graph for the same real-world entity. This is checked by a transitive algorithm that follows the re-direction links until it finds instances that do not re-direct to another instance. In this case, URank selects the most “informative” one, namely the one with the most triples. Another problematic case is when this re-direction sub-graph is not acyclic, something that happens rarely and usually it is temporary until the next DBpedia update. Nevertheless, we catered for this case as well using a closed set search.

3 Evaluation

In order to evaluate URank, we have performed several experiments. First of all, we clarify that we have the correct answers (namely the correct DBpedia entries) for all the ranking sites, so we are able to evaluate and compare the effectiveness of each of the search mechanisms of Algorithm 1. These correct answers have been obtained manually by first running URank and then checking manually only the entries that did not have an absolute match (string distance 1.0). Then, we have conducted for each ranking site experiments turning on and off the following features/mechanisms of URank, in many combinations:

- DBpedia lookup service
- SPARQL endpoint query
- Domain-specific restrictions
- Wikipedia keyword search

For each experiment we count the correct answers *CA* (i.e. those entries that the retrieved URIs coincide with the correct URIs), the incorrect answers *IA* (i.e. those entries that the retrieved URIs do not coincide with the correct URIs), and the unanswered entries *UA* (i.e. the ones that the experiment did not manage to retrieve any URI). Notice that we assume (and it is true for the experiments we have conducted) that all Universities do have a Wikipedia /DBpedia entry. From the above measurements we calculate the precision, recall and F-measure metrics for the queries, using Eqs. (1) – (3), respectively.

$$precision = \frac{CA}{CA + IA} \tag{1}$$

$$recall = \frac{CA}{CA + UA} \tag{2}$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

Our first experiment measures the effectiveness of each query method, namely its purpose is to compare the DBpedia lookup service against the SPARQL endpoint query method using the template derived from the Faceted Browser, and Search & Find Service (see Footnote 14). Notice that the domain-specific restrictions and the Wikipedia keyword search are turned off. Results are shown in Table 6 for the DBpedia lookup service and Table 7 for the SPARQL endpoint query. Results clearly indicate

Table 6. Measurements for the DBPedia lookup service.

Ranking site	CA	UA	IA	Precision	Recall	F
ARWU	433	60	7	98,41 %	87,83 %	92,82 %
Leiden	472	24	4	99,16 %	95,16 %	97,12 %
QS	512	80	8	98,46 %	86,49 %	92,09 %
THE	382	12	6	98,45 %	96,95 %	97,70 %
URAP	627	112	11	98,28 %	84,84 %	91,07 %
Webometrics	521	65	14	97,38 %	88,91 %	92,95 %
Total/Average	2947	353	50	98,33 %	89,30 %	93,60 %

Table 7. Measurements for the SPARQL endpoint query.

Ranking site	CA	UA	IA	Precision	Recall	F
ARWU	398	25	77	83,79 %	94,09 %	88,64 %
Leiden	429	6	65	86,84 %	98,62 %	92,36 %
QS	520	23	57	90,12 %	95,76 %	92,86 %
THE	356	4	40	89,90 %	98,89 %	94,18 %
URAP	541	123	86	86,28 %	81,48 %	83,81 %
Webometrics	511	20	69	88,10 %	96,23 %	91,99 %
Total/Average	2755	201	394	87,49 %	93,20 %	90,25 %

the superiority of the DBpedia lookup service in terms of Precision, for all ranking sites, and the superiority of the SPARQL endpoint query in terms of recall, for almost all ranking sites, with the sole exception of URAP. This is due to the fact that the DBpedia lookup service is stricter than the SPARQL query; therefore, it returns less results but with a better chance of being correct. The F-measure value is superior for the DBpedia lookup service, with the exception of QS list.

The same conclusion is evident from Figs. 10, 11 and 12, where a graphical comparison between the two query methods is presented for the three metrics. Furthermore, in these figures we compare the performance of each of these query methods alone with their combination, namely we have conducted another set of measurements where both query methods are used in combination. Results show that precision is slightly worse, whereas recall and F-measure are better when the two query methods are combined. This happens because the DBpedia lookup service is stricter concerning its answers, whereas the SPARQL endpoint query method more relaxed, therefore it tends to return more answers with lower accuracy, so their combination exhibits the advantages of both worlds.

Our second experiment measures the effectiveness of the domain-specific restrictions. Figure 13 shows results for all the metrics and both query methods, with and without the domain-specific restrictions. As expected, restrictions increase the precision, since more restrictions mean more accurate results. The effect is evident for the SPARQL endpoint query, because there was a lot of room for improvement, while it is negligible for the DBpedia lookup service, since its precision was already high. The exact opposite behavior is observed for recall, which is slightly worst for the lookup

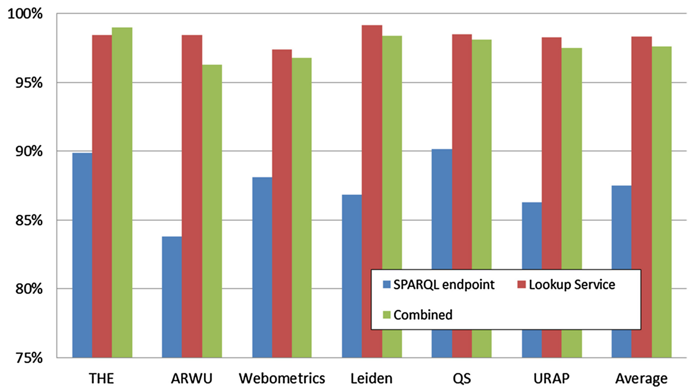


Fig. 10. Precision comparison for query methods

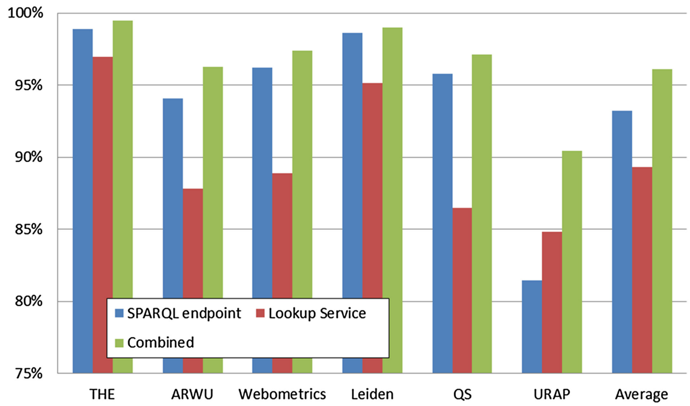


Fig. 11. Recall comparison for query methods

Fig. 12. F-measure comparison for query methods

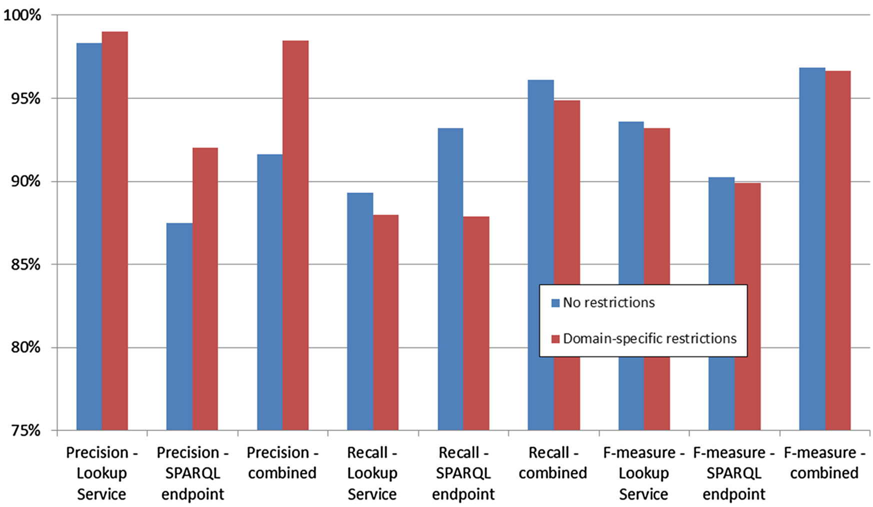


Fig. 13. Effectiveness of the domain-specific restrictions

service, but evidently worse for SPARQL query. This was expected, because more restricted queries mean fewer results. Overall, F-measure is slightly worse for both methods. The same behavior is observed for combining the two query methods.

Our last experiment measures the contribution of each of the query methods in retrieving the correct results. Table 8 shows how many correct results are due to which query method in the complete URank system. It is evident that the majority of the correct results are returned by the DBpedia lookup service (step 4), which is queried first. This choice was justified by the fact that the DBpedia lookup service has better performance (f-measure) than the SPARQL endpoint query method. The second best source of correct results is the direct URL/URI transformation (step 11) of the Wikipedia results returned after step 8. Notice that there are very few (actually 4) results that were returned after with Google search on Wikipedia. Also very few (only 2) are the correct results returned from step 12, making evident that the use of Wikipedia search

Table 8. Contribution of each query method to the result.

Ranking site	DBpedia Lookup	SPARQL endpoint	Wikipedia/Google	
			Direct URL/URI transformation	DBpedia Lookup/ SPARQL
ARWU	418	20	61	1
Leiden	468	9	23	0
QS	485	37	78	0
THE	375	11	14	0
URAP	611	21	118	0
Webometrics	505	33	61	1
TOTAL	2862	131	355	2

is a very competent complement of the DBpedia lookup service method. Finally, there are also correct results due to SPARQL endpoint query, but their overall contribution is very small (< 4 %). Figure 14 visualizes this comparison among methods.

Finally, Fig. 15 shows the contribution of each query method to the result of each of the experiments reported in this section. In this figure we can also compare the total correct results found by each of the tested stripped-down versions of URank, compared to the complete system. It can be concluded that the effect of using Wikipedia keyword search on top of either DBpedia lookup service or SPARQL endpoint query has almost the same result as having both of them (combined with Wikipedia). The actual results show a very small difference (1–5 less correct results, namely $\sim$ 0.1 %). This finding could be used to remove one of the steps 4 or 6, to increase the execution speed of URank.

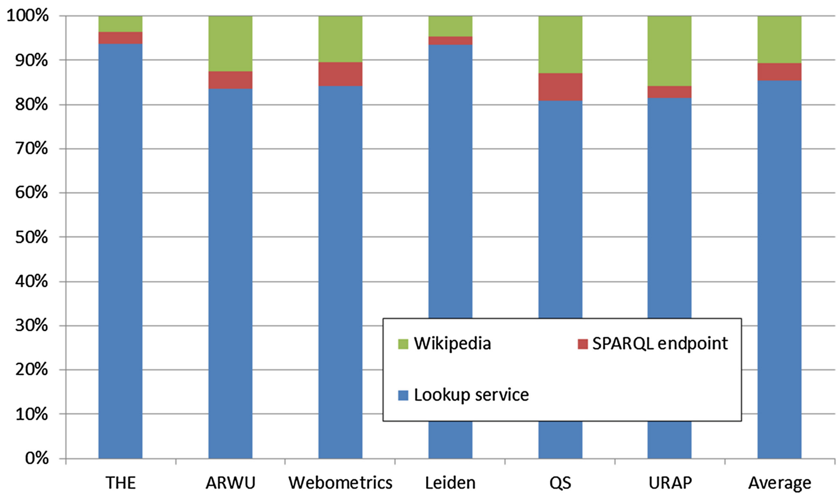


Fig. 14. Contribution of each query method to the result for the complete system

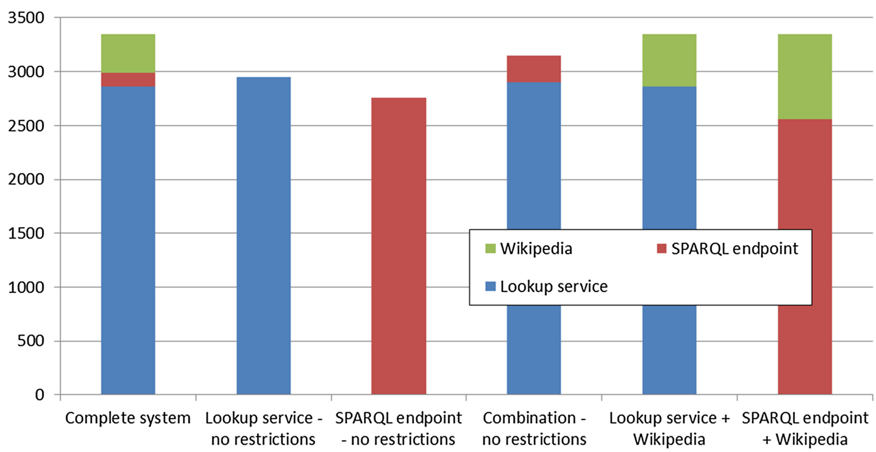


Fig. 15. Contribution of each query method to the result for each experiment

4 Concluding Discussion and Future Work

In this paper we have presented our effort to extract, link and merge University ranking datasets as Linked Open Data in the Semantic Web. This is a part of a larger project [2] that aims to statistically compare different University rankings in order to draw safe conclusions about their reliability. In order to collect the data found at different University ranking sites, we have developed, presented and evaluated a Prolog application, called URank, that (a) extracts data (University entities) from the various ranking list web sites, using the DeiXTo web extraction tool [12], (b) uniquely identifies the University entities within the above lists, by linking them to the corresponding DBpedia entities, and (c) constructs a combined data set that can be fed to the statistical comparison test, by merging the individual ranking datasets based on the discovered DBpedia link, as a unique primary key.

In order to develop the URank system several challenges had to be met. The first one has to do with the heterogeneity of the data formats and schemata of the ranking sites, as well as the different naming schemata for the Universities and the countries they are located. These challenges were met by (a) customized data extraction rules that were easily developed using the GUI of the DeiXTo system, (b) site-specific data transformations that were developed in Prolog, and (c) an ontology we developed for the individual ranking datasets.

The challenges concerning the unique identification of Universities using DBpedia required: (a) to use the appropriate querying method with the appropriate list of words in order to search for matching entities, and (b) to recognize when a correct match has been found. There are two main querying methods, the DBpedia lookup service and the SPARQL endpoint that uses a query template derived from the Openlink Virtuoso's Search & Find Service of DBpedia. Both have their own advantages and disadvantages and we decided to use them both complementary. In order to recognize if a correct match has been found we have used the SWI-Prolog's built-in string difference metric, which is tailored to ontology alignment [20] and worked well in our case.

The main problems encountered during the development and testing of URank had mainly to do with the domain-specific nature of our search and the fact that DBpedia is a crowd-sourced knowledge base; therefore, neither correct nor complete. In order to overcome these problems we have developed a few domain-specific filters/restrictions that the retrieved entities should obey in order to be considered correct matches, on top of the string distance metric. This increased the precision of the retrieved instances, as the evaluation phase has shown. However, domain-specific restrictions lowered the recall rate, namely less correct results were retrieved because the required information was simply not present in DBpedia entries.

A major boost to recall was given by a second phase that involved searching for the University name into Wikipedia using either the Wikipedia's search engine or even Google. This was needed because the way University names are found and retrieved from the ranking sites, usually differ a lot from their formal or usual names. This search provided Wikipedia articles for the Universities where a more suitable University name (the title of the Wikipedia article) could be found. Wikipedia entries directly correspond to DBpedia entries through a trivial URL/URI transformation, so in this way better matches can be found in the vast majority of cases. Of course, extracting information from Wikipedia pages involves heuristic (thus error prone) web extraction techniques, because pages are edited by humans. However, the use of Wikipedia, along with the other querying methods and restrictions, increased our precision and recall to 100 %.

Finally, a very crucial part of the entity linker is the scoring mechanism for the various DBpedia entries retrieved using all the above methods. Our scoring mechanism gives priority to (a) candidate entries retrieved through either the DBpedia lookup service or a SPARQL endpoint query, with a name that is very close (≥ 98 %) to the original University name, then to (b) candidate entries retrieved from direct URL/URI transformations of Wikipedia retrieved entries, and finally to (c) candidate entries retrieved by the combination of (b) and (a). The results of our evaluation show that 89.3 % of the correct answers are given by method (a), 10.5 % are given by method (b), and only 0.2 % by method (c).

The outcome of the entity linker is the individual RDF datasets for each ranking site, linked to the DBpedia LOD dataset through the *owl:sameAs* property. In order to generate a single merged dataset so that each university includes its rank at every ranking site, a unique primary key for each University must exist across the individual datasets. However, due to the different naming schemes of the ranking sites and due to the multiple DBpedia/Wikipedia entries for the same University, the query methods of the entity linker might end up to different DBpedia entries, for the same University. Usually these DBpedia entries re-direct to a single one (following Wikipedia redirections), so following this re-direction graph gives to URank a stable primary key mechanism. The outcome of the entity merger is an RDF dataset that can be used to compare the rankings of the Universities across the different ranking sites/lists.

Looking critically at URank, we can draw the conclusion that the system has achieved its purpose, namely to correctly collect and merge the University rankings for further statistical processing. However, one may consider the possibility of making the system independent from the domain of University rankings in the future, thus being able to collect and merge into a single table various lists of similar nature found in the Web. This needs a lot of improvements, mainly in the code itself, but also to the architecture of the system. Currently, domain-independent and domain-dependent features of the system are not so well separated in the code, despite their clear distinction in Fig. 1. This is because there are various heuristics (domain-dependent features) used in the code that need to be tightly integrated with the domain-dependent features, such as querying DBpedia and/or Wikipedia. However, this separation needs further exploration in order to develop a re-usable across domains system.

Other improvements for the system would be (a) a GUI (currently the text-based interface of Prolog is used), including a tighter integration with the DeiXTo GUI, (b) a more efficient and general purpose web data extractor, exploiting the full range capabilities of DeiXTo extraction rules, and (c) integration with an RDF triplestore, such as Openlink Virtuoso[17] or Sesame[18] [11], so that the generated datasets to persist and be shared on the LOD cloud.

[17] http://virtuoso.openlinksw.com/

[18] http://www.openrdf.org/

Compared to general purpose tools for entity extraction, such as [5, 13, 16, 17], URank does neither have to detect names, since the user does that using DeiXTo through its extraction rules, nor to classify the names by the type of entity (class) they refer to, since it is a domain-dependent application and extracted names are known to be University names. Therefore, URank cannot be characterized as an entity extraction application, nor it can be compared to such software.

On the other hand, URank can be considered as an entity linking software, since its purpose is to determine the identity of entities mentioned in a list of named entities, which is distinct from entity extraction/recognition in that it identifies not the occurrence of names (nor classifies them), but their reference. There are plenty of general-purpose entity-linking tools, such as [6, 14, 15, 18, 25], which usually use one knowledge base target to link entities, such as DBpedia, Wikipedia, or YAGO2 [8] using various techniques for matching (e.g. lexical) and context disambiguation (relatedness, similarity, coherence). URank instead is a domain-specific tool that focuses on a specific target type of the linked entity and takes advantage of domain-specific knowledge in order to improve precision and recall of the system. One of our main future aims is to compare the performance of URank to one or more of the above general purpose tools for entity linking. Initial experimentations with DBpedia Spotlight [14] have resulted in ~ 86 % F-measure, which is not as good as 100 % that URank achieved due to its domain-specific tweaking.

References

1. Aguillo, I.F., Bar-llan, J., Levene, M.: Priego, J.L.O: Comparing University Rankings. Scientometrics **85**(1), 243–256 (2010)
2. Angelis, L., Bassiliades, N., Manolopoulos, Y.: Evaluation of University International Rankings (in Greek). In: Proceedings of the Conference on Quality Assurance and Quality Management: Governance and Good Practices, Thessaloniki (2012)
3. Buela-Casal, G., Gutiérrez-Martínez, O., Bermúdez-Sánchez, M.P., Vadillo-Muñoz, O.: Comparative study of international academic rankings of universities. Scientometrics **71**, 349–365 (2007)
4. Cheng, Y., Liu, N.C.: Examining major rankings according to the Berlin principles. High. Educ. Europe **33**(2–3), 201–208 (2008)
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: 40th Anniversary Meeting of the Association for Computational Linguistics (2002)
6. Ferragina, P., Scaiella, U.: TAGME: On-the-fly annotation of short text fragments (by wikipedia entities). In: 19th ACM International Conference on Information and Knowledge Management (CIKM ‘10), pp. 1625–1628. ACM (2010)
7. Ferrara, E., de Meo, P., Fiumara, G., Baumgartner, R.: Web Data Extraction, Applications and Techniques: A Survey. CoRR. arXiv:1207.0246 [cs.IR] (2012)
8. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: a spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194**, 28–61 (2013)
9. Huang, M.-H.: A comparison of three major academic rankings for world universities: from a research evaluation perspective. J. Libr. Inf. Stud. **9**(1), 1–25 (2011)

10. Ioannidis, J., Patsopoulos, N., Kavvoura, F., Tatsioni, A., Evangelou, E., Kouri, I., Contopoulos-Ioannidis, D., Liberopoulos, G.: International ranking systems for universities and institutions: a critical appraisal. BMC Med. **5**(1), 30 (2007)
11. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: a generic architecture for storing and querying RDF and RDF schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
12. Kokkoras, F., Ntonas, K., Bassiliades, N.: DEiXTo: a web data extraction suite. In: 6th Balkan Conference in Informatics (BCI-2013), pp. 9–12. ACM, Thessaloniki (2013)
13. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
14. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: 7th International Conference on Semantic Systems (I-Semantics 2011), pp. 1–8. ACM, Graz (2011)
15. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: 17th ACM Conference on Information and Knowledge Management (CIKM '08), pp. 509–518. ACM (2008)
16. Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning multilingual named entity recognition from wikipedia. Artif. Intell. **194**, 151–175 (2013)
17. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: 13th Conference on Computational Natural Language Learning (CoNLL '09), pp. 147–155. Association for Computational Linguistics, Stroudsburg (2009)
18. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT '11), vol. 1, pp. 1375–1384. Association for Computational Linguistics, Stroudsburg (2011)
19. Rauhvargers, A.: EUA Report on Rankings 2011. Global University Rankings and their Impact. European University Association, Brussels (2011)
20. Stoilos, G., Stamou, G., Kollias, S.D.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
21. Stolz, I., Hendel, D.D., Horn, A.S.: Ranking of rankings: benchmarking twenty-five higher education ranking Systems in Europe. High. Educ. **60**(5), 507–528 (2010)
22. Taylor, P., Braddock, R.: International university ranking systems and the idea of university excellence. J. High. Educ. Policy Manage. **29**(3), 245–260 (2007)
23. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and Maintaining Links on the Web of Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 650–665. Springer, Heidelberg (2009)
24. Wielemaker, J., Schrijvers, T., Triska, M., Lager, T.: SWI-Prolog. Theory Pract. Logic Program. – Prolog Syst. 12(1-2), 67–96 (2012)
25. Yosef, M.A., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: AIDA: an online tool for accurate disambiguation of named entities in text and tables. In: Proceedings of the VLDB Endowment, vol. 4(12), pp. 1450–1453 (2011)

Frameworks and Tools

Soft Computing Algorithm for Arithmetic Multiplication of Fuzzy Sets Based on Universal Analytic Models

Yuriy Kondratenko[1(✉)] and Volodymyr Kondratenko[2]

[1] Department of Intelligent Information Systems,
Petro Mohyla Black Sea State University,
68-th Desantnykiv Str. 10, Mykolaiv 54003, Ukraine
y_kondratenko@rambler.ru
[2] Department of Mathematical and Statistical Sciences,
University of Colorado Denver, Denver, USA
volodymyr.kondratenko@email.ucdenver.edu

Abstract. This paper deals with the investigations of increasing dependability of computing operations for fuzzy numbers with triangular and bell-shape membership functions (MFs). Special attention is paid to the synthesis of analytic models of the MFs for the results of fuzzy arithmetic operations. New analytical models of the result's MFs with the description of synthesis procedures for the multiplication operation with triangular fuzzy numbers in R^+ and R are presented in a universal style. The general analytic models for determination of α-cuts parameters (direct and inverse approaches) for result fuzzy sets are given. Modeling results confirm the efficiency of the proposed models and fuzzy arithmetic algorithms for fuzzy information processing.

Keywords: Fuzzy number · Arithmetic operations · Multiplication · Direct model · Inverse model

1 Introduction

While performing the tasks of organizational control there always happen such situations when the original conditions of decision making are not clearly defined and characterized by insufficient awareness of the person who makes decisions, particularly in conflict situations or under extreme conditions. For mathematical formalization of processes and systems of this class there appeared a need to create a new mathematical approach. This approach is a theory of fuzzy sets developed by professor Zadeh [19]. Since the theory of fuzzy sets appeared, the specialists have had a great interest in it in terms of practical applications of mathematical methods in all fields of science and technology. The scientists around the world are aware of fundamental theoretical developments in the theory of fuzzy sets and fuzzy logic [2, 3, 11, 15, 17, 18].

Fuzzy set theory has a special notion of membership function [12, 15, 19] that exists in the interval [0,1]. Each element x of the fuzzy set, for example set $\underset{\sim}{A}$, corresponds to a specific value of the membership function $\mu_{\underset{\sim}{A}}(x) \in [0, 1]$. Thus, fuzzy set

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 49–77, 2014.
DOI: 10.1007/978-3-319-13206-8_3

$\underset{\sim}{A}$ that is specified on the basis of the universal set E, is called [19] the set of pairs $(x, \mu_{\underset{\sim}{A}}(x))$, where $x \in E$, $\mu_{\underset{\sim}{A}}(x) \in [0, 1]$.

Fuzzy sets and fuzzy logic are used for tasks of decision making and control in uncertainty, in particular for problems of routes and trajectory optimization [8]. The solution of the problems causes the necessity of fulfilling the operations of fuzzy arithmetic, in particular operations with fuzzy sets including addition, subtraction, multiplication and division.

Inverse models of resulting membership functions that are based on using α-cuts [4] do not always provide high performance of computing operations and often lead to complications in solving control problems in real time. Thus, the development of generalized analytic models, based on the direct approach that allow to formalize fuzzy arithmetic operations to improve their operating speed and accuracy is an important direction of research that is associated with increased dependability [5] of intelligent systems.

One of the most difficult fuzzy arithmetic operations in terms of its mathematical formalization is an operation of multiplication.

Computational algorithms for the operations of multiplication on the basis of using α-cuts of the relevant fuzzy sets [4, 5, 9] (inverse approach) have high computational complexity, as it is performed in turn for all α- levels ($\alpha_i \in [0,\ 1]$, $i = 0,\ 1,\ 2,\ \ldots,\ r$, $\alpha_0 = 0$, $\alpha_r = 1$) with the step of discreteness $\Delta\alpha$, which value, taking into consideration that $\alpha_{i+1} = \alpha_i + \Delta\alpha$, significantly affects the accuracy and operating speed of the performance of computational procedures [5]. Therefore, α- cuts of the fuzzy set $\underset{\sim}{A} \in R$ is ordinary (in terms of conditions $\mu_{\underset{\sim}{A}}(x) \geq \alpha$) subset that contains elements $x \in R$ whose degree of membership to a set $\underset{\sim}{A}$ is not less than value α, that is $A_\alpha = \{x \,|\, \mu_{\underset{\sim}{A}}(x) \geq \alpha\}$, $\alpha \in [0, 1]$.

Subsets A_α та B_α that determine the appropriate α-cuts of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ can be written as follows: $A_\alpha = [a_1(\alpha),\ a_2(\alpha)]$, $B_\alpha = [b_1(\alpha),\ b_2(\alpha)]$, where $\alpha \in [0,\ 1]$, $\underset{\sim}{A}, \underset{\sim}{B} \in R^+$, and arithmetic operation of multiplication can be written as [4, 5, 9, 12, 15, 19]

$$\begin{aligned} A_\alpha(\cdot)B_\alpha &= [a_1(\alpha),\ a_2(\alpha)](\cdot)[b_1(\alpha),\ b_2(\alpha)] \\ &= [a_1(\alpha)b_1(\alpha),\ a_2(\alpha)b_2(\alpha)]. \end{aligned} \tag{1}$$

In addition to calculations based on the mentioned above α-cuts [4, 5] for implementation of fuzzy arithmetic operations where computational algorithms are often used that are realized through the use of max-min or min-max convolutions [4, 14] that in some cases leads to increased complexity and reduced operating speed of performance or to the moment of obtaining the resulting MFs that do not meet the requirements of convexity and normality of fuzzy sets.

The aim of this work is a synthesis of analytical models of resulting MFs. Their use in fuzzy arithmetic will give the opportunity to significantly reduce the volume, complexity and accuracy, and to improve their operating speed. Then a more detailed analysis of the properties of arithmetic operation of multiplication of fuzzy triangular numbers will be given. These properties are the most common while using the theory of fuzzy sets for designing control systems, decision making support systems and intelligence expert systems. This fuzzy triangular number is called fuzzy number $\underset{\sim}{A}$ whose MF $\mu_{\underset{\sim}{A}}(x)$ is of triangular shape and mathematical presentation of triangular fuzzy number has the form

$$\underset{\sim}{A} = (a_1, a_0, a_2),$$

where $\mu_{\underset{\sim}{A}}(a_1) = 0$; $\mu_{\underset{\sim}{A}}(a_0) = 1$; $\mu_{\underset{\sim}{A}}(a_2) = 0$.

Generalized model A_α, synthesized on the basis of inverse approach, and direct model in a form of a triangular membership function $\mu_{\underset{\sim}{A}}(x)$ of triangular fuzzy number $\underset{\sim}{A}$ are determined by the appropriate relevant dependencies (2) and (3):

$$A_\alpha = [a_1(\alpha),\ a_2(\alpha)] = [a_1 + \alpha(a_0 - a_1),\ a_2 - \alpha(a_2 - a_0)\,], \tag{2}$$

$$\mu_{\underset{\sim}{A}}(x) = \begin{cases} 0, \forall(x \le a_1) \cup (x \ge a_2) \\ (x - a_1)/(a_0 - a_1), \forall(a_1 < x \le a_0) \\ (a_2 - x)/(a_2 - a_0), \forall(a_0 < x < a_2) \end{cases}. \tag{3}$$

2 Universal Analytical Models for Multiplication of Triangular Fuzzy Numbers in R^+

2.1 Synthesis of Inverse Model for Multiplication of Two Fuzzy Numbers

We shall illustrate the methods of forming inverse $C_\alpha = [c_1(\alpha), c_2(\alpha)]$ and direct $\mu_C(x)$ generalized analytical models of resulting MF for the operation of fuzzy triangular numbers multiplication $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$.

Firstly we shall form the inverse generalized model $A_\alpha = [a_1(\alpha), a_2(\alpha)]$ for a given (Fig. 1a) triangular fuzzy number $\underset{\sim}{A} = (a_1, a_0, a_2)$ in the set of non-negative real numbers R^+ [4, 5, 6, 9, 10] for the case $(a_1 < a_0 < a_2)$.

Let us analyze the left branch of the triangular fuzzy number $\underset{\sim}{A}$ for α- cut on the basis (2).

It is possible to write

$$\alpha = (a_1(\alpha) - a_1)/(a_0 - a_1),$$

where one can define

$$a_1(\alpha) = a_1 + (a_0 - a_1)\alpha,$$

where $a_1 = a_1(0) \ge 0$, $a_0 - a_1 > 0$, as $a_0 > a_1$, since $\underset{\sim}{A} \in R_0^+$.

Let us introduce the designations: $K_1 = a_0 - a_1$; $K_2 = a_1$, where $K_1 > 0$ and $K_2 \geq 0$, regarding this it can be written as:

$$a_1(\alpha) = K_2 + K_1\alpha.$$

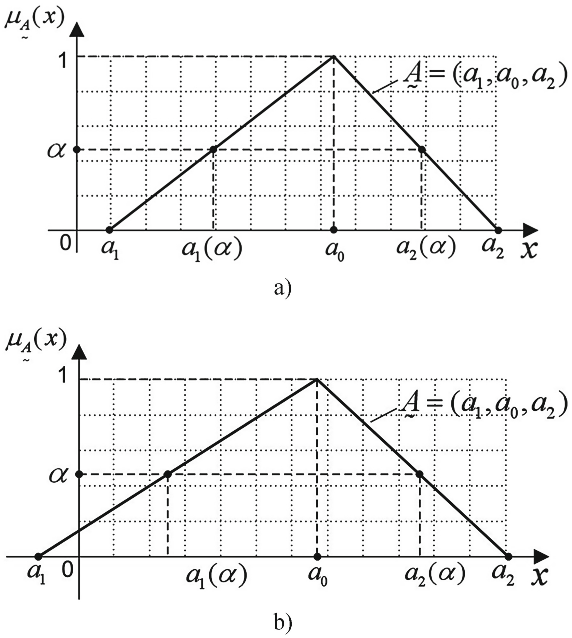


Fig. 1. Triangular Fuzzy Number $\underset{\sim}{A}$: (a) $\underset{\sim}{A} \in R^+$; (b) $\underset{\sim}{A} \in R$

Let us similarly analyze the right branch of a fuzzy number $\underset{\sim}{A}$:

$$\alpha = (a_2(\alpha) - a_2)/(a_0 - a_2),$$

$$a_2(\alpha) = a_2 + (a_0 - a_2)\alpha.$$

Having marked $K_3 = a_0 - a_2$, $K_4 = a_2$ ($K_3 < 0$, because $a_2 = a_2(0) > a_0$; $k_4 > 0$, as $\underset{\sim}{A} \in R_0^+$), we shall receive

$$a_2(\alpha) = K_4 + K_3\alpha.$$

However, as there is inequality $a_2 > |a_0 - a_2|$, then $K_4 > |K_3|$, respectively.

The α- cut for the left and right branches of the triangular fuzzy number $\underset{\sim}{A}$ can be represented as follows (taking into account coefficients K_i, $i = 1\ldots4 : K_1 > 0; K_2 > 0$; $K_3 < 0$; $K_4 > 0$; $K_4 > |K_3|$):

$$A_\alpha = [a_1(\alpha), a_2(\alpha)] = [K_2 + K_1\alpha,\ K_4 + K_3\alpha]. \tag{4}$$

Corresponding $\alpha-$ cut for triangular fuzzy number $\underset{\sim}{B} = (b_1, b_0, b_2)$ has the form

$$B_\alpha = [b_1(\alpha), b_2(\alpha)]$$

Having marked for the case $(b_1 < b_0 < b_2)$:

$$S_1 = b_0 - b_1, S_2 = b_1, S_3 = b_0 - b_2, S_4 = b_2,$$

we shall receive the modified $\alpha-$ cut B_α (taking into account the coefficients $S_i,\ i = 1\ldots4:\ S_1 > 0;\ S_2 \geq 0;\ S_3 < 0;\ S_4 > 0;\ S_4 > |S_3|$):

$$B_\alpha = [b_1(\alpha),\ b_2(\alpha)] = [S_2 + S_1\alpha,\ S_4 + S_3\alpha]. \tag{5}$$

Based on (4) and (5) we shall receive an inverse model for $\alpha-$ cut of a fuzzy set $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$:

$$\begin{aligned} C_\alpha = A_\alpha(\cdot)B_\alpha &= [a_1(\alpha), a_2(\alpha)](\cdot)[b_1(\alpha), b_2(\alpha)] \\ &= [a_1(\alpha)b_1(\alpha), a_2(\alpha)b_2(\alpha)] \\ &= [(K_2 + K_1\alpha)(S_2 + S_1\alpha),\ (K_4 + K_3\alpha)(S_4 + S_3\alpha)] \\ &= \left[K_1S_1\alpha^2 + (K_1S_2 + K_2S_1)\alpha + K_2S_2,\ K_3S_3\alpha^2 + (K_3S_4 + K_4S_3)\alpha + K_4S_4\right] \\ &= [C_1(\alpha), C_2(\alpha)]. \end{aligned} \tag{6}$$

2.2 Synthesis of Direct Model for Multiplication of Two Fuzzy Numbers

Left branch of resulting MF. For the direct model $\mu_{\underset{\sim}{C}}(x)$ on the basis of proposed approach we shall consider in a more detailed way a constituent $C_1(\alpha)$ for $\alpha-$ cut (6) of the resulting fuzzy set $\underset{\sim}{C}$, formed by the operation of multiplying $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$:

$$C_1(\alpha) = K_1S_1\alpha^2 + (K_1S_2 + K_2S_1)\alpha + K_2S_2.$$

The solution of the equation

$$K_1S_1\alpha^2 + (K_1S_2 + K_2S_1)\alpha + (K_2S_2 - C_1(\alpha)) = 0$$

will have the following roots

$$\alpha_{1,2} = \left(-(K_1S_2 + K_2S_1) \pm \sqrt{(K_1S_2 - K_2S_1)^2 + 4K_1S_1C_1(\alpha)}\right)/(2K_1S). \tag{7}$$

Let us analyze the roots α_1 and α_2 according to (7), which can be written as

$$\alpha_{1,2} = \left(-V_1 \pm \sqrt{Q_1}\right)/W_1 = -V_1/W_1 \pm \sqrt{Q_1}/W_1,$$

taking into consideration the following signs:

$$V_1 = K_1S_2 + K_2S_1,$$

$$Q_1 = (K_1S_2 - K_2S_1)^2 + 4K_1S_1C_1(\alpha),$$

$$W_1 = 2K_1S_1.$$

To form a direct model of the resulting nonlinear membership function $\mu_{\underset{\sim}{c}}(x)$ it is necessary to check the performance of the condition $\alpha_{1,2} \in [0;1]$.

Let us consider in details [8] the components V_1, Q_1, W_1 :

(a) $V_1 = K_1S_2 + K_2S_1$, taking into account correlations $K_1 > 0$, $K_2 > 0$, $S_1 > 0$, $S_2 > 0$ where there inequality $V_1 > 0$ takes place and respectively inequality $-V_1 < 0$;
(b) $Q_1 = (K_1S_2 - K_2S_1)^2 + 4K_1S_1C_1(\alpha) : C_1(\alpha) > 0$ as $\underset{\sim}{A}$, $\underset{\sim}{B} \in R_0^+$ because $K_1 > 0$, $S_1 > 0$, then always $Q_1 > 0$;
(c) $W_1 = 2K_1S_1$, as $K_1 > 0, S_1 > 0$, then $W_1 > 0$.

As $-V_1/W_1 < 0$, and the roots $\alpha_{1,2}$ have requirements $\alpha_{1,2} \in [0;1]$, then the root that satisfies the given condition under $V_1 > 0$, $Q_1 > 0$, $W_1 > 0$ will be

$$\alpha_1 = \left(-V_1 + \sqrt{Q_1}\right)/W_1,$$

that is

$$\alpha_1 = \left(-(K_1S_2 + K_2S_1) + \sqrt{(K_1S_2 - K_2S_1)^2 + 4K_1S_1C_1(\alpha)}\right)/(2K_1S_1). \tag{8}$$

The root

$$\alpha_2 = \left(-V_1 - \sqrt{Q_1}\right)/W_1$$

does not satisfy the condition $\alpha_2 \in [0;1]$, as there is always a condition $\alpha_2 < 0$.

The transition from inverse to direct approach [4, 5, 6, 8] shows that x is a parameter of the function $\alpha = f(C_1(\alpha))$, that is $\alpha = f(x)$, where $x = C_1(\alpha) \in [c_1, c_0]$, and then (8) can be represented as

$$\alpha_1 = \left(-(K_1S_2 + K_2S_1) + \sqrt{(K_1S_2 - K_2S_1)^2 + 4K_1S_1x}\right)/(2K_1S_1). \tag{9}$$

Right branch of resulting MF. Let us consider in detail the second component of the inverse model (6)

$$C_2(\alpha) = K_3S_3\alpha^2 + (K_3S_4 + K_4S_3)\alpha + K_4S_4.$$

The solution of the corresponding equation

$$K_3S_3\alpha^2 + (K_3S_4 + K_4S_3)\alpha + (K_4S_4 - C_2(\alpha)) = 0$$

will be the roots

$$\alpha_{3,4} = \left(-(K_3S_4 + K_4S_3) \pm \sqrt{(K_3S_4 - K_4S_3)^2 + 4K_3S_3C_2(\alpha)}\right)/(2K_3S_3).$$

The analysis of roots α_3 and α_4 shows, that the root $\alpha_3 > 1$ does not satisfy the condition

$$\alpha_3 \in [0; 1]$$

and therefore the only acceptable root will be the root α_4

The transition from inverse to direct approach [4, 5, 6, 8] allows to transform $\alpha = f(C_2(\alpha))$ into $\alpha = f(x)$, $x = C_2(\alpha) \in [c_0, c_2]$, and thus

$$\alpha_4 = \left(-(K_3S_4 + K_4S_3) - \sqrt{(K_3S_4 - K_4S_3)^2 + 4K_3S_3x}\right)/(2K_3S_3). \qquad (10)$$

The direct model of resulting MF and modeling results. The analysis of the roots α_1 and α_4 allows to make a conclusion that nonlinear dependence $\alpha = f(x)$ under intervals $x \in [c_1, c_0], x \in [c_0, c_2]$ is one-valued function.

The investigations allow to form a direct analytical model for one-valued nonlinear resulting membership function $\mu_{\underset{\sim}{C}}(x)$ of fuzzy set $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ that is formed by multiplying triangular fuzzy numbers $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in R^+:

$$\mu_{\underset{\sim}{C}}(x) = \begin{cases} 0, \\ \quad \text{for } (\forall x \le K_2S_2) \bigcup (\forall x \ge K_4S_4) \\ \frac{-(K_1S_2 + K_2S_1) + \sqrt{(K_1S_2 - K_2S_1)^2 + 4K_1S_1x}}{2K_1S_1}, \\ \quad \text{for } \forall x \in [K_2S_2, K_1S_1 + K_1S_2 + K_2S_1 + K_2S_2] \\ \frac{-(K_3S_4 + K_4S_3) - \sqrt{(K_3S_4 - K_4S_3)^2 + 4K_3S_3x}}{2K_3S_3}, \\ \quad \text{for } \forall x \in [K_3S_3 + K_3S_4 + K_4S_3 + K_4S_4, K_4S_4] \\ 1, \\ \quad \text{for } \forall x = c_0 = K_1S_1 + K_1S_2 + K_2S_1 + K_2S_2 \end{cases}, \qquad (11)$$

where $c_0 = K_1S_1 + K_1S_2 + K_2S_1 + K_2S_2 = K_3S_3 + K_3S_4 + K_4S_3 + K_4S_4$.

The given direct approach allows to form nonlinear resulting membership function $\mu_{\underset{\sim}{C}}(x)$ on the basis of known coefficients $K_i,\ S_i(i = 1..4)$ of fuzzy numbers $\underset{\sim}{A}$ (4) and $\underset{\sim}{B}$ (5) in R^+.

By substituting previously set marks

$$K_1 = a_0 - a_1, K_2 = a_1, K_3 = a_0 - a_2, K_4 = a_2,$$

$$S_1 = b_0 - b_1, S_2 = b_1, S_3 = b_0 - b_2, S_4 = b_2,$$

we shall get (Table 1) a direct model $\mu_{\underset{\sim}{C}}(x)$ that is realized directly on the basis of parameters $(a_0,\ a_1,\ a_2)$, $(b_0,\ b_1,\ b_2)$ of triangular fuzzy numbers $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in R^+.

The chart of the corresponding resulting membership function $\mu_{\underset{\sim}{C}}(x)$ under the realizing the operation of multiplication of triangular fuzzy numbers

$$\underset{\sim}{A} = (5,\ 7,\ 12)$$

and

$$\underset{\sim}{B} = (2,\ 9,\ 14)$$

using a developed direct model, presented in Table 1, shown in Fig. 2.

Table 1. Analytical model $\mu_{\underset{\sim}{C}}(x)$: direct approach.

Resulting membership function $\mu_{\underset{\sim}{C}}(x) =$
$\forall (x < a_1b_1 \cup x > a_2b_2), \qquad = 0$
$\forall x \in [a_1b_1, a_0b_0),$ $= \frac{-[(a_0-a_1)b_1+a_1(b_0-b_1)]+\sqrt{[(a_0-a_1)b_1-a_1(b_0-b_1)]^2+4(a_0-a_1)(b_0-b_1)x}}{2(a_0-a_1)(b_0-b_1)}$
$\forall x = a_0b_0, = 1$
$\forall x \in (a_0b_0, a_2b_2], = \frac{-[(a_0-a_2)b_2+a_2(b_0-b_2)]-\sqrt{((a_0-a_2)b_2-a_2(b_0-b_2))^2+4(a_0-a_2)(b_0-b_2)x}}{2(a_0-a_2)(b_0-b_2)}$

3 Direct and Inverse Analytical Models of Fuzzy Sets with Bell-Shape Membership Functions

Let's consider the class of bell-shape membership functions with following direct model [14, 15], for example for fuzzy set $\underset{\sim}{A}$,

$$\mu_{\underset{\sim}{A}}(x) = 1 / \left(1 + \left(\frac{x - b}{c}\right)^2\right) \tag{12}$$

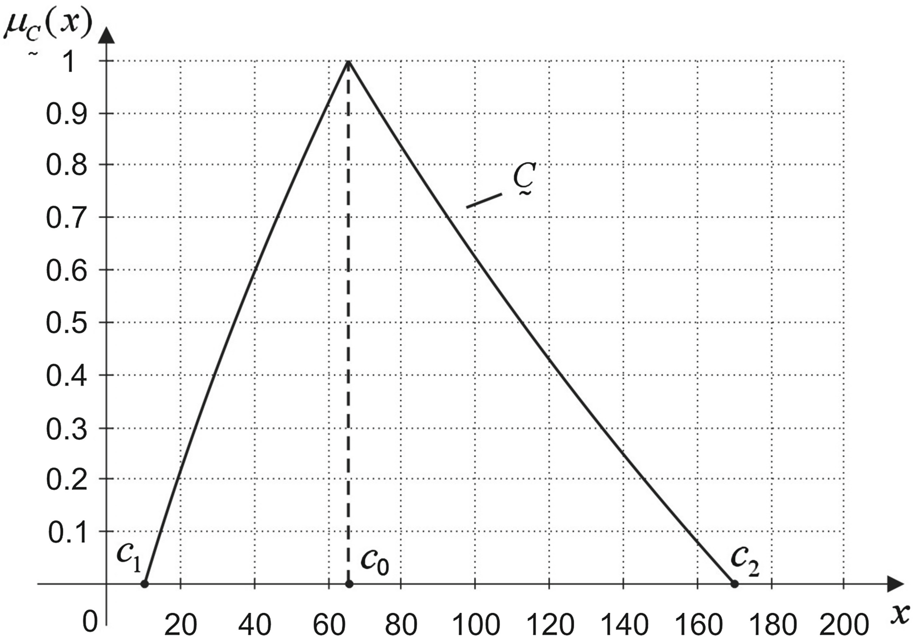


Fig. 2. Implementation of the direct model $\mu_{\tilde{C}}(x)$

and a synthesis procedure for analytic models of membership functions as results of arithmetic operations implementation. We will consider the cases when parameters of bell-shape MFs in model (12) are positive ($b > 0, c > 0$) and the approach is based on using α-cut models.

First of all, we form the inverse (horizontal) model of fuzzy set $\tilde{A}$

$$A_\alpha = [a_1(\alpha), a_2(\alpha)], \alpha \in [0,\ 1],$$

taking into account the equivalence relation between notations [4] in direct and inverse models and substituting the next parameters $\mu_{\tilde{A}}(x) = \alpha,\ \ x = a_{1,2}(\alpha)$ in direct model (12):

$$\alpha = 1/\left(1 + \left[\frac{a_{1,2}(\alpha) - b}{c}\right]^2\right), \tag{13}$$

where $a_{1,2}(\alpha) \in \{a_1(\alpha), a_2(\alpha)\}$.

At the next step it is necessary to find the roots of the Eq. (13) based on the following transformations (14)–(16)

$$\alpha = \frac{c^2}{c^2 + \left[a_{1,2}(\alpha) - b\right]^2}, \tag{14}$$

$$\alpha = \frac{c^2}{c^2 + a_{1,2}^2(\alpha) - 2ba_{1,2}(\alpha) + b^2}, \tag{15}$$

$$c^2 = c^2\alpha + a_{1,2}^2(\alpha)\alpha - 2ba_{1,2}(\alpha)\alpha + b^2\alpha. \tag{16}$$

As result we can obtain the square Eq. (17) corresponding to parameters $a_{1,2}(\alpha)$

$$\alpha a_{1,2}^2(\alpha) - 2\alpha b a_{1,2}(\alpha) + \left(b^2\alpha + c^2\alpha - c^2\right) = 0. \tag{17}$$

The roots of (17) can be calculated as (18)

$$a_{1,2}(\alpha) = \frac{2\alpha b \pm \sqrt{4\alpha^2 b^2 - 4\alpha(b^2\alpha + c^2\alpha - c^2)}}{2\alpha} \tag{18}$$

or as (19)

$$a_{1,2}(\alpha) = \frac{2\alpha b \pm \sqrt{4\alpha c^2 - 4\alpha^2 c^2}}{2\alpha} \tag{19}$$

The formula (19) can be transformed to (20)

$$a_{1,2}(\alpha) = b \pm c\sqrt{\frac{1}{\alpha} - 1}, \tag{20}$$

that means that roots of Eq. (17) can be finally calculated as:

$$a_1(\alpha) = b - c\sqrt{\frac{1}{\alpha} - 1} \tag{21}$$

and

$$a_2(\alpha) = b + c\sqrt{\frac{1}{\alpha} - 1}. \tag{22}$$

In this case the inverse (horizontal) model (17) of fuzzy set $\underset{\sim}{A}$ can be written in the following form:

$$A_\alpha = [a_1(\alpha), a_2(\alpha)] = \left[b - c\sqrt{\frac{1}{\alpha} - 1}, b + c\sqrt{\frac{1}{\alpha} - 1}\right]. \tag{23}$$

4 Universal Analytical Models for Multiplication of Bell-Shape Fuzzy Numbers in R^+

4.1 Synthesis of Inverse Analytical Model for Resulting Fuzzy Set

For analytic models synthesis of fuzzy sets which can result in fuzzy sets after fuzzy arithmetic operations implementation let's consider two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ with membership functions of type (12), which are presented by models of α-cuts [4]:

$$A_\alpha = [a_1(\alpha), a_2(\alpha)] \tag{24}$$

$$B_\alpha = [b_1(\alpha), b_2(\alpha)]. \tag{25}$$

Using the designations $b = p_1, c = s_1$ for model A_α in (24) and $b = p_2, c = s_2$ for model B_α in (25) and using the general horizontal model (23) we can determine:

$$A_\alpha = [a_1(\alpha), a_2(\alpha)] = \left[p_1 - s_1\sqrt{\frac{1}{\alpha} - 1}, p_1 + s_1\sqrt{\frac{1}{\alpha} - 1}\right], \tag{26}$$

$$B_\alpha = [b_1(\alpha), b_2(\alpha)] = \left[p_2 - s_2\sqrt{\frac{1}{\alpha} - 1}, p_2 + s_2\sqrt{\frac{1}{\alpha} - 1}\right]. \tag{27}$$

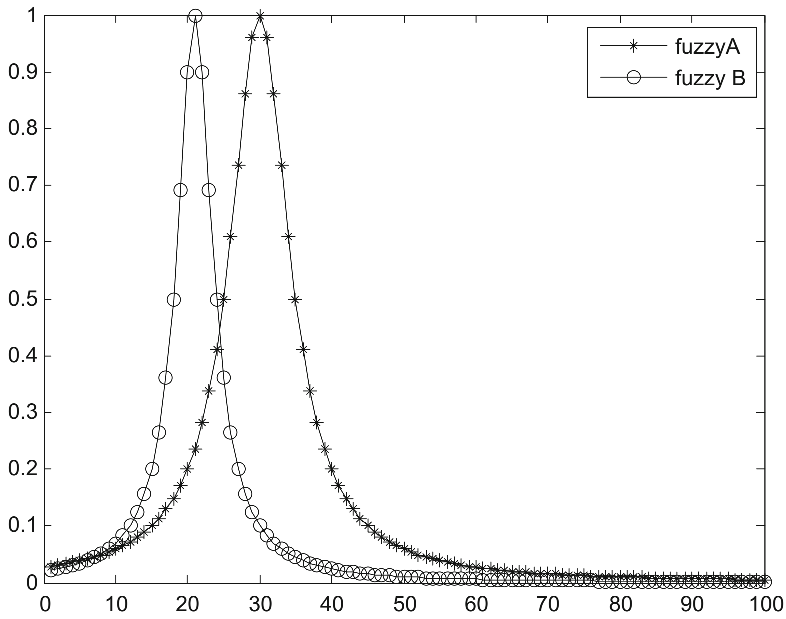


Fig. 3. Fuzzy sets $\underset{\sim}{A}, \underset{\sim}{B}$ with parameters of bell-shape membership functions: $p1 = 30, p2 = 21$, $s1 = 5, s2 = 3$

Let's consider the arithmetic operation of multiplication $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ for two fuzzy numbers $\underset{\sim}{A} \subset R^+$ and $\underset{\sim}{B} \subset R^+$ with bell-shape membership functions (12) according to the algorithm used in [2, 7, 11]:

$$\begin{aligned} C_\alpha = A_\alpha(\cdot)B_\alpha &= [a_1(\alpha)(\cdot)b_1(\alpha), a_2(\alpha)(\cdot)b_2(\alpha)] \\ &= [c_1(\alpha), c_2(\alpha)]. \end{aligned} \tag{28}$$

Substituting the horizontal models (26) and (27) into (28) we can form the inverse model of resulting membership function for implementation of multiplication operation $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ for fuzzy sets $\underset{\sim}{A} \subset R^+$ and $\underset{\sim}{B} \subset R^+$:

$$\begin{aligned} C_\alpha = A_\alpha(\cdot)B_\alpha &= \begin{bmatrix} \left(p_1 - s_1\sqrt{\frac{1}{\alpha} - 1}\right)(\cdot)\left(p_2 - s_2\sqrt{\frac{1}{\alpha} - 1}\right), \\ \left(p_1 + s_1\sqrt{\frac{1}{\alpha} - 1}\right)(\cdot)\left(p_2 + s_2\sqrt{\frac{1}{\alpha} - 1}\right) \end{bmatrix} \\ &= \begin{bmatrix} p_1p_2 - \sqrt{\frac{1}{\alpha} - 1}(p_1s_2 + s_1p_2) + s_1s_2\left(\frac{1}{\alpha} - 1\right), \\ p_1p_2 + \sqrt{\frac{1}{\alpha} - 1}(p_1s_2 + s_1p_2) + s_1s_2\left(\frac{1}{\alpha} - 1\right) \end{bmatrix} \\ &= [c_1(\alpha), c_2(\alpha)]. \end{aligned} \tag{29}$$

The resulting fuzzy set, which was created on the basis of model (29) after multiplication $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ of two fuzzy sets (Fig. 3), is represented in Fig. 4.

4.2 Synthesis of Direct Model for Resulting MF

Let's introduce new designations:

$$k_1 = p_1p_2, k_2 = p_1s_2 + s_1p_2, k_3 = s_1s_2,$$

where $k_1 > 0$, $k_2 >$, $k_3 > 0$, as $b > 0$, $c > 0$ in (12) and correspondently, $p_i > 0$, $s_i > 0, (i = 1, 2)$ in (26) and (27).

In this case we can represent resulting membership function (29) for implementation of multiplication operation as following

$$\begin{aligned} C_\alpha &= \left[k_1 - k_2\sqrt{\frac{1}{\alpha} - 1} + k_3\left(\frac{1}{\alpha} - 1\right), k_1 + k_2\sqrt{\frac{1}{\alpha} - 1} + k_3\left(\frac{1}{\alpha} - 1\right)\right] \\ &= [c_1(\alpha), c_2(\alpha)]. \end{aligned} \tag{30}$$

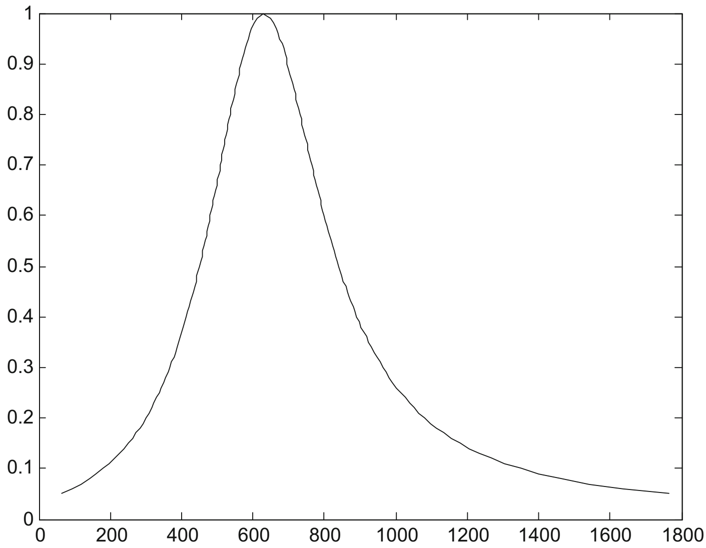


Fig. 4. Resulting fuzzy set for multiplication $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ based on the inverse model (29) with $\Delta\alpha = 0,01$

Investigation of the left branch and its properties for resulting MF. Let's consider in more details a left branch of Eq. (30)

$$c_1(\alpha) = k_1 - k_2\sqrt{\frac{1}{\alpha} - 1} + k_3\left(\frac{1}{\alpha} - 1\right), \tag{31}$$

which may be transformed to the form

$$k_1 + k_3\left(\frac{1}{\alpha} - 1\right) - c_1(\alpha) = k_2\sqrt{\frac{1}{\alpha} - 1}. \tag{32}$$

Square the left and right sides of the Eq. (32)

$$\left(k_1 + k_3\left(\frac{1}{\alpha} - 1\right) - c_1(\alpha)\right)^2 = \left(k_2\sqrt{\frac{1}{\alpha} - 1}\right)^2. \tag{33}$$

At the next step it is necessary to simplify the Eq. (33) based on the following transformations (34)–(36)

$$\begin{aligned} k_1^2 + k_3^2\left(\frac{1}{\alpha} - 1\right)^2 + c_1^2(\alpha) + 2k_1k_3\left(\frac{1}{\alpha} - 1\right) - 2k_1c_1(\alpha) - 2k_3\left(\frac{1}{\alpha} - 1\right)c_1(\alpha) \\ = k_2^2\left(\frac{1}{\alpha} - 1\right) \end{aligned} \tag{34}$$

$$\begin{aligned} & k_1^2 + \frac{k_3^2}{\alpha^2} - 2\frac{k_3^2}{\alpha} + k_3^2 + c_1^2(\alpha) + \frac{2k_1k_3}{\alpha} - 2k_1k_3 \\ & \quad - 2k_1c_1(\alpha) - \frac{2k_3}{\alpha}c_1(\alpha) + 2k_3c_1(\alpha) - \frac{k_2^2}{\alpha} + k_2^2 = 0 \end{aligned}, \tag{35}$$

$$\begin{aligned} & \frac{k_3^2}{\alpha^2} + \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)\frac{1}{\alpha} + k_1^2 + k_3^2 + c_1^2(\alpha) \\ & \quad - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2 = 0 \end{aligned}. \tag{36}$$

Multiplying the left and right sides of the Eq. (36) for α^2

$$\begin{aligned} & (k_1^2 + k_3^2 + c_1^2(\alpha) - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2)\alpha^2 \\ & \quad + \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)\alpha + k_3^2 = 0 \end{aligned} \tag{37}$$

and find the roots $\alpha_{1,2}$ of Eq. (37)

$$\alpha_{1,2} = \frac{-\left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right) \pm \sqrt{\begin{aligned} & \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)^2 - 4\left(k_1^2 + k_3^2 + \right. \\ & \left. + c_1^2(\alpha) - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2\right)k_3^2 \end{aligned}}}{2\left(k_1^2 + k_3^2 + c_1^2(\alpha) - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2\right)}$$

or

$$\alpha_{1,2} = \left(-v_1 \pm \sqrt{q_1}\right)/w_1, \tag{38}$$

where indicated:

$$v_1 = 2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2, \tag{39}$$

$$\begin{aligned} q_1 & = \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)^2 \\ & \quad - 4\left(k_1^2 + k_3^2 + c_1^2(\alpha) - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2\right)k_3^2 \\ & = \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)^2 - 4\left((k_3 + c_1(\alpha) - k_1)^2 + k_2^2\right)k_3^2 \end{aligned}, \tag{40}$$

$$\begin{aligned} w_1 & = 2\left(k_1^2 + k_3^2 + c_1^2(\alpha) - 2k_1k_3 - 2k_1c_1(\alpha) + 2k_3c_1(\alpha) + k_2^2\right) \\ & = 2\left((k_3 + c_1(\alpha) - k_1)^2 + k_2^2\right) \end{aligned}. \tag{41}$$

To form a direct model of the resulting nonlinear membership function $\mu_{\underset{\sim}{c}}(x)$ it is necessary to check the performance of the condition $\alpha_{1,2} \in [0; 1]$.

Let us consider in details the components (39)–(41) in roots (38), in particular, v_1, q_1, w_1:

(a) for value v_1 the two cases should be considered:
  (1) $v_1 > 0$, if $2k_1k_3 > \left(2k_3^2 + 2k_3c_1(\alpha) + k_2^2\right)$ and
  (2) $v_1 < 0$, if $2k_1k_3 < \left(2k_3^2 + 2k_3c_1(\alpha) + k_2^2\right)$;

(b) the value q_1 theoretically may be positive or negative, but taking into account that it is necessary to calculate $\sqrt{q_1}$ and roots (38) are real numbers, then q_1 should be only positive number: $q_1 > 0$;

(c) analyzing (41) we may conclude that for any cases $w_1 > 0$.

According to (38) we can rewrite

$$\alpha_{1,2} = -v_1/w_1 \pm \sqrt{q_1}/w_1. \tag{42}$$

Let's analyze the situation when $v_1 > 0$ and in this case we have inequality

$$k_1 - k_3 > c_1(\alpha) + \frac{k_2^2}{2k_3}. \tag{43}$$

The Eq. (40) can be represented as

$$q_1 = z - y, \tag{44}$$

where components z and y may be transformed in the following way:

$$\begin{aligned} z &= \left(2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2\right)^2 \\ &= 4k_1^2k_3^2 - 8k_1k_3^3 - 8k_1k_3^2c_1(\alpha) - 4k_1k_2^2k_3 + 4k_3^4 \\ &\quad + 8k_3^3c_1(\alpha) + 4k_2^2k_3^2 + 4k_3^2c_1^2(\alpha) + 4k_2^2k_3c_1(\alpha) + k_2^4 \end{aligned}; \tag{45}$$

$$\begin{aligned} y &= 4\left((k_3 + c_1(\alpha) - k_1)^2 + k_2^2\right)k_3^2 \\ &= 4k_1^2k_3^2 + 4k_3^4 + 4k_3^2c_1^2(\alpha) - 8k_1k_3^3 - 8k_1k_3^2c_1(\alpha) + 8k_3^3c_1(\alpha) + 4k_2^2k_3^2 \end{aligned}. \tag{46}$$

Substituting (45) and (46) into (44) we can obtain

$$q_1 = 4k_2^2k_3\left(c_1(\alpha) + \frac{k_2^2}{4k_3} - k_1\right). \tag{47}$$

Analyzing (43) and taking into account that $k_3 > 0$ it is possible to conclude

$$k_1 > c_1(\alpha) + \frac{k_2^2}{2k_3}$$

and respectively

$$c_1(\alpha) + \frac{k_2^2}{2k_3} - k_1 < 0. \tag{48}$$

The condition (48) means that in this case according to (47) we have

$$q_1 < 0. \tag{49}$$

Taking into account that q_1 should be always a positive number it is necessary to say the considered situation $(v_1 > 0)$ is uncorrected.

Let's analyze the second situation when $v_1 < 0$ for roots (42) and correlation between v_1 and w_1:

$$-v_1 = -2k_1k_3 + 2k_3^2 + 2k_3c_1(\alpha) + k_2^2, \tag{50}$$

$$\begin{aligned} w_1 &= 2\Big((k_3 + c_1(\alpha) - k_1)^2 + k_2^2\Big) \\ &= 2(c_1(\alpha) - k_1)^2 + 2k_3^2 - 4k_1k_3 + 4k_3c_1(\alpha) + 2k_2^2 \end{aligned}. \tag{51}$$

First of all, it is necessary to check the condition

$$w_1 > -v_1$$

or

$$w_1 + v_1 > 0. \tag{52}$$

Substituting (50) and (51) into (52) we will obtain

$$\begin{aligned} w_1 + v_1 &= 2(c_1(\alpha) - k_1)^2 + 2k_3^2 - 4k_1k_3 + 4k_3c_1(\alpha) + 2k_2^2 \\ &\quad + 2k_1k_3 - 2k_3^2 - 2k_3c_1(\alpha) - k_2^2 \\ &= 2(c_1(\alpha) - k_1)^2 - 2k_1k_3 + 2k_3c_1(\alpha) + k_2^2 > 0 \end{aligned}. \tag{53}$$

Taking into account that

$$k_1 > 0, k_3 > 0, k_2 > 0, c_1(\alpha) > 0$$

in the inequality (53), such components will be always positive

$$2(c_1(\alpha) - k_1)^2 > 0;\ 2k_3c_1(\alpha) > 0;\ k_2^2 > 0;\ 2k_1k_3 > 0$$

and only one component is negative

$$-2k_1k_3 < 0.$$

Let's compare one of the positive components in (53)

$$k_2^2 = (p_1s_2 + p_2s_1)^2 = (p_1s_2)^2 + 2p_1s_2p_2s_1 + (p_2s_1)^2$$

with negative component in (53)

$$-2k_1k_3 = -2(p_1p_2)(s_1s_2) = -2p_1p_2s_1s_2$$

in the following way

$$k_2^2 - 2k_1k_3 = (p_1s_2)^2 + 2p_1s_2p_2s_1 + (p_2s_1)^2 - 2p_1p_2s_1s_2 = (p_1s_2)^2 + (p_2s_1)^2 > 0. \quad (54)$$

The inequality (54) confirms that in (53)

$$2(c_1(\alpha) - k_1)^2 + 2k_3c_1(\alpha) + k_2^2 > 2k_1k_3$$

and inequalities and correlations

$$w_1 + v_1 > 0,\ w_1 > -v_1 \quad (55)$$

are correct.
In this case we can see that

$$-v_1/w_1 < 1 \quad (56)$$

and at the same time taking into account that $v_1 < 0$, $w_1 > 0$ we have

$$-v_1/w_1 > 0 \quad (57)$$

and finally

$$0 < (-v_1/w_1) < 1,\ -v_1/w_1 \in [0,\ 1].$$

Analyzing (39)–(41) we may represent q_1 and $\sqrt{q_1}$ in following forms

$$q_1 = v_1^2 - 2k_3^2w_1,$$

$$\sqrt{q_1} = \sqrt{v_1^2 - 2k_3^2w_1}.$$

Comparing $\sqrt{q_1}$ and $-v_1$ (for case, when $v_1 < 0$) we can see that $-v_1 > \sqrt{q_1}$ and, correspondently,

$$-v_1 - \sqrt{q_1} > 0. \quad (58)$$

Let's check the condition

$$\alpha_2 \in [0,\ 1] \quad (59)$$

for root $\alpha_2 = \left(-v_1 - \sqrt{q_1}\right)/w_1$ according to (38) and (42).

Based on (56)–(58) we can obtain the next inequalities:

$$w_1 > -v_1 - \sqrt{q_1},$$

$$0 < \left(-v_1 - \sqrt{q_1}\right)/w_1 < 1$$

and, correspondingly, $0 < \alpha_2 < 1$ or $\alpha_2 \in [0,\ 1]$.
Let's check the condition

$$\alpha_1 \in [0,\ 1] \tag{60}$$

for root $\alpha_1 = \left(-v_1 + \sqrt{q_1}\right)/w_1$ according to (38) and (42).
The condition (60) is correct for $-v_1 > 0$, $\sqrt{q_1} > 0$, $w_1 > 0$, if

$$w_1 \geq \left(-v_1 + \sqrt{q_1}\right), \tag{61}$$

$$(w_1 + v_1) \geq \sqrt{q_1}$$

or, correspondingly, for

$$(w_1 + v_1) \geq \sqrt{v_1^2 - 2k_3^2 w_1}. \tag{62}$$

Square the left and right sides of the inequality (62)

$$(w_1 + v_1)^2 \geq \left(v_1^2 - 2k_3^2 w_1\right)$$

and make the following sequence transformations

$$w_1^2 + 2w_1 v_1 + v_1^2 \geq v_1^2 - 2k_3^2 w_1,$$

$$w_1^2 + 2w_1 v_1 \geq -2k_3^2 w_1,$$

$$w_1 + 2v_1 + 2k_3^2 \geq 0. \tag{63}$$

Substituting (39) and (41) into (63) we can obtain

$$2((k_3 + C_1(\alpha) - k_1)^2 + k_2^2) + 2\left(2k_1 k_3 - 2k_3^2 - 2k_3 C_1(\alpha) - k_2^2\right) + 2k_3^2 \geq 0. \tag{64}$$

After dividing both sides of inequality (64) by 2, exponentiation and reduction of similar the expression (64) takes the following form

$$k_1^2 + c_1^2(\alpha) - 2k_1 c_1(\alpha) \geq 0,$$

$$(c_1(\alpha) - k_1)^2 \geq 0. \tag{65}$$

As inequality (65) and corresponding inequality (61) are confirmed, then

$$\left(-v_1+\sqrt{q_1}\right)/w_1<1,$$

and condition (60) is satisfied.
Taking into account that both conditions (59) and (60)

$$\alpha_1\in[0,1],\ \alpha_2\in[0,1]$$

are satisfied for roots $\alpha_1,\ \alpha_2$ of left branch's Eq. (31), the resulting value α_L^* can be find in the following way

$$\alpha_L^*=\max\{\alpha_1,\alpha_2\}: \tag{66}$$

$$\alpha_L^*=\frac{-\left(2k_1k_3-2k_3^2-2k_3c_1(\alpha)-k_2^2\right)+\sqrt{\begin{array}{l}\left(2k_1k_3-2k_3^2-2k_3c_1(\alpha)-k_2^2\right)^2-4\left(k_1^2+k_3^2+\right.\\ \left.+c_1^2(\alpha)-2k_1k_3-2k_1c_1(\alpha)+2k_3c_1(\alpha)+k_2^2\right)k_3^2\end{array}}}{2\left(k_1^2+k_3^2+c_1(\alpha)-2k_1k_3-2k_1c_1(\alpha)+2k_3c_1(\alpha)+k_2^2\right)}.$$

Investigation of the right branch and its properties for resulting MF. Using proposed approach it is possible to make all similar transformation for right branch of Eq. (30)

$$c_2(\alpha)=k_1+k_2\sqrt{\frac{1}{\alpha}-1}+k_3\left(\frac{1}{\alpha}-1\right), \tag{67}$$

which has two roots $\alpha_3,\ \alpha_4$.
As result we will obtain that both conditions

$$\alpha_3\in[0,1],\ \alpha_4\in[0,1]$$

are satisfied for roots $\alpha_3,\ \alpha_4$ of right branch of Eq. (30) and the resulting value α_R^* can be find in the following way

$$\alpha_R^*=\max\{\alpha_3,\alpha_4\}: \tag{68}$$

$$\alpha_R^*=\frac{-\left(2k_1k_3-2k_3^2-2k_3c_2(\alpha)-k_2^2\right)+\sqrt{\begin{array}{l}(2k_1k_3-2k_3^2-2k_3c_2(\alpha)-k_2^2)^2-4(k_1^2+k_3^2+\\ +c_2^2(\alpha)-2k_1k_3-2k_1c_2(\alpha)+2k_3c_2(\alpha)+k_2^2)k_3^2\end{array}}}{2\left(k_1^2+k_3^2+c_2^2(\alpha)-2k_1k_3-2k_1c_2(\alpha)+2k_3c_2(\alpha)+k_2^2\right)}.$$

Generalized direct model of resulting MF for multiplication of two bell-shape fuzzy sets. The transition from inverse to direct approach allows to transform (66)

$\alpha_L^* = f(c_1(\alpha))$ into $\alpha_L^* = f(x)$ for interval $x \in [0, k_1]$ and (68) $\alpha_R^* = f(c_2(\alpha))$ into $\alpha_R^* = f(x)$ for interval $x \in [k_1, \infty]$.

The analysis of the roots α_L^* and α_R^* allows to make a conclusion that nonlinear dependence $\alpha = f(x)$ for $x \in R^+$ is one-valued function.

The investigations allow to form a direct analytical model (69)–(71) for one-valued nonlinear resulting membership function $\mu_{\underset{\sim}{C}}(x)$ of fuzzy set $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$ that is formed by multiplying bell-shape fuzzy numbers $\tilde{\underset{\sim}{A}}$ та $\underset{\sim}{B}$ in R^+:

(a) $\forall x \in [0, p_1 p_2]$:

$$\mu_{\underset{\sim}{C}}(x) = \\ = \frac{-\left(2k_1k_3 - 2k_3^2 - 2k_3x - k_2^2\right) + \sqrt{\begin{aligned}&(2k_1k_3 - 2k_3^2 - 2k_3x - k_2^2)^2 - 4(k_1^2 + k_3^2 + \\ &+x^2 - 2k_1k_3 - 2k_1x + 2k_3x + k_2^2)k_3^2\end{aligned}}}{2\left(k_1^2 + k_3^2 + x^2 - 2k_1k_3 - 2k_1x + 2k_3x + k_2^2\right)}; \tag{69}$$

(b) for $x = p_1 p_2$:

$$\mu_{\underset{\sim}{C}}(x) = 1; \tag{70}$$

(c) $\forall x \in [p_1 p_2, \infty]$:

$$\mu_{\underset{\sim}{C}}(x) = \\ = \frac{-\left(2k_1k_3 - 2k_3^2 - 2k_3x - k_2^2\right) + \sqrt{\begin{aligned}&(2k_1k_3 - 2k_3^2 - 2k_3x - k_2^2)^2 - 4(k_1^2 + k_3^2 + \\ &+x^2 - 2k_1k_3 - 2k_1x + 2k_3x + k_2^2)k_3^2\end{aligned}}}{2\left(k_1^2 + k_3^2 + x^2 - 2k_1k_3 - 2k_1x + 2k_3x + k_2^2\right)}. \tag{71}$$

5 Inverse and Direct Analytic Models for Multiplication of Two Fuzzy Sets with Triangular Membership Functions in *R*

5.1 Synthesis of Inverse Model of Resulting MF

The task of synthesis of inverse and direct analytical models becomes complicated while performing the operation of multiplication of triangular fuzzy numbers $\underset{\sim}{A}$ (Fig. 1b) and $\underset{\sim}{B}$, that exist in the set of all real numbers *R*. We shall use the approach discussed above that is based on the analysis of the corresponding square roots for synthesis of inverse and direct models while implementing the operation of multiplication in R.

The algorithm of implementing the multiplication operation for triangular fuzzy numbers [4, 5, 9, 11]

$$\underset{\sim}{A} = (a_1, a_0, a_2)$$

and

$$\underset{\sim}{B} = (b_1, b_0, b_2)$$

while using α-cuts

$$A_\alpha = [a_1(\alpha),\ a_2(\alpha)] = [a_1 + (a_0 - a_1)\alpha,\ a_2 + (a_0 - a_2)\alpha]$$

and

$$B_\alpha = [b_1(\alpha),\ b_2(\alpha)] = [b_1 + (b_0 - b_1)\alpha,\ b_2 + (b_0 - b_2)\alpha]$$

is based on the next inverse model:

$$\begin{aligned} C_\alpha &= [c_1(\alpha),\ c_2(\alpha)] \\ &= \begin{bmatrix} \min\{a_1(\alpha)b_1(\alpha), a_2(\alpha)b_1(\alpha), a_1(\alpha)b_2(\alpha), a_2(\alpha)b_2(\alpha)\}, \\ \max\{a_1(\alpha)b_1(\alpha), a_2(\alpha)b_1(\alpha), a_1(\alpha)b_2(\alpha), a_2(\alpha)b_2(\alpha)\} \end{bmatrix} \\ &= \begin{bmatrix} \min\left\{ \begin{array}{c} [a_1 + (a_0 - a_1)\alpha][b_1 + (b_0 - b_1)\alpha], [a_2 + (a_0 - a_2)\alpha][b_1 + (b_0 - b_1)\alpha], \\ [a_1 + (a_0 - a_1)\alpha][b_2 + (b_0 - b_2)\alpha], [a_2 + (a_0 - a_2)\alpha][b_2 + (b_0 - b_2)\alpha] \end{array} \right\}, \\ \max\left\{ \begin{array}{c} [a_1 + (a_0 - a_1)\alpha][b_1 + (b_0 - b_1)\alpha], [a_2 + (a_0 - a_2)\alpha][b_1 + (b_0 - b_1)\alpha], \\ [a_1 + (a_0 - a_1)\alpha][b_2 + (b_0 - b_2)\alpha], [a_2 + (a_0 - a_2)\alpha][b_2 + (b_0 - b_2)\alpha] \end{array} \right\} \end{bmatrix} \end{aligned} \tag{72}$$

Where $c_1(0) = \min\{a_1b_1, a_2b_1, a_1b_2, a_2b_2\};$

$c_2(0) = \max\{a_1b_1, a_2b_1, a_1b_2, a_2b_2\};$

$c_1(1) = c_2(1) = a_0b_0.$

5.2 Synthesis of Direct Model of Resulting MF

Proposition 1. Using inverse model of resulting MF (72), we can form the direct model $\mu_C(x) = \mu_{A(\cdot)B}(x)$ in the following way:

$$\mu_C(x) = \begin{cases} 0, (\forall x < G_1) \bigcup (\forall x > G_2) \\ \alpha^* | \alpha^* \in \{\alpha_i,\}, i = 1\ldots8, (\forall x \in [G_1, a_0 b_0)) \bigcup (\forall x \in (a_0 b_0, G_2]) \\ 1, \forall x = a_0 b_0 \end{cases}, \quad (73)$$

Where $G_1 = \min\{a_1 b_1, a_2 b_1, a_1 b_2, a_2 b_2\};$

$$G_2 = \max\{a_1 b_1, a_2 b_1, a_1 b_2, a_2 b_2\}.$$

In the direct model (73) the roots of four square equations are used. These equations are formed while analyzing every of four components of the inverse model (72). In particular:

(a) for component

$$\begin{aligned}[a_1(\alpha) b_1(\alpha)] &= [a_1 + \alpha(a_0 - a_1)][b_1 + \alpha(b_0 - b_1)] \\ &= \alpha^2 (a_0 - a_1)(b_0 - b_1) + \alpha[(a_1(b_0 - b_1) + b_1(a_0 - b_1)] + a_1 b_1\end{aligned}$$

we form the equation

$$\alpha^2 (a_0 - a_1)(b_0 - b_1) + \alpha[a_1(b_0 - b_1) + b_1(a_0 - a_1)] + (a_1 b_1 - x) = 0,$$

the roots of which are

$$\alpha_{1,2} = \frac{-[a_1(b_0 - b_1) + b_1(a_0 - a_1)]}{2(a_0 - a_1)(b_0 - b_1)} \pm \frac{\sqrt{[a_1(b_0 - b_1) + b_1(a_0 - a_1)]^2 - 4(a_0 - a_1)(b_0 - b_1)(a_1 . b_1 - x)}}{2(a_0 - a_1)(b_0 - b_1)}; \quad (74)$$

(b) for the component

$$[a_2(\alpha) b_1(\alpha)] = [a_2 + (a_0 - a_2)\alpha][b_1 + (b_0 - b_1)\alpha]$$

the roots of the formed equation

$$\alpha^2 (b_0 - b_2)(a_0 - a_1) + \alpha[a_1(b_0 - b_2) + b_2(a_0 - a_1)] + (a_1 b_2 - x) = 0$$

are

$$\alpha_{3,4} = \frac{-[b_2(a_0 - a_1) + a_1(b_0 - b_2)]}{2(b_0 - b_2)(a_0 - a_1)} \pm \frac{\sqrt{[b_2(a_0 - a_1) + a_1(b_0 - b_2)]^2 - 4(b_0 - b_2)(a_0 - a_1)(a_1 b_2 - x)}}{2(b_0 - b_2)(a_0 - a_1)} \quad (75)$$

(c) for the component

$$[a_1(\alpha)b_2(\alpha)] = [a_1 + (a_0 - a_1)\alpha][b_2 + (b_0 - b_2)\alpha]$$

the roots of the formed equation

$$\alpha^2(a_0 - a_2)(b_0 - b_1) + \alpha[a_2(b_0 - b_1) + b_1(a_0 - a_2)] + (a_2 b_1 - x) = 0$$

are

$$\alpha_{5,6} = \frac{-[a_2(b_0 - b_1) + b_1(a_0 - a_2)]}{2(a_0 - a_2)(b_0 - b_1)} \pm \frac{\sqrt{[a_2(b_0 - b_1) + b_1(a_0 - a_2)]^2 - 4(a_0 - a_2)(b_0 - b_1)(a_2 b_1 - x)}}{2(a_0 - a_2)(b_0 - b_1)}; \quad (76)$$

(d) for component

$$[a_2(\alpha)b_2(\alpha)] = [a_2 + (a_0 - a_2)\alpha][b_2 + (b_0 - b_2)\alpha]$$

the roots of the formed equation

$$\alpha^2(b_0 - b_2)(a_0 - a_1) + \alpha[a_1(b_0 - b_2) + b_2(a_0 - a_1) + (a_1 b_2 - x) = 0$$

are

$$\alpha_{7,8} = \frac{-[a_2(b_0 - b_2) + b_2(a_0 - a_2)]}{2(a_0 - a_2)(b_0 - b_2)} \pm \frac{\sqrt{[a_2(b_0 - b_2) + b_2(a_0 - a_2)]^2 - 4(a_0 - a_2)(b_0 - b_2)(a_2 b_2 - x)}}{2(a_0 - a_2)(b_0 - b_2)}. \quad (77)$$

The algorithm according to which the value of α^* is selected based on (74)–(77) has the following interpretation.

We define a subset of indices $I_1 \in I$ under the condition

$$I_1 = \{i | i \in I \cap \alpha_i \in [0, 1]\}$$

The subset of indices I_1 belongs to the set of indices

$$I = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

of all roots (74)–(77) defined above: $\alpha_i, \ i = 1 \ldots 8$.

Proposition 2. The investigations made by the authors show that the presence of some roots that satisfy the condition $\alpha_i \in [0, 1]$ the parameter α^* is defined as follows

$$\alpha^* = \max_{i \in I_1} \{\alpha_i\}. \quad (78)$$

5.3 Modeling Results

Functional dependences of the roots $\alpha_i = f_i(x), i = 1\ldots8$ from the parameter x while implementing the operation of multiplication of triangular fuzzy numbers in $\boldsymbol{R}$:

$$\underset{\sim}{A} = (-3,\ 1,\ 8)$$

and

$$\underset{\sim}{B} = (-2,\ 2,\ 4)$$

are given in Fig. 5, and the chart of the resulting membership function for model (73) – in Fig. 6.

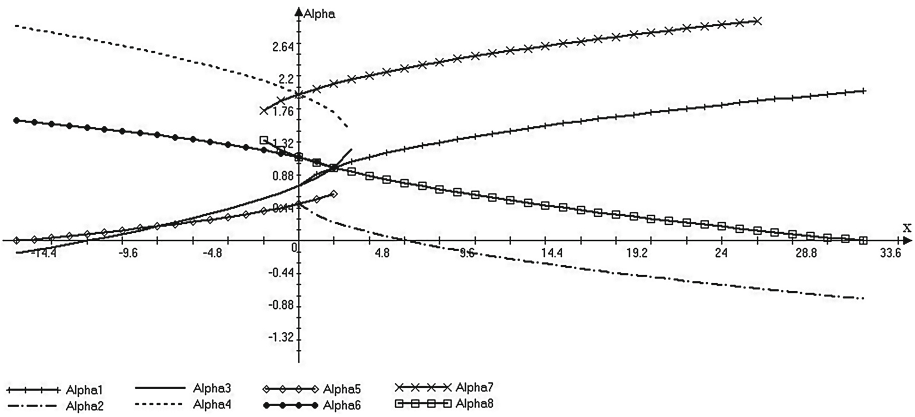


Fig. 5. Functional dependences of $\alpha_i = f_i(x), i = 1\ldots8$

Figure 6 illustrates that during the process of changing x from -16 to 32, the thunking α (Alpha) takes place according to the chain

$$\alpha_5 - \alpha_3 - \alpha_1 - \alpha_8.$$

(points 1, 2, 3 and 4 in Fig. 6).

6 Applied Aspects of Universal Analytic Models Implementation in Decision Support Processes

The multiplication of fuzzy sets is very important operation which is most complicated operation in fuzzy arithmetic and requires a lot of time for calculation processes. The implementation of developed direct analytic models for calculation of resulting membership functions $\mu_{\underset{\sim}{C}}(x)$ according to Table 1 and (70) allows using one step automation mode for operation $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B}$. In some cases such direct analytic models $\mu_{\underset{\sim}{C}}(x) = \mu_{\underset{\sim}{A}(\cdot)\underset{\sim}{B}}(x)$ may have efficient introducing to evaluation, decision making and decision support processes.

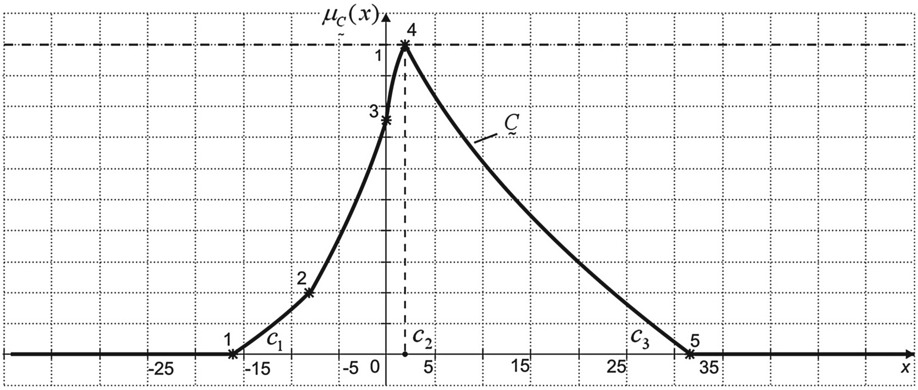


Fig. 6. Resulting fuzzy set $\underset{\sim}{C} = \underset{\sim}{A}(\cdot)\underset{\sim}{B} : \underset{\sim}{A} \subset R, \underset{\sim}{B} \subset R$

Let's consider several examples of real life problems solving [10] for decision support processes, where E is a set of alternatives

$$E = \{E_1, E_2, E_3, \ldots, E_i, \ldots, E_L\}. \tag{79}$$

6.1 Transportation Problem

The transportation problem is well-known operation research problem [7, 8] where it is necessary to find the best alternative solution $E^* \subset E$ for cargo transportation from several depots $N = \{N_1, N_2, \ldots, N_m\}$ to several customers $S = \{S_1, S_2, \ldots, S_n\}$.

The ship transportation problem (STP) can be formulated as a classical transportation problem which can be solved by linear programming methods with some modifications taking into account real conditions of marine environment.

At the general statement STP minimizes of the cost (goal/objective) function $Z(c, x)$ concerning to the transportation of various kind of cargoes (oil, coal, fuel etc.) from several (or one) supplying ports (deports) to various receiving ports (nodes):

$$\text{Min Z}(c, x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}, \tag{80}$$

where: x_{ij} is a quantity of cargo which can be transported from deport i to port $j, i = 1, \ldots, m; j = 1, 2, \ldots, n,;$

c_{ij} is associated with a transported cost of cargo unit (for example, one ship's capacity) for cargo transportation from deport i to port $j, i = 1,\ldots,m; j = 1,2,\ldots,n.$

It is very important to solve STP from the viewpoint of changeable character of marine conditions as problem in fuzzy environment.

Really, it is well-known that marine environment under consideration may create super-changeable conditions and the degree of fuzziness of such environment has very high level. So, in marine (or sea) environment each coefficient c_{ij} of goal function $Z(c,x)$ is, in general case, uncertain. It depends on the various external factors because each (ij)-separated service in STP includes transport penalty (cost) c_{ij}^T, unload penalty c_{ij}^L, penalties c_{ij}^S and c_{ij}^E, which represent, respectively, satisfaction of the safety's and ecology's requirements at the j-th port (j=1,2,…,n) during unloading operations and other components, for example, port's sanitary penalty or custom's penalty, which can be included to the one of the abovementioned cost components c_{ij}^K $(K = (T,L,S,E))$. In this case goal function (80) can be written as fuzzy function

$$Min\, \underset{\sim}{Z}(c,x) = \sum_{i=1}^{m}\sum_{j=1}^{n} \underset{\sim}{c}_{ij} x_{ij}, \quad (81)$$

where all coefficients $\underset{\sim}{c}_{ij}$ $(i = 1,2,\ldots m; j = 1,2,\ldots n)$ are fuzzy sets in the universal set of positive numbers R^+.

Let's consider in detail abovementioned disturbed factors for marine environment according to

$$\underset{\sim}{c}_{ij} = \underset{\sim}{c}_{ij}^{T} \oplus \underset{\sim}{c}_{ij}^{L} \oplus \underset{\sim}{c}_{ij}^{S} \oplus \underset{\sim}{c}_{ij}^{E}.$$

The distance between i-th deport and j-th port defines the value of transport penalty $\underset{\sim}{c}_{ij}^{T}$. It is most uncertain component, which depends on, first of all, weather conditions in served marine region:

(a) when weather is fine the captain of the supplied ship can choose the regular path for his ship from i-th deport to j-th port and this regular path, as usual (if no obstacles exist), has approximately straightforward character or, in the other words, it should be, desirably, a shortest way;
(b) when the weather has a tendency to change into storm conditions according to current meteorological prognosis the captain can choose another, from the viewpoint of ship's safety, path which as usual is more longer than regular path. Captain should form this new ship's path taking into account the distance between ship and bay (or various bays, which lie in the nearest distances from the ship's path). Sometimes the captain makes a decision to interrupt ship movement from the reason of ship's safety and to stay at the nearest bay for uncertain period which depends on improving of weather conditions.

In some cases the captain changes course of ship according to appearances of some temporary restrictive zones or dynamical obstacles etc. Finally, any type of time delay influences to the increasing of total distance of ship's path and fuel consumption and leads to increasing of penalty component $\underset{\sim}{c}_{ij}^{T}$. The power (force) and directions of wind and sea current have as usual changeable character that also influences to penalty component $\underset{\sim}{c}_{ij}^{T}$, in some cases, by increasing or decreasing of $\underset{\sim}{c}_{ij}^{T}$ according to value $\underset{\sim}{c}_{ij}^{T}$ with grade of membership function $\mu_{\underset{\sim}{c}_{ij}^{T}}\left(\hat{c}_{ij}^{T}\right) = 1$.

The uncertainness of unloading penalty $\underset{\sim}{c}_{ij}^{L}$ depends on quantity of receiving cargo at the j-th port and this parameter in STP is uncertain and "a priori" identified as approximately in terms "about $\hat{c}_{ij}^{L}$ " or "average $\hat{c}_{ij}^{L}$ " for real situations.

Penalty components $\underset{\sim}{c}_{ij}^{S}$ and $\underset{\sim}{c}_{ij}^{E}$ depend on the j-th port rules as well as the weather conditions also. It is often necessary to provide the certain requirements to ensure limits of ship's hill, trim and stress for hull e.g. [7, 8] to install a special type of floating protection around sea's unloaded area for restriction of fuel dissemination at the sear surface if any damages may happen in unloading operations. Sometimes (and it depends on cargo type) penalty $\underset{\sim}{c}_{ij}^{S}$ includes the expenses for fire-ship providing a safety of unloading operations during unloading time.

Analysing all coefficients $\underset{\sim}{c}_{ij}$, we can form matrix $\underset{\sim}{C}$ $(\underset{\sim}{c}_{ij} \neq \underset{\sim}{c}_{ij}, i = 1, 2, \ldots m;$ $j = 1, 2, \ldots n)$ as asymmetrical fuzzy matrix of penalties $\underset{\sim}{c}_{ij}$ for solving of STP

$$\underset{\sim}{C} = \left\| \underset{\sim}{c}_{ij} \right\|_{(m \times n)} = \begin{vmatrix} \underset{\sim}{c}_{11} & \underset{\sim}{c}_{12} & \cdots & \underset{\sim}{c}_{1n} \\ \underset{\sim}{c}_{21} & \underset{\sim}{c}_{22} & \cdots & \underset{\sim}{c}_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \underset{\sim}{c}_{m1} & \underset{\sim}{c}_{m2} & \cdots & \underset{\sim}{c}_{mn} \end{vmatrix}. \quad (82)$$

It should be mentioned that in some real situations cargo demands $\sum_{j=1}^{n} \underset{\sim}{x}_{ij}$ of several ports (destinations) are uncertain values (preplanned demand is a such uncertain type as "about VALUE", "Approximately VALUE", "between VALUE_1 and VALUE_2") and respectively the value of total quantity of supplied cargo for each deport is also uncertain.

In such situation the goal function (81) can be transformed to such fuzzy function as

$$Min\, \underset{\sim}{Z}(c, x) = \sum_{i=1}^{m} \sum_{j=1}^{n} \underset{\sim}{c}_{ij} \underset{\sim}{x}_{ij}, \quad (83)$$

where both components $\underset{\sim}{c}_{ij}, \underset{\sim}{x}_{ij}$ are corresponding fuzzy sets.

Solving optimization problem (83) with corresponding restrictions for real transportation task it is necessary multi-timely to calculate the results

$$\underset{\sim}{c}_{ij}(\cdot)\underset{\sim}{x}_{ij}$$

of multiplication operation for different pairs of fuzzy sets $\underset{\sim}{c}_{ij}, \underset{\sim}{x}_{ij} (i = 1\ldots m; j = 1\ldots n)$.

In the cases of decision making process when the fuzzy sets $\underset{\sim}{c}_{ij}, \underset{\sim}{x}_{ij}$ are represented as triangular fuzzy numbers it is very efficient to use developed analytic models $\mu_{\underset{\sim}{C}}(x) = \mu_{\underset{\sim}{A}(\cdot)\underset{\sim}{B}}(x)$ in Sects. 2, 4 and 5 of this chapter.

6.2 Decision Making in Agriculture Sector

In some cases it is necessary to find best alternative from (79) based on the prognosis of total profit after agriculture season using multiplication operation

$$\underset{\sim}{C}_i = \underset{\sim}{A}_i(\cdot)\underset{\sim}{B}_i, \tag{84}$$

where $\underset{\sim}{A}_i$ is triangular fuzzy number which corresponds to future output value of i-th agriculture product, $(i = 1\ldots L)$; $\underset{\sim}{B}_i$ is triangular fuzzy number which corresponds to future price value for a unit of i-th agriculture product, $(i = 1\ldots L)$; $\underset{\sim}{C}_i$ is fuzzy number which corresponds to future profit value at the end of agriculture season, in particular, after realization of i-th agriculture product, $(i = 1\ldots L)$.

7 Conclusions

The usage of the developed analytical models (11), (69)–(71), (73), (78) has significant advantage for accuracy of calculations, time of modeling and program implementation of the formed models in comparison with step by step models of multiplication operation of triangular fuzzy numbers based on the algorithms of sorting and max-min convolutions [4, 12]. Suggested approach can be used for different types of MFs and fuzzy models, in particular, for triangular MFs, bell-shape MFs, trapezoidal MFs, etc. Modeling results for multiplication of different fuzzy numbers with triangular and bell-shape membership functions confirm the efficiency of proposed universal analytic models for different applications, in particular, for soft computing based on reconfigurable technology [13], risk analysis in the test diagnosis of the digital components for systems of critical applications [1], and solving real life decision support problems [2, 3, 6, 7, 16], partly presented in Sect. 6 of the chapter.

References

1. Drozd, A., Kharchenko, V., Antoshchuk, S., Sulima, J., Drozd, M.: Checkability of the digital components in safety-critical systems: Problems and solutions. In: Proceedings of IEEE East-West Design and Test Symposium (EWDTS 2011), pp. 411–416 (2011)
2. Gil-Aluja, J., Gil-Lafuente, A.M., Klimova, A.: The optimization of the economic segmentation by means of fuzzy algorithms. J. Comput. Optim. Econ. Finance **1**(3), 169–186 (2011)
3. Gil-Lafuente, A.M.: Fuzzy Logic in Financial Analysis. StudFuzz, vol. 175. Springer, Heidelberg (2005)
4. Kaufmann, A., Gupta, M.: Introduction to Fuzzy Arithmetic: Theory and Applications. Van Nostrand Reinhold Company, New York (1985)
5. Kondratenko, V.: Synthesis of analytic models for computing operations with fuzzy numbers: Operation of multiplication. J. Tech. News **1**(20), **2**(21), 85–90 (2005) (in Ukrainian)
6. Kondratenko, Y.P.: Optimisation Problems in Marine Transportation. Incidencia de las Relaciones Economicas Internacionales en la Recuperacion Economica del Area Mediterranea, pp. 43–52. Real Academia de Ciencias Economicas y Financieras, Barcelona (2011)
7. Kondratenko, G.V., Kondratenko, Y.P., Romanov, D.O.: Fuzzy models for capacitative vehicle routing problem in uncertainty. In: Proceeding of 17th International DAAAM Symposium "Intelligent Manufacturing and Automation: Focus on Mechatronics and Robotics", pp. 205–206, Vienna, Austria (2006)
8. Kondratenko, V., Iablonska, K., Kondratenko, Y. The universal analytical models of fuzzy arithmetic computing operations: direct and inverse approach. In: 17th Zittau East-West Fuzzy Collquium, pp. 112–119. Wissenschaftliche Berichte, Heft 106/2010, Nr.2454-2490 (2010)
9. Kondratenko, Y.P., Sidenko, Ie.V.: Method of actual correction of the knowledge database of fuzzy decision support system with flexible hierarchical structure. J. Comput. Optim. Econ. Finance **4**(2–3), 57–76 (2012)
10. Kondratenko, V.Y.: Analysis of characteristics and mathematical model of result membership function for multiplication of fuzzy numbers. Visnyk of Kherson National Technical University **1**(24), 488–494 (2006) (in Ukrainian)
11. Leonenkov, A.V.: Fuzzy simulation in the environment MatLab and FuzzyTECH. BHV-Petersburg, St. Petersburg (2005). (in Russian)
12. Piegat, A.: Fuzzy Modeling and Control. Springer, Heidelberg (2001)
13. Palagin, A.V., Opanasenko, V.N.: Reconfigurable computing technology. J. Cybern. Syst. Anal. **43**(5), 675–686 (2007)
14. Rotshtein, A.P.: Intellectual Technologies of Identification: Fuzzy Logic, Genetic Algorithms, Neuron Networks. UNIVERSUM, Vinnitsa (1999). (in Russian)
15. Gao, S., Zhang, Z.: Multiplication operation on fuzzy numbers. J. Softw. **4**(4), (2009). http://ojs.academypublisher.com/index.php/jsw/article/download/0404331338/1061
16. Werners, B., Kondratenko, Y.P.: Tanker routing problem with fuzzy demands of served ships. Int. J. Syst. Res. Inf. Technol. **1**, 47–64 (2009)
17. Yager, R., Filev, D.: Essential of Fuzzy Modeling and Control. Wiley, New York (1994)
18. Yen, J., Langari, R.: Fuzzy Logic: Intelligence, Control and Information. Prentice Hall, Upper Saddle River (1999)
19. Zadeh, L.A.: Fuzzy Sets. Inf. Control **8**, 338–353 (1965)

On a Decidable Formal Theory for Abstract Continuous-Time Dynamical Systems

Ievgen Ivanov[1,2(✉)], Mykola Nikitchenko[1], and Uri Abraham[3]

[1] Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ivanov.eugen@gmail.com, nikitchenko@unicyb.kiev.ua
[2] Paul Sabatier University, Toulouse, France
[3] Ben-Gurion University, Beer-Sheva, Israel
abraham@cs.bgu.ac.il

Abstract. We propose a decidable formal theory which describes high-level properties of abstract continuous-time dynamical systems called Nondeterministic Complete Markovian Systems (NCMS). NCMS is a rather general class of systems which can represent discrete and/or continuous evolutions in continuous time and which is sufficient for modeling a wide range of real-time information processing and cyber-physical systems (CPS). We illustrate the obtained results with a proof of the mutual exclusion property of a CPS which implements Peterson's algorithm.

Keywords: Real-time system · Cyber-physical system · Dynamical system · Continuous-time system · Formal theory · Decidability · Verification

Key Terms. Mathematical Model, Specification Process, Verification Process

1 Introduction

A large amount of computing systems used today act as agents interacting with physical processes. They are now frequently called cyber-physical systems [1,2]. Examples include autonomous automotive systems, robotics, process control, medical devices, energy conservation, etc. [3]. Let us give some quotes from the Cyber-Physical Systems (CPS) concept map [4] by S.S. Sunder of NIST (USA), E.A. Lee of UC Berkeley (USA) and others:

"CPS integrates the dynamics of the physical processes with those of the software and networking, providing abstractions and modeling, design, and analysis techniques for the integrated whole" [4].

"Classical models of computation in computer science, rooted in Turing-Church theories for non-concurrent systems, and in nondeterministic transition

This work was supported in part by the research project No. 11BF015-02 "Formal specifications and methods of development of reliable software systems", Taras Shevchenko National University of Kyiv, Ukraine.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 78–99, 2014.
DOI: 10.1007/978-3-319-13206-8_4

systems and process algebras for concurrent systems, do not handle temporal dynamics well" [4].

"A key CPS challenge is to conjoin the engineering abstractions for continuous dynamics (such as differential equations) with computer science abstractions (such as algorithms)" [4].

Besides, the following research needs in CPS are outlined in [1]: Abstraction and Architectures, Distributed Computations and Networked Control, and Verification and Validation. With regard to the first aspect (Abstraction and Architectures) it is stated that

"Innovative approaches to abstraction and architectures that enable seamless integration of control, communication, and computation must be developed for rapid design and deployment of CPS" [1].

The mentioned challenges imply the importance of development of adequate system models of various levels of abstraction and generality with emphasis on system's temporal behavior which should not be restricted to a purely discrete or purely continuous evolution.

Although not aimed specifically at solving the mentioned challenges, many concrete models that combine a discrete and continuous behavior in some way were studied in control theory, theory of differential equations, and computer science, e.g. variable structure systems [5–7], impulsive differential equations [8,9], differential equations with discontinuous right hand sides [10], switched systems [11], hybrid control systems [12–16], hybrid automata [17–19], phase transition systems [20], hybrid reactive modules [21], hybrid I/O automata [22]. The mentioned models may be useful for solving certain CPS-related problems, but they do not handle well such important aspects of CPS as levels of abstractions and distributed organization. As a result, high-level properties of CPS (e.g. mutual exclusion of access to a shared resource by different components within a CPS) are difficult to validate (and even formalize) in such models.

A possible solution to this problem is application of temporal [23–26] and dynamic logics [27–29] which allow reasoning about continuous-time systems (e.g. Temporal Logic of Reals [23], Duration Calculus [25], Monadic Second Order Logic over the structure of boolean finitely variable signals [24], continuous-time interpretation of Temporal Logic of Action [30], Real-time Temporal Logic [31], Differential Dynamic Logic [29], etc.). One can specify common high-level properties of interest of CPS rather straightforwardly in such logics and prove them. However, for expressive logics of this kind [24,27–29] the validity problem is known to be undecidable (which is frequently related to the undecidability of the reachability problem for various classes of hybrid systems), so the search for decidable but useful fragments is still a relevant topic of research. It should be noted that most logics (fragments) that are known to be decidable are concerned with reasoning about systems which satisfy a kind of a finite variability assumption (non-Zenoness) [24,32]: discrete-valued time-varying quantities of interest associated with the system can be modeled as piecewise constant functions (with finite sets of discontinuities over each bounded time segment). Although this assumption is reasonable in many practical cases, some real-world systems have adequate mathematical models which violate it (the simplest example is a hybrid automaton with Zeno executions which models of the dynamics of a bouncing ball [33]). This makes the problem of investigation of decidable formal theories which allow reasoning about more general classes of continuous-time dynamical systems important.

In this paper we will define and prove decidability of one such formal theory. For this purpose we need to select a sufficiently general class of dynamical systems. As we are interested in high-level properties, we will require such dynamical systems to be abstract models of CPS (which do not reflect excessive details about their structure and operation). We will also use the following informal assumption: each logically possible realization of the run-time behavior of a CPS can be modeled as an evolution of a (single) global state in the global continuous time (although the global state and time may be treated as purely mathematical objects with no physical manifestation). This suggests that the overall behavior of a CPS can be modeled as a set of functions (trajectories) from the global time domain to the set of global states which represent logically possible realizations of the system's run-time behavior. On the other hand, well-known types of continuous-time dynamical systems have associated sets of trajectories as functions from a time domain to a state space. So in this paper we will consider a dynamical system to be a model of a CPS, if its associated set of trajectories represents the set of possible run-time behaviors of this CPS.

Classes of continuous-time dynamical systems of various levels of generality were considered in many works [5,34–44]. Classical approaches to the definition of a dynamical system, such as those proposed by A.A. Markov, V.V. Nemytskii and V.V. Stepanov [35] and others (an overview can be found in [43]) can be considered as axiomatizations of properties of systems described by differential equations. As was noted in the work [38], the following properties of ordinary differential equations were of main concern in various axiomatizations: (1) local existence of solutions, (2) indefinite prolongability (global existence) of solutions, (3) unicity of solutions, (4) autonomness (the right-hand side of the equation does not depend explicitly on time). However, in a number of works [5,37–40,43,45,46], etc., there was a tendency to remove some of these properties from basic assumptions and consider increasingly general classes of dynamical systems (a comparison of many such approaches is given in [43]). In particular, in [38] it was proposed to eliminate the properties (1)–(4) mentioned above from the axiomatization to obtain a far-reaching generalization of dynamical systems. Variants of such a generalization include the notion of a process [38,47] and the (process-independent) notion of a solution system [38] by O. Hájek. Similar ideas also appeared in some other works [40,43,44]. The general notion of a solution system by O. Hájek was used as basis of a more restricted notion of a Nondeterminisitc Complete Markovian System (NCMS) introduced in [48] for the purpose of studying the relation between the existence of global and local trajectories of dynamical systems. A NCMS is associated with a set of trajectories which are partial functions on the non-negative real time domain which satisfies certain assumption [48]. Although the class of NCMS is more restricted than the class of solution systems, it is sufficient for representing a very general class of causal (nonanticipative) systems which interact with the external environment using continuous-time input and output signals [49,50], so we assume it to be sufficient for high-level modeling of real-time information processing systems and cyber-physical systems. For the reasons mentioned above, we will assume that the class of NCMS is sufficiently general, does not impose restrictions on the dynamics like the finite-variability assumption, and is suitable as a class of high-level models of CPS.

In this paper we will define a language for expressing various essential properties of NCMS, define a language interpretation, and show that the associated formal theory (the set of sentences which are valid in all interpretations) is decidable. To prove the decidability result we will use a reduction to the decidability of the monadic second-order theory of order of the real segment $[0, 1]$ with quantification over F_σ-subsets (countable unions of closed sets) which was proved by M.O. Rabin [51] (as a consequence of the decidability of $S2S$, the monadic second-order theory of two successors). Then we will describe an example of an application of the obtained results for proving the mutual exclusion property of a CPS which implements Peterson's algorithm. This example is an extension of the approach to proving properties of distributed algorithms which was proposed in [52].

2 Preliminaries

2.1 Notation

We use the following notation: $\mathbb{N} = \{1, 2, 3, ...\}$, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, $\mathbb{R}$ is the set of real numbers, $\mathbb{R}_+$ is the set of nonnegative real numbers, $f : A \to B$ is a total function from A to B, $f : A \tilde{\to} B$ is a partial function from A to B, 2^A is the power set of a set A, $f|_X$ is the restriction of a function f to a set X. If A, B are sets, then B^A denotes the set of all total functions from A to B and ${}^A B$ denotes the set of all partial function from A to B. For a function $f : A \tilde{\to} B$ the symbol $f(x) \downarrow$ ($f(x) \uparrow$) means that $f(x)$ is defined (respectively undefined) on the argument x. We do not distinguish formally the notion of a function and a functional binary relation. When we write that a function $f : A \tilde{\to} B$ is total or surjective, we mean that f is total on A specifically (i.e. $f(x) \downarrow$ for all $x \in A$), or, respectively, is onto B (i.e. for each $y \in B$ there exists $x \in A$ such that $y = f(x)$). The domain and range of a partial function $f : A \tilde{\to} B$ are $dom(f) = \{x \mid f(x) \downarrow\}$ and $range(f) = \{y \mid \exists x \; f(x) \downarrow \wedge \, y = f(x)\}$ respectively (note that in some areas of mathematics and computer science, including category theory, it is assumed that the domain and range of a partial function from A to B are A and B respectively, but we do not follow this convention in this paper and instead use the definitions given above). For partial functions f, g,$f(x) \cong g(x)$ denotes the strong equality (where x is a fixed value): $f(x) \downarrow$ if and only if $g(x) \downarrow$, and $f(x) \downarrow$ implies $f(x) = g(x)$. By $f \circ g$ we denote the functional composition: $(f \circ g)(x) \cong f(g(x))$.

By T we denote the non-negative real time scale $[0, +\infty)$, equipped with a topology induced by the standard topology on $\mathbb{R}$ and by $Bool = \{true, false\}$ we denote the set of boolean values equipped with the discrete topology.

The symbols $\neg$, $\vee$, $\wedge$, $\Rightarrow$, and $\Leftrightarrow$ denote the logical operations of negation, disjunction, conjunction, implication, and equivalence respectively.

2.2 Nondeterministic Complete Markovian Systems (NCMS)

Let $T = \mathbb{R}_+$ be the non-negative real time scale. Denote by $\mathfrak{T}$ the set of all intervals (connected subsets) in T which have cardinality greater than one (i.e. which are non-empty and non-singleton sets).

Let Q be a set (a state space) and Tr be some set of functions of the form $s : A \to Q$, where $A \in \mathfrak{T}$. Let us call its elements trajectories.

Definition 1. *[48,49] A set of trajectories Tr is closed under proper restrictions (CPR), if $s|_A \in Tr$ for each $s \in Tr$ and $A \in \mathfrak{T}$ such that $A \subseteq dom(s)$.*

Definition 2. *[48,49]*

(1) A trajectory $s_1 \in Tr$ is a subtrajectory of $s_2 \in Tr$ (denoted as $s_1 \sqsubseteq s_2$), if $dom(s_1) \subseteq dom(s_2)$ and $s_1 = s_2|_{dom(s_1)}$.
(2) A trajectory $s_1 \in Tr$ is a proper subtrajectory of $s_2 \in Tr$ (denoted as $s_1 \sqsubset s_2$), if $s_1 \sqsubseteq s_2$ and $s_1 \neq s_2$.

The set $(Tr, \sqsubseteq)$ is a (possibly empty) partially ordered set (poset).

Definition 3. *[48,49] A CPR set of trajectories Tr is*

(1) Markovian (Fig. 1), if for each $s_1, s_2 \in Tr$ and $t \in T$ such that
$t = \sup dom(s_1) = \inf dom(s_2)$, $s_1(t)\downarrow$, $s_2(t)\downarrow$, and $s_1(t) = s_2(t)$,
the following function s belongs to Tr:
$s(t) = s_1(t)$, if $t \in dom(s_1)$, and
$s(t) = s_2(t)$, if $t \in dom(s_2)$.
(2) complete, if each non-empty chain in $(Tr, \sqsubseteq)$ has a supremum.

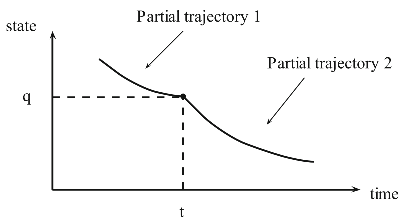


Fig. 1. Markovian property of NCMS. If one trajectory ends and another begins in a state q at time t, then their concatenation is a trajectory.

Definition 4. *[48] A nondeterministic complete Markovian system (NCMS) is a triple (T, Q, Tr), where Q is a set (state space) and Tr (trajectories) is a set of functions $s : T \tilde{\to} Q$ such that $dom(s) \in \mathfrak{T}$, which is CPR, complete, and Markovian.*

This is an *intensional* definition. An alternative *extensional* definition (or an overview of the class of all NCMS) can be given using the notion of an *LR representation* of NCMS which is described below.

Definition 5. *Let $s_1, s_2 : T \tilde{\to} Q$. Then s_1 and s_2 coincide:*

(1) on $A \subseteq T$, if $s_1|_A = s_2|_A$ and $A \subseteq dom(s_1) \cap dom(s_2)$ (this is denoted as $s_1 \doteq_A s_2$);
(2) in a left neighborhood of $t \in T$, if $t > 0$ and there exists $t' \in [0, t)$ such that $s_1 \doteq_{(t',t]} s_2$ (this is denoted as $s_1 \doteq_{t-} s_2$);
(3) in a right neighborhood of $t \in T$, if there exists $t' > t$, such that $s_1 \doteq_{[t,t')} s_2$ (this is denoted as $s_1 \doteq_{t+} s_2$).

Let Q be a set. Denote by $ST(Q)$ the set of pairs (s, t) where $s : A \to Q$ for some $A \in \mathfrak{T}$ and $t \in A$.

Definition 6. *[48,49] A predicate $p : ST(Q) \to Bool$ is called*

(1) left-local, if $p(s_1, t) \Leftrightarrow p(s_2, t)$ whenever $(s_1, t), (s_2, t) \in ST(Q)$ and $s_1 \doteq_{t-} s_2$, and, moreover, $p(s, t)$ whenever t is the least element of $dom(s)$;
(2) right-local, if $p(s_1, t) \Leftrightarrow p(s_2, t)$ whenever $(s_1, t), (s_2, t) \in ST(Q)$, $s_1 \doteq_{t+} s_2$, and, moreover, $p(s, t)$ whenever t is the greatest element of $dom(s)$.

Denote by $LR(Q)$ the set of all pairs (l, r), where $l : ST(Q) \to Bool$ is a left-local predicate and $r : ST(Q) \to Bool$ is a right-local predicate.

Definition 7. *[49] A pair $(l, r) \in LR(Q)$ is called a LR representation of a NCMS $\Sigma = (T, Q, Tr)$, if $Tr = \{s : A \to Q \mid A \in \mathfrak{T} \wedge (\forall t \in A\ l(s, t) \wedge r(s, t))\}$.*

Theorem 1. *[49, Theorem1]*

(1) Each pair $(l, r) \in LR(Q)$ is a LR representation of a NCMS with the set of states Q.
(2) Each NCMS has a LR representation.

Although an LR representation of a given NCMS needs not be unique, there always exists a unique *least* LR representation.

Definition 8. *The least LR representation (l^*, r^*) of a NCMS $\Sigma = (T, Q, Tr)$ is a LR representation (l, r) of Σ such that $l^*(s, t) \Rightarrow l(s, t)$ and $r^*(s, t) \Rightarrow r(s, t)$ for all $(s, t) \in ST(Q)$.*

Theorem 2. *(1) Each NCMS has a unique least LR representation (l^*, r^*).*
(2) If $\Sigma = (T, Q, Tr)$ is a NCMS and (l^*, r^*) is its least LR representation, then for all $(s, t) \in ST(Q)$,

$$l^*(s, t) \Leftrightarrow t = 0 \vee (\exists t' \in (0, t)\ s|_{(t',t]} \in Tr \cup \{s|_{\{t\}}\});$$
$$r^*(s, t) \Leftrightarrow \exists t' \in (t, +\infty)\ s|_{[t,t')} \in Tr \cup \{s|_{\{t\}}\}.$$

Proof (Sketch).

(1) Let $\Sigma = (T, Q, Tr)$ be a NCMS. Let us define predicates $l^* : ST(Q) \to Bool$ and $r^* : ST(Q) \to Bool$ as follows: $l^*(s,t)$ if and only if $l(s,t)$ holds for each LR representation (l,r) of Σ, and $r^*(s,t)$ if and only if $r(s,t)$ holds for each LR representation (l,r) of Σ. It is easy to check that a pointwise conjunction of any non-empty set of left-local predicates on $ST(Q)$ is left-local and a pointwise conjunction of any non-empty set of right-local predicates on $ST(Q)$ is right-local. By Theorem 1, Σ has a LR representation, so l^* is left-local and r^* is right-local, so $(l^*, r^*) \in LR(Q)$. Moreover, for any $A \in \mathfrak{T}$ and $s : A \to Q$, $l^*(s,t) \wedge r^*(s,t)$ holds for all $t \in A$ if and only if $l(s,t) \wedge r(s,t)$ holds for all $t \in A$ and all LR representations (l,r) of Σ, i.e. if and only if $s \in Tr$. Thus (l^*, r^*) is a LR representation of Σ and is the least LR representation. Uniqueness of the least LR representation of a NCMS follows straightforwardly from its definition.

(2) Let us introduce predicates $l_0, r_0 : ST(Q) \to Bool$ such that for all s, t:

$$l_0(s,t) \Leftrightarrow t = 0 \vee (\exists t' \in (0,t)\ s|_{(t',t]} \in Tr \cup \{s|_{\{t\}}\});$$
$$r_0(s,t) \Leftrightarrow \exists t' \in (t, +\infty)\ s|_{[t,t')} \in Tr \cup \{s|_{\{t\}}\}.$$

Using the Markovian and CPR properties of Tr is straightforward to show that l_0 is left-local, r_0 is right-local, and a function $s : T \tilde{\to} Q$ such that $dom(s) \in \mathfrak{T}$ belongs to Tr if and only if $l_0(s,t) \wedge r_0(s,t)$ for all $t \in dom(s)$. Then $(l_0, r_0) \in LR(Q)$ and (l_0, r_0) is a LR representation of Σ. Moreover, for any LR representation (l,r) of Σ, it is easy to show that $l_0(s,t) \Rightarrow l(s,t)$ and $r_0(s,t) \Rightarrow r(s,t)$ for each $(s,t) \in ST(Q)$ by noting that $s \in Tr$ if and only if $l(s,t') \wedge r(s,t')$ for all $t' \in dom(s)$. Then (l_0, r_0) is a least LR representation of Σ. From the item (1) of this theorem, a least LR representation of Σ is unique, so $l_0 = l_*$ and $r_0 = r_*$. □

For each NCMS Σ denote by $LR_{min}(\Sigma)$ the least LR representation of Σ. Let $LR^*(Q)$ be the set of all pairs $(l,r) \in LR(Q)$ such that for each $(s,t) \in ST(Q)$:

(1) if $l(s,t)$ and t is a non-minimal element of $dom(s)$, then there exists $t' \in (0,t)$ such that $l(s,t'') \wedge r(s,t'')$ for all $t'' \in (t',t)$;
(2) if $r(s,t)$ and t is a non-maximal element of $dom(s)$, then there exists $t' > t$ such that $l(s,t'') \wedge r(s,t'')$ for all $t'' \in (t,t')$.

Theorem 3. *(1) If $(l,r) \in LR^*(Q)$, then there exists a NCMS $\Sigma = (T, Q, Tr)$ such that $(l,r) = LR_{min}(\Sigma)$.*
(2) If $\Sigma = (T, Q, Tr)$ is a NCMS, then $LR_{min}(\Sigma) \in LR^*(Q)$.

Proof (Sketch).

(1) Follows straightforwardly from Theorem 1(1) and Theorem 2.
(2) Let $\Sigma = (T, Q, Tr)$ be a NCMS and $(l^*, r^*) = LR_{min}(\Sigma)$. Then $(l^*, r^*) \in LR(Q)$, (l^*, r^*) is a LR representation of Σ, and from Theorem 2(2) it follows immediately that $(l^*, r^*) \in LR^*(Q)$. □

NCMS can be used as an abstraction of concrete mathematical models. Some examples are given below. More examples (including various discrete-continuous models) can be obtained using Theorem 1.

Example 1. Let $d \in \mathbb{N}$, $Q = \mathbb{R}^d$, and $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$. Let Tr be the set of all functions $s : A \to Q$, $A \in \mathfrak{T}$ such that s is locally absolutely continuous on A (i.e. is absolutely continuous on every segment $[a, b] \subseteq A$) and satisfies the differential equation $\frac{ds(t)}{dt} = f(t, s(t))$ almost everywhere on A (in the sense of Lebesgue measure), i.e. s is a Caratheodory solution.

Then (T, Q, Tr) is a NCMS.

The proof of this follows from the definition of NCMS.

Example 2. Let $(Q, \to)$ be a state transition system, i.e. Q is a set and $\to \subseteq Q \times Q$ is a binary relation. Assume that Q is equipped with the discrete topology (i.e. all subsets are open). Let Tr be the set of all piecewise-constant left-continuous functions $s : A \to Q$ which for all non-maximal $t \in A$ satisfy the condition $s(t+) \downarrow$ and

$$\begin{cases} s(t+) = s(t), & t \notin \mathbb{N}_0, \\ s(t) \to s(t+), & t \in \mathbb{N}_0, \end{cases}$$

where $s(t+)$ denotes the right limit at t (see Fig. 2).

Then (T, Q, Tr) is a NCMS.

The proof of this follows from the definition of NCMS.

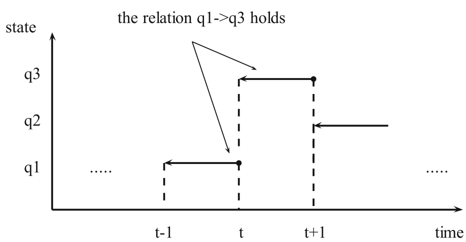


Fig. 2. A trajectory which models an execution of a (discrete-time) state transition system $(Q, \to)$. At the integer time moments the system changes its current state q to a next state q' such that $q \to q'$.

3 Main Result

We will define a formal language in which one can express relations between a finite set of arbitrary distinguished trajectories and a finite set of arbitrary NCMS. In particular, membership of a distinguished trajectory in the set of trajectories of a given NCMS and certain forms of partial coincidence of a distinguished trajectory with one of the trajectories of a given NCMS can be expressed in this language. The partial coincidence of trajectories can be considered as a "pattern matching"-like mechanism for specification of the behavior of dynamical systems and proving their properties.

Taking into account the properties of NCMS mentioned in Sect. 2.2, testing the existence of a trajectory of a NCMS which coincides with a given function on a given time interval or in a neighborhood of a given time moment can be done rather straightforwardly using the least LR representation. This suggests that a language should be able to express in some form the components of the least LR representations of a NCMS. We implement this approach as follows.

Let $\mathcal{T} = \{t_i \mid i \in \mathbb{N}\}$, $\mathcal{C} = \{C_i \mid i \in \mathbb{N}\}$, and $\mathcal{F} = \{F_i \mid i \in \mathbb{N}\}$ be countable pairwise-disjoint sets of variable names (it is assumed that the symbols within each set are identified by their indices).

Let $\mathcal{A}$ be the set of all atomic formulas of the forms $D_i(t_k)$, $E^-_{i,j}(t_k)$, $E^+_{i,j}(t_k)$, $L_{i,j}(t_k)$, $R_{i,j}(t_k)$, $t_i < t_j$, $t_i = t_j$, $t_i \in C_j$, $t_i \in F_j$ $(i, j, k \in \mathbb{N})$, where D_i for all $i \in \mathbb{N}$ and $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$ for all $i, j \in \mathbb{N}$ are distinct unary predicate symbols (identified by their indices) and $<$, $=$, $\in$ are binary predicate symbols.

Let $\mathcal{L}$ be the set of all well-formed (second-order) formulas (a *formal language*) composed of atomic formulas from $\mathcal{A}$, symbols of logical connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), and variable names from $\mathcal{T}, \mathcal{C}, \mathcal{F}$ bound by existential and universal quantifier symbols ($\exists t_i$, $\forall t_i$, $\exists C_i$, $\forall C_i$, $\exists F_i$, $\forall F_i$).

Let $\mathcal{L}^c$ be the set of all sentences (closed formulas) in $\mathcal{L}$.

To define an interpretation of formulas, let us introduce the following notions and notations.

- $Cl(X)$ and $F_\sigma(X)$, where $X \subseteq \mathbb{R}$ is a nonempty set, denote the sets of all closed subsets and F_σ-subsets of X respectively (in the sense of the induced topology on X).
- $\mathcal{S}$ is the set of all algebraic structures of the form

$$(T, <, (D_i)_{i\in\mathbb{N}}, (E^-_{i,j})_{i,j\in\mathbb{N}}, (E^+_{i,j})_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}),$$

  where $T = \mathbb{R}_+$, $<$ is the standard strict order on T, and D_i, $E^-_{i,j}$, $E^+_{i,j}$ $L_{i,j}$, $R_{i,j}$ for all $i, j \in \mathbb{N}$ are unary predicates on T.
- $\models \subseteq \mathcal{S} \times \mathcal{L}$ is a logical validity relation defined as follows: for each $\varPhi \in \mathcal{L}$ and $S = (T, <, (D_i)_{i\in\mathbb{N}}, (E^-_{i,j})_{i,j\in\mathbb{N}}, (E^+_{i,j})_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}) \in \mathcal{S}$, $S \models \varPhi$ if and only if the universal closure of $\varPhi$ holds in the structure S under the interpretation which assumes that the variables $t_1, t_2, ...$ range over T, the variables $C_1, C_2, ...$ range over $Cl(T)$, the variables $F_1, F_2, ...$ range over $F_\sigma(T)$, the symbol $<$ is interpreted as the relation $<$ on T, the symbol $=$ is interpreted as the equality on T, the symbol $\in$ is interpreted as set membership, the symbols D_i, $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$, for $i, j \in \mathbb{N}$ are interpreted as the predicates D_i, $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$ on T respectively.
- $\mathcal{M}$ is the class of all tuples $((s_i)_{i\in\mathbb{N}}, (\varSigma_i)_{i\in\mathbb{N}})$ such that there exists a set Q (*states*) such that:
  (1) for each $i \in \mathbb{N}$ $s_i : T \tilde{\rightarrow} Q$ (*i-th distinguished trajectory*) is a function such that $dom(s_i) \in \mathfrak{T} \cup \{\emptyset\}$ (note that a nowhere defined function is not a trajectory of any NCMS, but we allow it here for convenience);

(2) each Σ_i, $i \in \mathbb{N}$ is a NCMS with the set of states Q (*i-th distinguished NCMS*).

We will call the elements of $\mathcal{M}$ *models* (*of formulas*).

– A structure associated with $M = ((s_i)_{i\in\mathbb{N}}, (\Sigma_i)_{i\in\mathbb{N}}) \in \mathcal{M}$ is a unique tuple $(T, <, (D_i)_{i\in\mathbb{N}}, (E^-_{i,j})_{i,j\in\mathbb{N}}, (E^+_{i,j})_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}) \in \mathcal{S}$ such that:
(1) $D_i(t) \Leftrightarrow t \in dom(s_i)$ for all $i, j \in \mathbb{N}$ and $t \in T$;
(2) $E^-_{i,j}(t) \Leftrightarrow s_i \doteq_{t-} s_j$ for all $i, j \in \mathbb{N}$ and $t \in T$;
(3) $E^+_{i,j}(t) \Leftrightarrow s_i \doteq_{t+} s_j$ for all $i, j \in \mathbb{N}$ and $t \in T$;
(4) $L_{i,j}(t) \Leftrightarrow l^*_i(s_j, t)$ and $R_{i,j}(t) \Leftrightarrow r^*_i(s_j, t)$ for each $i, j \in \mathbb{N}$ and $t \in dom(s_j)$, where $(l^*_i, r^*_i) = LR_{min}(\Sigma_i)$;
(5) $\neg L_{i,j}(t)$ and $\neg R_{i,j}(t)$ for all $i, j \in \mathbb{N}$ and $t \in T$ such that $t \notin dom(s_j)$.
This structure is denoted as $St(M)$.
– $\models^m \subseteq \mathcal{M} \times \mathcal{L}$ is a logical validity relation defined as follows: for each $\Phi \in \mathcal{L}$ and $M \in \mathcal{M}$, $M \models^m \Phi$ if and only if $St(M) \models \Phi$. This is interpreted as follows: Φ is assumed to be valid in a model M ($M \models^m \Phi$), if Φ is valid in the structure associated with M.
– $Th = \{\Phi \in \mathcal{L}^c \mid \forall M \in \mathcal{M}\ M \models^m \Phi\}$ is a formal theory which consists of all sentences valid in all models.

In short, Th consists of the sentences Φ valid in all models $M \in \mathcal{M}$ which consist of a sequence of distinguished trajectories $(s_i)_{i\in\mathbb{N}}$ and a sequence of distinguished NCMS $(\Sigma_i)_{i\in\mathbb{N}}$ under the assumption that in Φ the variables $t_1, t_2, ...$ range over T, $C_1, C_2, ...$ range over $Cl(T)$, $F_1, F_2, ...$ range over $F_\sigma(T)$, the symbols $<$ and $=$ mean the standard strict order and equality on T, $\in$ means set membership, the atomic formula $D_i(t_k)$ means $t_k \in dom(s_i)$, $E^-_{i,j}(t_k)$ means $s_i \doteq_{t_k-} s_j$, $E^+_{i,j}(t_k)$ means $s_i \doteq_{t_k+} s_j$, and $L_{i,j}(t_k)$, $R_{i,j}(t_k)$ mean that $s_j(t_k)\downarrow$ and $l^*_i(s_j, t_k)$ and $r^*_i(s_j, t_k)$ hold, where $(l^*_i, r^*_i) = LR_{min}(\Sigma_i)$.

Informally, the formulas in Th express valid relations between trajectories and NCMS that hold for arbitrary distinguished trajectories and arbitrary NCMS.

Theorem 4. *The formal theory Th is decidable.*

We will prove this theorem in the next section.

4 Proof of Decidability

In this section we give a proof of Theorem 4 using a series of lemmas.

Definition 9. *A predicate $P : T \to Bool$ is called*

– *right-stable, if for each $t \in T$, $P(t)$ implies that there exists $t' > t$ such that $P(t'')$ holds for all $t'' \in [t, t')$;*
– *left-stable, if for each $t \in T\backslash\{0\}$, $P(t)$ implies that there exists $t' < t$ such that $P(t'')$ holds for all $t'' \in (t', t]$.*

Note that a *truth set* of a predicate is the set of arguments on which it holds.

Lemma 1. *(1) The truth set of a right-stable predicate is an* F_σ*-set.*
(2) The truth set of a left-stable predicate is an F_σ*-set.*

Proof. (1) Let $P : T \to Bool$ be right-stable and $A = \{t \mid P(t)\}$ be the truth set of P. Then for each $t \in T$ there exists $b(t) > t$ such that $P(t)$ implies $P(t')$ for all $t' \in [t, b(t))$. Then $A = \bigcup_{t \in A}[t, b(t))$ is an open set in the sense of the Sorgenfrey (right half-open interval) topology on $\mathbb{R}$ [53]. The latter topology is hereditarily Lindelöf [53], which implies that every open cover of A (as an open subspace of the Sorgenfrey line) has a countable sub-cover. The sets $\{[t, b(t)) \mid t \in A\}$ form an open cover of A in this sense, so there is a countable subset $C \subseteq \{[t, b(t)) \mid t \in A\}$ such that $A = \bigcup C$. All elements of C are F_σ sets in the sense of the topology on T, so A is an F_σ-set in the same sense.

(2) Analogous to the proof of the item (1). □

One of the consequences of the theorem on the decidability of $S2S$ by M.O. Rabin is decidability of a second order theory of a real segment $[0, 1]$ with quantification over closed subsets and over F_σ-subsets [51].

More specifically, let $\mathcal{A}^2_<$ be the set of atomic formulas of the forms $t_i < t_j$, $t_i = t_j$, $t_i \in C_j$, $t_i \in F_j$ $(i, j \in \mathbb{N})$, where $t_i \in \mathcal{T}, C_i \in \mathcal{C}, F_i \in \mathcal{F}$ and $<, =, \in$ are binary predicate symbols.

Let $\mathcal{L}^2_<$ be the set of all well-formed (second-order) formulas composed of atomic formulas from $\mathcal{A}^2_<$, symbols of logical connectives ($\neg$, $\wedge$, $\vee$, $\to$, $\leftrightarrow$), and variable names from $\mathcal{T}$, $\mathcal{C}$, $\mathcal{F}$ bound by existential and universal quantifier symbols ($\exists t_i$, $\forall t_i$, $\exists C_i$, $\forall C_i$, $\exists F_i$, $\forall F_i$).

For each nonempty set $X \subseteq \mathbb{R}$ let $Th^2_<(X)$ be the set of all sentences (closed formulas) in $\mathcal{L}^2_<$ which are valid in the structure $(X, <_X)$, where $<_X$ is the restriction of the standard order on reals to X, under the interpretation which assumes that $t_1, t_2, ...$ range over X, the variables $C_1, C_2, ...$ range over $Cl(X)$, $F_1, F_2, ...$ range over $F_\sigma(X)$, the symbol $<$ is interpreted as $<_X$, the symbol $=$ is interpreted as equality on X, and the symbol $\in$ is interpreted as set membership.

Lemma 2. *The theory* $Th^2_<(T)$ *is decidable.*

Proof (Sketch). The theory $Th^2_<([0, 1])$ is known to be decidable [51, Theorem 2.9]. Using this result it is straightforward to prove that $Th^2_<([0, 1))$ is decidable, because in the language $\mathcal{L}^2_<$ one can express the following predicates P_1, P_2, the truth sets of the interpretations of which are $Cl([0, 1))$ and $F_\sigma([0, 1))$:

$$P_1(C_1) := \forall t_1 \ (t_1 \in C_1 \to (\exists t_2 \ t_1 < t_2));$$
$$P_2(F_1) := \forall t_1 \ (t_1 \in F_1 \to (\exists t_2 \ t_1 < t_2)).$$

Taking into account that any continuous increasing bijection $[0, +\infty) \to [0, 1)$ is a homeomorphism between T and $[0, 1)$ as topological spaces and is an order-isomorphism between T and $[0, 1)$ as ordered sets, it is straightforward to show that $Th^2_<([0, 1)) = Th^2_<(T)$. Thus $Th^2_<(T)$ is decidable. □

Let us fix some injective computable functions $\mathbf{d} : \mathbb{N} \to \mathbb{N}$, $\mathbf{en} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, $\mathbf{ep} : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, $\mathbf{l} : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, $\mathbf{r} : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, $\mathbf{e} : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, and $\mathbf{f} : \mathbb{N} \to \mathbb{N}$ with pairwise disjoint ranges. We will use them to map the predicate symbols D_i, $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$ of $\mathcal{L}$ to distinct F_σ-variable names $F_{\mathbf{d}(i,j)}$, $F_{\mathbf{en}(i,j)}$, $F_{\mathbf{ep}(i,j)}$, $F_{\mathbf{l}(i,j)}$, $F_{\mathbf{r}(i,j)}$ of $\mathcal{L}^2_<$, to have distinct intermediate F_σ-variable names $F_{\mathbf{e}(i,j)}$, $i, j \in \mathbb{N}$, and to map variable names F_i of $\mathcal{L}$ to variable names $F_{\mathbf{f}(i)}$ of $\mathcal{L}^2_<$.

For each $i, j, k \in \mathbb{N}$ let us define the following $\mathcal{L}^2_<$-formulas:

$$\begin{aligned}
&Dom_j(F_{\mathbf{d}(j)}) := \\
&\quad ((\neg\exists t_1\ t_1 \in F_{\mathbf{d}(j)}) \vee (\exists t_1\exists t_2(t_1 \in F_{\mathbf{d}(j)} \wedge t_2 \in F_{\mathbf{d}(j)} \wedge t_1 < t_2)))\wedge \\
&\quad \wedge \forall t_1\forall t_2\forall t_3(t_1 \in F_{\mathbf{d}(j)} \wedge t_2 \in F_{\mathbf{d}(j)} \wedge t_1 < t_3 \wedge t_3 < t_2 \to t_3 \in F_{\mathbf{d}(j)})
\end{aligned}$$

$$\begin{aligned}
&Eq_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{e}(i,i)}, F_{\mathbf{e}(i,j)}, F_{\mathbf{e}(j,i)}, F_{\mathbf{e}(j,k)}, F_{\mathbf{e}(i,k)}, F_{\mathbf{d}(j)}, F_{\mathbf{en}(i,j)}, F_{\mathbf{ep}(i,j)}) := \\
&\quad \forall t_1\,(t_1 \in F_{\mathbf{d}(i)} \to t_1 \in F_{\mathbf{e}(i,i)})\wedge \\
&\quad \wedge\forall t_1\,(t_1 \in F_{\mathbf{e}(i,j)} \to t_1 \in F_{\mathbf{e}(j,i)})\wedge \\
&\quad \wedge\forall t_1\,(t_1 \in F_{\mathbf{e}(i,j)} \wedge t_1 \in F_{\mathbf{e}(j,k)} \to t_1 \in F_{\mathbf{e}(i,k)}))\wedge \\
&\quad \wedge\forall t_1\,(t_1 \in F_{\mathbf{e}(i,j)} \to t_1 \in F_{\mathbf{d}(i)} \wedge t_1 \in F_{\mathbf{d}(j)})\wedge \\
&\quad \wedge\forall t_1\,(t_1 \in F_{\mathbf{en}(i,j)} \leftrightarrow \\
&\qquad \leftrightarrow \exists t_2(t_2 < t_1 \wedge \forall t_3(t_2 < t_3 \wedge (t_3 < t_1 \vee t_3 = t_1) \to t_3 \in F_{\mathbf{e}(i,j)})))\wedge \\
&\quad \wedge\forall t_1\,(t_1 \in F_{\mathbf{ep}(i,j)} \leftrightarrow \\
&\qquad \leftrightarrow \exists t_2(t_1 < t_2 \wedge \forall t_3((t_1 < t_3 \vee t_1 = t_3) \wedge t_3 < t_2 \to t_3 \in F_{\mathbf{e}(i,j)})))
\end{aligned}$$

$$\begin{aligned}
&Lloc_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{en}(i,j)}, F_{\mathbf{l}(k,i)}, F_{\mathbf{l}(k,j)}) := \\
&\quad \forall t_1\,(t_1 \in F_{\mathbf{en}(i,j)} \to (t_1 \in F_{\mathbf{l}(k,i)} \leftrightarrow t_1 \in F_{\mathbf{l}(k,j)}))\wedge \\
&\quad \wedge \forall t_1\,(t_1 \in F_{\mathbf{d}(i)} \wedge (\neg\exists t_2\,(t_2 < t_1 \wedge t_2 \in F_{\mathbf{d}(i)})) \to t_1 \in F_{\mathbf{l}(k,i)})\wedge \\
&\quad \wedge \forall t_1\,(\neg t_1 \in F_{\mathbf{d}(i)} \to \neg t_1 \in F_{\mathbf{l}(k,i)})
\end{aligned}$$

$$\begin{aligned}
&Rloc_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{ep}(i,j)}, F_{\mathbf{r}(k,i)}, F_{\mathbf{r}(k,j)}) := \\
&\quad \forall t_1\,(t_1 \in F_{\mathbf{ep}(i,j)} \to (t_1 \in F_{\mathbf{r}(k,i)} \leftrightarrow t_1 \in F_{\mathbf{r}(k,j)}))\wedge \\
&\quad \wedge \forall t_1\,(t_1 \in F_{\mathbf{d}(i)} \wedge (\neg\exists t_2\,(t_1 < t_2 \wedge t_2 \in F_{\mathbf{d}(i)})) \to t_1 \in F_{\mathbf{r}(k,i)})\wedge \\
&\quad \wedge \forall t_1\,(\neg t_1 \in F_{\mathbf{d}(i)} \to \neg t_1 \in F_{\mathbf{r}(k,i)})
\end{aligned}$$

$$\begin{aligned}
&Min_{i,j}(F_{\mathbf{d}(j)}, F_{\mathbf{l}(i,j)}, F_{\mathbf{r}(i,j)}) := \\
&\quad \forall t_1\ (t_1 \in F_{\mathbf{l}(i,j)} \wedge t_1 \in F_{\mathbf{d}(j)} \wedge \exists t_2(t_2 < t_1 \wedge t_2 \in F_{\mathbf{d}(j)}) \to \\
&\qquad \to \exists t_3(t_3 < t_1 \wedge \forall t_4(t_3 < t_4 \wedge t_4 < t_1 \to t_4 \in F_{\mathbf{l}(i,j)} \wedge t_4 \in F_{\mathbf{r}(i,j)})))\wedge \\
&\quad \wedge \forall t_1\ (t_1 \in F_{\mathbf{r}(i,j)} \wedge t_1 \in F_{\mathbf{d}(j)} \wedge \exists t_2(t_1 < t_2 \wedge t_2 \in F_{\mathbf{d}(j)})) \to \\
&\qquad \to \exists t_3(t_1 < t_3 \wedge \forall t_4(t_1 < t_4 \wedge t_4 < t_3 \to t_4 \in F_{\mathbf{l}(i,j)} \wedge t_4 \in F_{\mathbf{r}(i,j)}))
\end{aligned}$$

Informally, Dom_j expresses that $F_{\mathbf{d}(j)}$ is the domain of a distinguished trajectory j (the empty set or a non-degenerate interval), $Eq_{i,j,k}$ is intended to express the properties of the relations of left and right coincidence of trajectories, $Lloc_{i,j,k}$ and $Rloc_{i,j,k}$ are intended to express left-locality and right-locality of predicates, and $Min_{i,j}$ is intended to express the properties of the members of $LR^*(Q)$ for arbitrary Q (least LR representations of NCMS).

For each $\Phi \in \mathcal{L}$ denote by $\mathfrak{m}(\Phi)$ the maximal value of among all indices i, j of the predicate symbols of the forms D_i, $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$ which appear in Φ, or 1, if no such symbols appears in Φ.

For each $\Phi \in \mathcal{L}^c$ let us define an associated formula $Trans(\Phi)$ as follows. Let $\Phi \in \mathcal{L}^c$ and $u_1, u_2, ..., u_m$ be an increasing sequence of all elements of the set

$$\{\mathbf{d}(j), \mathbf{l}(i,j), \mathbf{r}(i,j), \mathbf{ep}(i,j), \mathbf{en}(i,j), \mathbf{e}(i,j)\} \mid i,j \in \{1,2,...,\mathfrak{m}(\Phi)\}\},$$

and Φ' be the formula obtained from Φ by renaming all variable names of the form F_i to $F_{\mathbf{f}(i)}$ and subsequent replacement of all atomic sub-formulas of Φ of the forms $D_i(t_k)$, $E^-_{i,j}(t_k)$, $E^+_{i,j}(t_k)$, $L_{i,j}(t_k)$, $R_{i,j}(t_k)$ $(i,j,k \in \mathbb{N})$ with $t_k \in F_{\mathbf{d}(i,j)}$, $t_k \in F_{\mathbf{en}(i,j)}$, $t_k \in F_{\mathbf{ep}(i,j)}$, $t_k \in F_{\mathbf{l}(i,j)}$, $t_k \in F_{\mathbf{r}(i,j)}$ respectively.

Then $Trans(\Phi)$ denotes the following formula:

$$\forall F_{u_1} \forall F_{u_2} ... \forall F_{u_m} \bigwedge_{j=1}^{\mathfrak{m}(\Phi)} Dom_j(F_{\mathbf{d}(j)}) \wedge \bigwedge_{i=1}^{\mathfrak{m}(\Phi)} \bigwedge_{j=1}^{\mathfrak{m}(\Phi)} \bigwedge_{k=1}^{\mathfrak{m}(\Phi)} ($$
$$Eq_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{e}(i,i)}, F_{\mathbf{e}(i,j)}, F_{\mathbf{e}(j,i)}, F_{\mathbf{e}(j,k)}, F_{\mathbf{e}(i,k)}, F_{\mathbf{d}(j)}, F_{\mathbf{en}(i,j)}, F_{\mathbf{ep}(i,j)}) \wedge$$
$$\wedge\ Lloc_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{en}(i,j)}, F_{\mathbf{l}(k,i)}, F_{\mathbf{l}(k,j)}) \wedge$$
$$\wedge\ Rloc_{i,j,k}(F_{\mathbf{d}(i)}, F_{\mathbf{ep}(i,j)}, F_{\mathbf{r}(k,i)}, F_{\mathbf{r}(k,j)})) \wedge$$
$$\wedge \bigwedge_{i=1}^{\mathfrak{m}(\Phi)} \bigwedge_{j=1}^{\mathfrak{m}(\Phi)} Min_{i,j}(F_{\mathbf{d}(j)}, F_{\mathbf{l}(i,j)}, F_{\mathbf{r}(i,j)}) \to \Phi'$$

Obviously, $Trans(\Phi)$ is a closed formula in $\mathcal{L}^2_<$.

Informally, $Trans$ translates $\mathcal{L}$-sentences into $\mathcal{L}^2_<$-sentences for the purpose of reduction of testing membership in Th to testing membership in $Th^2_<(T)$.

Denote $I_0 = \mathbb{N}$ and $I_n = \{1, 2, ..., n\}$ for each $n \in \mathbb{N}$.

For each $n \in \mathbb{N} \cup \{0\}$ let $\mathcal{S}_n$ be the set of all structures

$$(T, <, (D_i)_{i \in \mathbb{N}}, (E^-_{i,j})_{i,j \in \mathbb{N}}, (E^+_{i,j})_{i,j \in \mathbb{N}}, (L_{i,j})_{i,j \in \mathbb{N}}, (R_{i,j})_{i,j \in \mathbb{N}}) \in \mathcal{S}$$

such that there exists an indexed family $(E_{i,j})_{i,j \in I_n}$ of predicates on T which satisfy the following conditions:

(1) for each $i, j \in I_n$ the truth sets of D_i, $E^-_{i,j}$, $E^+_{i,j}$, $L_{i,j}$, $R_{i,j}$, $E_{i,j}$ are in $F_\sigma(T)$;
(2) for each $i \in I_n$ the truth set of D_i belongs to $\mathfrak{T} \cup \{\emptyset\}$;
(3) for each $i, j, k \in I_n$ and $t \in T$ the following properties of $E_{i,j}$ hold:
 - $D_i(t) \Rightarrow E_{i,i}(t)$;
 - $E_{i,j}(t) \Rightarrow E_{j,i}(t)$;
 - $E_{i,j}(t) \wedge E_{j,k}(t) \Rightarrow E_{i,k}(t)$;
 - $E_{i,j}(t) \Rightarrow D_i(t) \wedge D_j(t)$;

- $E^-_{i,j}(t) \Leftrightarrow \exists t' \in [0,t)\ \forall t'' \in (t',t]\ E_{i,j}(t'')$;
- $E^+_{i,j}(t) \Leftrightarrow \exists t' > t\ \forall t'' \in [t,t')\ E_{i,j}(t'')$;

(4) for each $i,j,k \in I_n$ and $t \in T$ the following properties of $L_{k,i}$ hold:
- $E^-_{i,j}(t) \Rightarrow (L_{k,i}(t) \Leftrightarrow L_{k,j}(t))$;
- if t is a minimal element of $\{t' \mid D_i(t')\}$, then $L_{k,i}(t)$;
- $\neg D_i(t) \Rightarrow \neg L_{k,i}(t)$;

(5) for each $i,j,k \in I_n$ and $t \in T$ the following properties of $R_{k,i}$ hold:
- $E^+_{i,j}(t) \Rightarrow (R_{k,i}(t) \Leftrightarrow R_{k,j}(t))$;
- if t is a maximal element of $\{t' \mid D_i(t')\}$, then $R_{k,i}(t)$;
- $\neg D_i(t) \Rightarrow \neg R_{k,i}(t)$;

(6) for each $i,j \in I_n$ and $t \in T$ the following holds:
- if $L_{i,j}(t)$ and t is a non-minimal element of $\{t' \mid D_j(t')\}$, then there exists $t' \in [0,t)$ such that $L_{i,j}(t'') \wedge R_{i,j}(t'')$ for all $t'' \in (t',t)$;
- if $R_{i,j}(t)$ and t is a non-maximal element of $\{t' \mid D_j(t')\}$, then there exists $t' > t$ such that $L_{i,j}(t'') \wedge R_{i,j}(t'')$ for all $t'' \in (t,t')$.

Lemma 3. *Let $\Phi \in \mathcal{L}^c$. Then $Trans(\Phi) \in Th^2_<(T)$ if and only if $S \models \Phi$ for all $S \in \mathcal{S}_{\mathfrak{m}(\Phi)}$.*

Proof (Sketch). Follows straightforwardly from the definition of $Trans$, $\mathcal{S}_{\mathfrak{m}(\Phi)}$, and the interpretation of $\mathcal{L}^2_<$ formulas. □

Lemma 4. *Let $\Phi \in \mathcal{L}$, $n \in \mathbb{N}$, $n \geq \mathfrak{m}(\Phi)$. Then $S \models \Phi$ for all $S \in \mathcal{S}_n$ if and only if $S \models \Phi$ for all $S \in \mathcal{S}_0$.*

Proof (Sketch).

"If": Assume that $S \models \Phi$ for all $S \in \mathcal{S}_n$. Using the definition of $\mathcal{S}_0$, it is easy to check that $\mathcal{S}_0 \subseteq \mathcal{S}_n$. Then $S \models \Phi$ for all $S \in \mathcal{S}_0$.

"Only if": Assume that $S \models \Phi$ for all $S \in \mathcal{S}_0$. Let

$$S' = (T, <, (D_i)_{i\in\mathbb{N}}, (E^-_{i,j})_{i,j\in\mathbb{N}}, (E^+_{i,j})_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}) \in \mathcal{S}_n$$

be an arbitrary element. Then there exists a family $(E_{i,j})_{i,j\in I_n}$ of predicates on T such that the properties (1)–(6) of $\mathcal{S}_n$ hold for S' and $(E_{i,j})_{i,j\in I_n}$.

Let us prove that $S' \models \Phi$.

For each $i,j \in I_n$ let $D'_i = D_i$, $E'^-_{i,j} = E^-_{i,j}$, $E'^+_{i,j} = E^+_{i,j}$. For each $i,j \in \mathbb{N}$ such that $(i,j) \notin I_n \times I_n$ let us define unary predicates D'_i, $E'^-_{i,j}$, $E'^+_{i,j}$ on T such that $\neg D'_i(t)$, $\neg E'^-_{i,j}(t)$, $\neg E'^+_{i,j}(t)$ for all $t \in T$.

For each $i,j \in \mathbb{N}$ let us define unary predicates $L'_{i,j}$, $R'_{i,j}$ on T such that
- if $(i,j) \in I_n \times I_n$, then $L'_{i,j} = L_{i,j}$, otherwise for all $t \in T$, $L'_{i,j}(t)$ if and only if t is a minimal element of the truth set of D'_j;
- if $(i,j) \in I_n \times I_n$, then $R'_{i,j} = R_{i,j}$, otherwise for all $t \in T$, $R'_{i,j}(t)$ if and only if t is a maximal element of the truth set of D'_j.

Let

$$S'' = (T, <, (D'_i)_{i\in\mathbb{N}}, (E'^{-}_{i,j})_{i,j\in\mathbb{N}}, (E'^{+}_{i,j})_{i,j\in\mathbb{N}}, (L'_{i,j})_{i,j\in\mathbb{N}}, (R'_{i,j})_{i,j\in\mathbb{N}}).$$

Obviously, $S'' \in \mathcal{S}$. The structures S' and S'' have the same carrier set. Also, because $n \geq \mathfrak{m}(\Phi)$, the symbols of all predicates which appear in Φ have the same interpretations in S' and S'', so $S'' \models \Phi$ if and only if $S' \models \Phi$.

Let us show that $S'' \in \mathcal{S}_0$. Let $(E'_{i,j})_{i,j\in I_0}$ be an indexed family of predicates such that for each $i, j \in \mathbb{N}$, if $(i, j) \in I_0 \times I_0$, then $E'_{i,j} = E_{i,j}$, and if $(i, j) \notin I_0 \times I_0$, then $\neg E'_{i,j}(t)$ for all $t \in T$.

It is easy to check that the properties (1)–(3) and (6) of $\mathcal{S}_0$ hold for S'' and $(E'_{i,j})_{i,j\in I_0}$. Let us prove the property (4) of $\mathcal{S}_0$ for S'' and $(E'_{i,j})_{i,j\in I_0}$.

Let $i, j, k \in I_0$ and $t \in T$.

- Let us show that $E'^{-}_{i,j}(t) \Rightarrow (L'_{k,i}(t) \Leftrightarrow L'_{k,j}(t))$. Assume that $E'^{-}_{i,j}(t)$. Then $i, j \in I_n$, so $E^{-}_{i,j}(t)$. If $k \in I_n$, then $L_{k,i}(t) \Leftrightarrow L_{k,j}(t)$, because the property (4) of $\mathcal{S}_n$ holds for S' and $(E_{i,j})_{i,j\in I_n}$, so $L'_{k,i}(t) \Leftrightarrow L'_{k,j}(t)$. Otherwise, $k \notin I_n$ and from the property (3) of $\mathcal{S}_n$ for S' and $(E_{i,j})_{i,j\in I_n}$ it follows that t is not a minimal element of the truth set of $E^{-}_{i,j}$ and is not a minimal element of the truth sets of $D_i = D'_i$ and $D_j = D'_j$. Then $\neg L'_{k,i}(t)$ and $\neg L'_{k,j}(t)$, whence $L'_{k,i}(t) \Leftrightarrow L'_{k,j}(t)$. We conclude that $E'^{-}_{i,j}(t) \Rightarrow (L'_{k,i}(t) \Leftrightarrow L'_{k,j}(t))$.
- Let us show that if t is a minimal element of $\{t' \mid D'_i(t')\}$, then $L'_{k,i}(t)$. Let t be a minimal element of $\{t' \mid D'_i(t')\}$. Then $D'_i(t)$, so $i \in I_n$ and t is a minimal element of $\{t' \mid D_i(t')\}$. If $k \in I_n$, then $L_{k,i}(t)$ by the property (4) of $\mathcal{S}_n$ for S' and $(E_{i,j})_{i,j\in I_n}$, so $L'_{k,i}(t)$. Otherwise, $k \notin I_n$ and t is a minimal element of $\{t' \mid D'_i(t')\}$, so $L'_{k,i}(t)$. In both cases, $L'_{k,i}(t)$.
- Let us show that $\neg D'_i(t) \Rightarrow \neg L'_{k,i}(t)$. Assume that $\neg D'_i(t)$. If $i \in I_n$ and $k \in I_n$, then $\neg D_i(t) \Rightarrow \neg L_{k,i}(t)$, $D_i = D'_i$, and $L_{k,i} = L'_{k,i}$, so $\neg L'_{k,i}(t)$. Otherwise, $i \notin I_n$ or $k \notin I_n$, whence $\neg L'_{k,i}(t)$, because t is not an element of the truth set of D'_i. In both cases, $\neg L'_{k,i}(t)$.

Thus the property (4) of $\mathcal{S}_0$ holds for S'' and $(E'_{i,j})_{i,j\in I_0}$. The property (5) of $\mathcal{S}_0$ can be proven for S'' and $(E'_{i,j})_{i,j\in I_0}$ analogously.

We conclude that $S'' \in \mathcal{S}_0$. Then $S'' \models \Phi$ by assumption. Thus $S' \models \Phi$. Because S' is arbitrary, we have $S' \models \Phi$ for all $S' \in \mathcal{S}_n$. □

Lemma 5. *$St(M) \in \mathcal{S}_0$ for each $M \in \mathcal{M}$.*

Proof. Let $M = ((s_i)_{i\in\mathbb{N}}, (\Sigma_i)_{i\in\mathbb{N}}) \in \mathcal{M}$ and Q be the (common) set of states of all Σ_i, $i \in \mathbb{N}$. Let

$$S = St(M) = (T, <, (D_i)_{i\in\mathbb{N}}, (E^{-}_{i,j})_{i,j\in\mathbb{N}}, (E^{+}_{i,j})_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}).$$

Then $S \in \mathcal{S}$. Let us show that $S \in \mathcal{S}_0$. For each $i \in \mathbb{N}$ let $(l^*_i, r^*_i) = LR_{min}(\Sigma_i)$. By Theorem 3(2), $(l^*_i, r^*_i) \in LR^*(Q)$. By the definition of $St(M)$, for all $i, j \in \mathbb{N}$ and $t \in T$, $D_i(t) \Leftrightarrow t \in dom(s_i)$, $E^{-}_{i,j}(t) \Leftrightarrow s_i \doteq_{t-} s_j$, and $E^{+}_{i,j}(t) \Leftrightarrow$

$s_i \dot{=}_{t+} s_j$. Moreover, if $s_j(t) \downarrow$, then $L_{i,j}(t) \Leftrightarrow l_i^*(s_j, t)$ and $R_{i,j}(t) \Leftrightarrow r_i^*(s_j, t)$, and if $s_j(t) \uparrow$, then $\neg L_{i,j}(t)$ and $\neg R_{i,j}(t)$.

For each $t \in T$ let $V(t) = \{(i,j) \in \mathbb{N} \times \mathbb{N} \mid E_{i,j}^-(t) \vee E_{i,j}^+(t)\}$ and $V^T(t)$ is a transitive closure of $V(t)$. Let $(E_{i,j})_{i,j\in\mathbb{N}}$ be an indexed family of predicates on T such that $E_{i,j}(t) \Leftrightarrow (i,j) \in V^T(t)$ for all $i, j \in \mathbb{N}$ and $t \in T$.

Note that for each $t \in T$, $V(t)$ is a symmetric binary relation, so $V^T(t)$ is also a symmetric binary relation.

It is easy to check that the properties (2) and (3) of $\mathcal{S}_0$ hold for S and $(E_{i,j})_{i,j\in\mathbb{N}}$. Taking into account that for each $i \in \mathbb{N}$, l_i^* is left-local and r_i^* is right-local, the properties (4) and (5) of $\mathcal{S}_0$ also hold for S and $(E_{i,j})_{i,j\in\mathbb{N}}$. The property (6) of $\mathcal{S}_0$ can be easily checked for S and $(E_{i,j})_{i,j\in\mathbb{N}}$ by taking into account that $(l_i^*, r_i^*) \in LR^*(Q)$ for all $i \in \mathbb{N}$.

Let us prove the remaining property (1) of $\mathcal{S}_0$ for S and $(E_{i,j})_{i,j\in\mathbb{N}}$, i.e. that the truth sets of D_i, $E_{i,j}^-$, $E_{i,j}^+$, $L_{i,j}$, $R_{i,j}$, $E_{i,j}$ belong to $F_\sigma(T)$ for all $i, j \in \mathbb{N}$.

Let us fix $i, j \in \mathbb{N}$. We have $\{t \mid D_i(t)\} = dom(s_i) \in \mathfrak{T} \cup \{\emptyset\} \subset F_\sigma(T)$.

It is easy to see that $E_{i,j}^-$ is a left-stable predicate and $E_{i,j}^+$ is a right-stable predicate. Then by Lemma 1, the truth sets of $E_{i,j}^-$ and $E_{i,j}^+$ belong to $F_\sigma(T)$.

Let P_j be a predicate on T such that $P_j(t)$ if and only if t is a non-minimal element of $dom(s_j)$. Let A, B be predicates on T such that $A(t) \Leftrightarrow L_{i,j}(t) \wedge P_j(t)$, $B(t) \Leftrightarrow L_{i,j}(t) \wedge \neg P_j(t)$ for all $t \in T$. Because $(l_i^*, r_i^*) \in LR^*(Q)$, A is a left-stable predicate, so by Lemma 1 the truth set of A belongs to $F_\sigma(T)$. Moreover, because $\neg L_{i,j}(t)$ for all $t \in T \backslash dom(s_j)$, $B(t)$ implies that t is a minimal element of $dom(s_j)$, so the truth set of B is either empty or a singleton set, so it is in $F_\sigma(T)$. Then $\{t \mid L_{i,j}(t)\} = \{t \mid A(t)\} \cup \{t \mid B(t)\} \in F_\sigma(T)$.

By analogy it is easy to prove that $\{t \mid R_{i,j}(t)\} \in F_\sigma(T)$.

Let us show that $\{t \mid E_{i,j}(t)\} \in F_\sigma(T)$. Let

$$W = \{\{t' \mid E_{i,j}^-(t')\} \mid i,j \in \mathbb{N}\} \cup \{\{t' \mid E_{i,j}^+(t')\} \mid i,j \in \mathbb{N}\}.$$

From the arguments mentioned it follows that $W \subseteq F_\sigma(T)$. Using the definition of $E_{i,j}$ it is easy to check that $\{t \mid E_{i,j}(t)\}$ can be represented as a countable union of finite intersections of elements of W. As finite intersections of F_σ-sets are F_σ-sets, it follows that $\{t \mid E_{i,j}(t)\} \in F_\sigma(T)$.

We conclude that the truth sets of D_i, $E_{i,j}^-$, $E_{i,j}^+$, $L_{i,j}$, $R_{i,j}$, $E_{i,j}$ are in $F_\sigma(T)$. The properties (1)-(6) of $\mathcal{S}_0$ hold for S and $(E_{i,j})_{i,j\in\mathbb{N}}$, so $S = St(M) \in \mathcal{S}_0$. $\square$

Lemma 6. *For each $S \in \mathcal{S}_0$ there exists $M \in \mathcal{M}$ such that $S = St(M)$.*

Proof (Sketch). Let

$$S = (T, <, (D_i)_{i\in\mathbb{N}}, (E_{i,j}^-)_{i,j\in\mathbb{N}}, (E_{i,j}^+)_{i,j\in\mathbb{N}}, (L_{i,j})_{i,j\in\mathbb{N}}, (R_{i,j})_{i,j\in\mathbb{N}}) \in \mathcal{S}_0.$$

Then $S \in \mathcal{S}$ and there exists an indexed family $(E_{i,j})_{i,j\in\mathbb{N}}$ of predicates on T such that the properties (1)–(6) of $\mathcal{S}_0$ hold for S and $(E_{i,j})_{i,j\in\mathbb{N}}$.

Let $Q = 2^{\mathbb{N}}$ and for each $i \in \mathbb{N}$, $s_i : T \tilde{\to} Q$ be a function such that $s_i(t) \downarrow = \{j \in \mathbb{N} \mid E_{i,j}(t)\}$ for each $t \in T$ such that $D_i(t)$ and $s_i(t) \uparrow$ for each $t \in T$ such that $\neg D_i(t)$. For each $k \in \mathbb{N}$ let $l_k : ST(Q) \to Bool$ and $r_k : ST(Q) \to Bool$ be predicates such that for each $(s,t) \in ST(Q)$:

– $l_k(s,t)$ if and only if either t is a minimal element of $dom(s)$, or there exists $i \in \mathbb{N}$ such that $s \doteq_{t-} s_i$ and $L_{k,i}(t)$.
– $r_k(s,t)$ if and only if either t is a maximal element of $dom(s)$, or there exists $i \in \mathbb{N}$ such that $s \doteq_{t+} s_i$ and $R_{k,i}(t)$.

It follows immediately that l_k is left-local and r_k is right-local. Then $(l_k, r_k) \in LR(Q)$. For each $k \in \mathbb{N}$ let $Tr_k = \{s : A \to Q \mid A \in \mathfrak{T} \wedge (\forall t \in A\ l_k(s,t) \wedge r_k(s,t))\}$ and $\Sigma_k = (T, Q, Tr_k)$. Let $M = ((s_i)_{i\in\mathbb{N}}, (\Sigma_k)_{k\in\mathbb{N}})$. The property (2) of $\mathcal{S}_0$ for S and $(E_{i,j})_{i,j\in\mathbb{N}}$ implies that $dom(s_i) \in \mathfrak{T} \cup \{\emptyset\}$ for all $i \in \mathbb{N}$ and Theorem 1(1) implies that Σ_k is a NCMS with the set of states Q for all $k \in \mathbb{N}$. Thus $M \in \mathcal{M}$.

For each $k \in \mathbb{N}$ let $(l_k^*, r_k^*) = LR_{min}(\Sigma_k)$.

Using the properties of $\mathcal{S}_0$ it is not difficult to check that for each $i,j \in \mathbb{N}$ and $t \in T$, $D_i(t) \Leftrightarrow t \in dom(s_i)$, $E^-_{i,j}(t) \Leftrightarrow s_i \doteq_{t-} s_j$, $E^+_{i,j}(t) \Leftrightarrow s_i \doteq_{t+} s_j$, and also if $s_j(t)\downarrow$, then $L_{i,j}(t) \Leftrightarrow l_i^*(s_j,t)$ and $R_{i,j}(t) \Leftrightarrow r_i^*(s_j,t)$ and if $s_j(t)\uparrow$, then $\neg L_{i,j}(t)$ and $\neg R_{i,j}(t)$. Then $S = St(M)$. □

Lemma 7. *Let $\Phi \in \mathcal{L}^c$. Then $Trans(\Phi) \in Th^2_<(T)$ if and only if $\Phi \in Th$.*

Proof. By Lemma 3, $Trans(\Phi) \in Th^2_<(T)$ if and only if $S \models \Phi$ for all $S \in \mathcal{S}_{\mathfrak{m}(\Phi)}$. By Lemma 4, $S \models \Phi$ for all $S \in \mathcal{S}_{\mathfrak{m}(\Phi)}$ if and only if $S \models \Phi$ for all $S \in \mathcal{S}_0$. Lemmas 5, 6 imply that $S \models \Phi$ for all $S \in \mathcal{S}_0$ if and only if $St(M) \models \Phi$ (i.e. $M \models^m \Phi$) for all $M \in \mathcal{M}$. Thus $Trans(\Phi) \in Th^2_<(T)$ if and only if $\Phi \in Th$. □

Now we can prove Theorem 4.

Proof of Theorem 4 (Sketch). Lemma 7 implies that $Th = \{\Phi \in \mathcal{L}^c \mid Trans(\Phi) \in Th^2_<(T)\}$. Then because $Th^2_<(T)$ is decidable by Lemma 2, it is straightforward to show that Th is decidable. □

5 Example of Application

Let us consider an example of application of the obtained results. Distributed CPS often contain several components (e.g. information processing units) which need an exclusive access to a single shared resource (e.g. an actuator) [2].

Consider a CPS which consists of three components (or processes), two of which (component 1 and component 2) share a certain resource. The resource may be accessed sequentially, but simultaneous access by two components is prohibited. To guarantee absence of simultaneous access to the resource, the components 1 and 2 communicate using shared memory and implement a variant of Peterson's mutual exclusion algorithm. The components 1 and 2 can read shared memory and the current state of another component at any time (possibly simultaneously), but cannot write directly into the shared memory. Instead, the component 3 has exclusive write access to the shared memory and acts as an arbiter which receives write requests from the components 1 and 2. If the component 3 receives one request at a given time moment, it satisfies it immediately. If it receives two requests simultaneously, it chooses one of them (arbitrarily), satisfies it, and declines another one.

Let us model the behavior of the described CPS as a set of trajectories from the time domain T to a set (global) states Q of the form $A_1 \times A_2 \times A_3 \times M \times R$, where A_i, $i = 1, 2, 3$ is the set of (individual) states of the component i, M is the set of states of the shared memory, and R is the set of states of the shared resource. For each $q \in Q$ we denote by $a_i(q)$ the projections of q on the coordinate i, $i = 1, 2, 3$, and by $m(q)$ denote the projection of q on the 4-th coordinate.

Let us assume that $A_1 = Bool$, $A_2 = Bool$, $A_3 = Bool \times Bool \times Bool \times Bool$, and $M = Bool$. The elements of A_i, $i = 1, 2$ indicate whether the component i is interested in obtaining the access the shared resource. The elements of M indicate which of the two components have the priority over another one. The elements of A_3 have the form (w_1, v_1, w_2, v_2), where for $i = 1, 2$, $w_i = true$ means that the component i is asking the component 3 to write the value v_i into the shared memory, and $w_i = false$ means that the component i is not asking the component 3 to change the shared memory. For each $q \in A_3$ we will denote by $w_1(q)$, $v_1(q)$, $w_2(q)$, $v_2(q)$ the projections of $a_3(q)$ on the 1, 2, 3, 4-th coordinate.

For each predicate $P : Q \to Bool$ let us denote

$$Tr^P = \{s : A \to Q \mid A \in \mathfrak{T} \wedge \forall t \in A \; P(s(t))\}$$

and $\Sigma^P = (T, Q, \Sigma^P)$. Obviously, Tr^P is CPR, complete, and Markovian set of trajectories, so Σ^P is a NCMS.

For each predicate $P : Q \to Bool$ denote by $\bar{P}$ the predicate on Q such that $\bar{P}(q) \Leftrightarrow \neg P(q)$ for all $q \in Q$.

We will specify the behavior and the mutual exclusion property of the described CPS in terms of the properties of the following tuple of NCMS:

$$(\Sigma_i)_{i=1,2,\ldots,14} = \\ = (\Sigma^{a_1}, \Sigma^{\bar{a}_1}, \Sigma^{a_2}, \Sigma^{\bar{a}_2}, \Sigma^{m}, \Sigma^{\bar{m}}, \Sigma^{w_1}, \Sigma^{\bar{w}_1}, \Sigma^{w_2}, \Sigma^{\bar{w}_2}, \Sigma^{v_1}, \Sigma^{\bar{v}_1}, \Sigma^{v_2}, \Sigma^{\bar{v}_2}).$$

Let us introduce the following names for the indices of the components of the tuple: $\mathbf{a}_1 = 1, \bar{\mathbf{a}}_1 = 2, \mathbf{a}_2 = 3, \bar{\mathbf{a}}_2 = 4, \mathbf{m} = 5, \bar{\mathbf{m}} = 6, \mathbf{w}_1 = 7, \bar{\mathbf{w}}_1 = 8, \mathbf{w}_2 = 9, \bar{\mathbf{w}}_2 = 10, \mathbf{v}_1 = 11, \bar{\mathbf{v}}_1 = 12, \mathbf{v}_2 = 13, \bar{\mathbf{v}}_2 = 14$.

Let Ψ, Φ_1, Φ_2, Φ_3 be the following formulas in the language $\mathcal{L}$:

$$\Psi := \forall t_1 \bigwedge_{k=1}^{7} \neg(L_{2k-1,1}(t_1) \wedge L_{2k,1}(t_1)) \wedge \neg(R_{2k-1,1}(t_1), R_{2k,1}(t_1))$$

$$\Phi_1(t_1) := (\exists t_2 \exists t_3 \exists t_4 \; t_2 < t_3 \wedge t_3 < t_4 \wedge t_4 < t_1 \wedge \\ (\forall t_5 (t_2 < t_5 \wedge (t_5 < t_1 \vee t_5 = t_1) \to L_{i_1,1}(t_5))) \wedge \\ \wedge L_{w_1,1}(t_3) \wedge L_{v_1,1}(t_3) \wedge R_{v_1,1}(t_3) \wedge \\ (\forall t_5 ((t_3 < t_5 \vee t_3 = t_5) \wedge (t_5 < t_1 \vee t_5 = t_1) \to R_{\bar{w}_1,1}(t_5))) \wedge (R_{\bar{i}_2,1}(t_4) \vee R_{m,1}(t_4)) \;)$$

$$\Phi_2(t_1) := (\exists t_2 \exists t_3 \exists t_4 \; t_2 < t_3 \wedge t_3 < t_4 \wedge t_4 < t_1 \wedge \\ (\forall t_5 (t_2 < t_5 \wedge (t_5 < t_1 \vee t_5 = t_1) \to L_{i_2,1}(t_5))) \wedge \\ \wedge L_{w_2,1}(t_3) \wedge L_{\bar{v}_2,1}(t_3) \wedge R_{\bar{v}_2,1}(t_3) \wedge \\ (\forall t_5 ((t_3 < t_5 \vee t_3 = t_5) \wedge (t_5 < t_1 \vee t_5 = t_1) \to R_{\bar{w}_2,1}(t_5))) \wedge (R_{\bar{i}_1,1}(t_4) \vee R_{\bar{m},1}(t_4)) \;)$$

$$\begin{aligned}\Phi_3 := \forall t_1((R_{\bar{w}_1,1}(t_1) \wedge R_{\bar{w}_2,1}(t_1) \rightarrow (R_{m,1}(t_1) \vee R_{\bar{m},1}(t_1))\wedge \\ \wedge (R_{w_1,1}(t_1) \wedge R_{\bar{w}_2,1}(t_1) \wedge R_{v_1,1}(t_1) \rightarrow R_{m,1}(t_1))\wedge \\ \wedge (R_{w_1,1}(t_1) \wedge R_{\bar{w}_2,1}(t_1) \wedge R_{\bar{v}_1,1}(t_1) \rightarrow R_{\bar{m},1}(t_1))\wedge \\ \wedge (R_{\bar{w}_1,1}(t_1) \wedge R_{w_2,1}(t_1) \wedge R_{v_2,1}(t_1) \rightarrow R_{m,1}(t_1))\wedge \\ \wedge (R_{\bar{w}_1,1}(t_1) \wedge R_{w_2,1}(t_1) \wedge R_{\bar{v}_2,1}(t_1) \rightarrow R_{\bar{m},1}(t_1)))\end{aligned}$$

$$\Phi := \neg\exists t_1(\Psi \wedge \Phi_1(t_1) \wedge \Phi_2(t_1) \wedge \Phi_3)$$

Informally, we can interpret $\Phi(t_i)$, $i = 1, 2$ as a statement that the component i of the CPS described above can access the shared resource at time t_1, and this happens only if there exist three preceding time moments ($t_2 < t_3 < t_4 < t_1$) such that the component i sets its individual state to $True$ at time t_2 (i.e. indicates that it is interested in accessing the shared resource) and sends a request to write the value $true$, if $i = 1$ or $false$, if $i = 2$ to the shared memory to the component 3 at time t_3, and at time t_4, either the component $3 - i$ is not interested in accessing the shared resource, or the value of the shared memory indicates the priority of the component i over the component $3 - i$. This statement can be considered as a high-level expression of the Peterson's algorithm. Also, informally, Φ_3 expresses the algorithm of the component 3 and Ψ is a consistency condition like "a predicate and its negation cannot hold simultaneously".

Then Φ can be interpreted as the statement that there is no time moment t_1 at which the components 1 and 2 of the CPS described above can simultaneously access the shared resource.

One can prove that $\Phi \in Th$ directly using definitions. However, this can also be checked by a decision procedure for Th described in the proof of Theorem 4, which gives a mechanized proof of the mutual exclusion property of an abstract version of Peterson's algorithm in the case of CPS.

6 Conclusions and Future Work

We have proposed a decidable formal theory for describing high-level properties of NCMS. The class of NCMS contains abstract dynamical systems which can represent discrete and continuous evolutions in continuous time and are sufficient for modeling a wide range of real-time information processing and cyber-physical systems. In the future works we plan to extend the introduced theory and apply it to verification of practical cyber-physical systems.

Acknowledgments. We would like to thank Dr. Martin Strecker and Prof. Louis Féraud of Institut de Recherche en Informatique de Toulouse (IRIT), France for the ideas which inspired this work.

References

1. Baheti, R., Gill, H.: Cyber-physical systems. In: Samad, T., Annaswamy, A. (eds.) The Impact of Control Technology, pp. 161–166. IEEE, New York (2011)

2. Lee, E.A., Seshia, S.A.: Introduction to embedded systems: a cyber-physical systems approach. Lulu.com (2013)
3. Shi, J., Wan, J., Yan, H., Suo, H.: A survey of cyber-physical systems. In: 2011 International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6. IEEE (2011)
4. Sunder, S.S., et al. (originator): Cyber-physical systems - a concept map. http://cyberphysicalsystems.org/
5. Barbashin, E.: Introduction to Stability Theory. Nauka (in Russian), Moscow (1967)
6. Emelyanov, S.: Variable Structure Systems. Nauka (in Russian), Moscow (1967)
7. Utkin, V.I.: Sliding Modes in Control and Optimization. Communications and Control Engineering. Springer, Heidelberg (1992)
8. Samoilenko, A., Perestyuk, N.: Impulsive Differential Equations. World Scientific, Singapore and River Edge (1995)
9. Lakshmikantham, V., Banov, D.D., Simeonov, P.S.: Theory of Impulsive Differential Equations. World Scientific Publishing Company, Singapore (1989)
10. Filippov, A.: Differential Equations with Discontinuous Righthand Sides: Control Systems. Mathematics and Its Applications. Springer, New York (1988)
11. Liberzon, D.: Switching in Systems and Control. Systems & Control: Foundations & Applications. Birkhauser Boston Inc., Boston (2003)
12. Witsenhausen, H.: A class of hybrid-state continuous-time dynamic systems. IEEE Trans. Autom. Control **11**, 161–167 (1966)
13. Tavernini, L.: Differential automata and their discrete simulators. Nonlinear Anal. Theor. Methods Appl. **11**, 665–683 (1987)
14. Nerode, A., Kohn, W.: Models for hybrid systems: automata, topologies, controllability, observability. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) HS 1991 and HS 1992. LNCS, vol. 736, pp. 317–356. Springer, Heidelberg (1993)
15. Branicky, M.S.: Studies in hybrid systems: modeling, analysis, and control. Ph.D. thesis, Massachusetts Institute of Technology (1995)
16. Goebel, R., Sanfelice, R.G., Teel, A.: Hybrid dynamical systems. IEEE Control Syst. Mag. **29**, 28–93 (2009)
17. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) HS 1991 and HS 1992. LNCS, vol. 736, pp. 209–229. Springer, Heidelberg (1993)
18. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoret. Comput. Sci. **138**, 3–34 (1995)
19. Henzinger, T.A.: The theory of hybrid automata. In: Proceedings of Eleventh Annual IEEE Symposium Logic in Computer Science LICS '96, pp. 278–292 (1996)
20. Maler, O., Manna, Z., Pnueli, A.: From timed to hybrid systems. In: de Bakker, J.W., Huizing, C., de Roever, W.P., Rozenberg, G. (eds.) REX 1991. LNCS, vol. 600, pp. 447–484. Springer, Heidelberg (1992)
21. Alur, R., Henzinger, T.A.: Modularity for timed and hybrid systems. In: Mazurkiewicz, A., Winkowski, J. (eds.) CONCUR 1997. LNCS, vol. 1243, pp. 74–88. Springer, Heidelberg (1997)
22. Lynch, N., Segala, R., Vaandrager, F.: Hybrid i/o automata. Inf. Comput. **185**, 105–157 (2003)
23. Barringer, H., Kuiper, R., Pnueli, A.: A really abstract concurrent model and its temporal logic. In: POPL, pp. 173–183. ACM Press (1986)

24. Rabinovich, A.: On the decidability of continuous time specification formalisms. J. Logic Comput. **8**, 669–678 (1998)
25. Chaochen, Z., Hansen, M.R., Sestoft, P.: Decidability and undecidability results for duration calculus. In: Enjalbert, P., Wagner, K.W., Finkel, A. (eds.) STACS 1993. LNCS, vol. 665, pp. 58–68. Springer, Heidelberg (1993)
26. Li, X.: Decidability of mean value calculus. J. Comp. Sci. Tech. **14**, 173–180 (1999)
27. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer, Heidelberg (2010)
28. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. Comput. **20**, 309–352 (2010)
29. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. **41**, 143–189 (2008)
30. Rabinovich, A.: On translations of temporal logic of actions into monadic second order logic. Theoret. Comput. Sci. **193**, 197–214 (1998)
31. Bellini, P., Mattolini, R., Nesi, P.: Temporal logics for real-time system specification. ACM Comput. Surv. **32**, 12–42 (2000)
32. Hirshfeld, Y., Rabinovich, A.: Logics for real time: decidability and complexity. Fundam. Inf. **62**, 1–28 (2004)
33. Zhang, J., Johansson, K.H., Lygeros, J., Sastry, S.: Zeno hybrid systems. Int. J. Robust Nonlinear Control **11**, 435–451 (2001)
34. Birkhoff, G.D.: Dynamical Systems. American Mathematical Society, New York (1927)
35. Nemytskii, V., Stepanov, V.: Qualitative Theory of Differential Equations. GITTL (in Russian), Moscow (1949)
36. Gottschalk, W.H., Hedlund, G.A.: Topological Dynamics. American Mathematical Society Colloquium Publications, New York (1955)
37. Roxin, E.: Local definition of generalized control systems. Mich. Math. J. **13**, 91–96 (1966)
38. Hájek, O.: Theory of processes, I. Czech. Math. J. **17**, 159–199 (1967)
39. Sell, G.R.: Differential equations without uniqueness and classical topological dynamics. J. Differ. Equ. **14**, 42–56 (1973)
40. Kloeden, P.E.: General control systems. In: Coppel, W.A. (ed.) Mathematical Control Theory. Lecture Notes in Mathematics, vol. 680, pp. 119–137. Springer, Heidelberg (1978)
41. Kalman, R.E., Falb, P.L., Arbib, M.A.: Topics in Mathematical System Theory. Pure & Applied Mathematics S. McGraw-Hill Education, New York (1969)
42. Mesarovic, M.D., Takahara, Y.: Abstract Systems Theory. Lecture Notes in Control and Information Sciences. Springer, Heidelberg (1989)
43. Matrosov, V.M., Anapolskiy, L., Vasilyev, S.: The Method of Comparison in Mathematical Systems Theory. Nauka (in Russian), Novosibirsk (1980)
44. Willems, J.C.: Paradigms and puzzles in the theory of dynamical systems. IEEE Trans. Autom. Control **36**, 259–294 (1991)
45. Bushaw, D.: Dynamical polysystems and optimisation. J. Differ. Equ. **2**, 351–356 (1963)
46. Movchan, A.: Stability of processes in two metrics. Applied Math. Mech. (in Russian) **24**, 988–1001 (1960)
47. Hájek, O.: Theory of processes, II. Czech. Math. J. **17**, 372–398 (1967)
48. Ivanov, I.: A criterion for existence of global-in-time trajectories of non-deterministic Markovian systems. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2012. CCIS, vol. 347, pp. 111–130. Springer, Heidelberg (2013)

49. Ivanov, I.: On existence of total input-output pairs of abstract time systems. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2013. CCIS, vol. 412, pp. 308–331. Springer, Heidelberg (2013)
50. Ivanov, I.: An abstract block formalism for engineering systems. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G., Zavileysky, M., Kravtsov, H., Kobets, V., Peschanenko, V.S. (eds.) ICTERI. CEUR Workshop Proceedings, vol. 1000, pp. 448–463. CEUR-WS.org (2013)
51. Rabin, M.: Decidability of second-order theories and automata on infinite trees. Trans. AMS **141**, 1–35 (1969)
52. Abraham, U., Ivanov, I., Nikitchenko, M.: Proving behavioral properties of distributed algorithms using their compositional semantics. In: Proceedings of the First International Seminar Specification and Verification of Hybrid Systems, 10–12 October 2011, Taras Shevchenko National University of Kyiv, pp. 9–19 (2011)
53. Luke, J. (originator): Sorgenfrey topology. Encyclopedia of Mathematics. http://www.encyclopediaofmath.org/index.php?title=Sorgenfrey_topology

Pre-automata and Complex Event Processing

Grygoriy Zholtkevych, Boris Novikov, and Volodymyr Dorozhinsky(✉)

Department of Theoretical and Applied Computer Science, V.N. Karazin Kharkiv National University, 4, Svobody Sqr., Kharkiv 61022, Ukraine
g.zholtkevych@karazin.ua, vdorozhinsky@gmail.com

Abstract. In the paper complex event processing systems are considered. The survey of problems and approaches to their solutions associated with this technology is contained in the paper. Based on this survey a mathematical model for complex event processing systems is proposed. Samples of such a model are called a CEP-machines. Authors have demonstrated that this model is closely related with the notion of a pre-automaton introduced in their earlier papers. The proposed model provides rigorous formulations of problems associated with complex event processing. Authors have proved that processing ability of such systems is determined by some compromise between the class of processed event stream and a number of complex events. Further in the paper the problem of CEP-machine synthesis basing on the formal specification has been solved. Authors have established the conditions ensuring possibility of algorithmic realisation for CEP-machines.

Keywords: Pre-automaton · Complex event · Mathematical model · Event processing · Response function · Computable function · Decidable set

1 Introduction

A modern development trend of information and communication technology demonstrates a stable growth of importance to monitor and analyse continuous information streams. These processes require execution "on-the-fly" thereby they should have the ability to respond in a real time mode.

As a rule, a continuous information stream is not continuous in the mathematical sense of this word. A continuous information stream can be considered rather as a sequence of elementary (atomic) events such that each of them carries some information fragment. A finite segment of an event stream is a complex event if this segment has semantic meaning in the context of our interest.

Complex Event Processing (CEP) is a method of tracking and analysing streams of data about things that happen (they are called events) [11, p.3] and about effects caused by them. The goal of complex event processing is to identify

Grygoriy Zholtkevych and Volodymyr Dorozhinsky regret inform about the sudden death of Prof. Boris Novikov.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 100–116, 2014.
DOI: 10.1007/978-3-319-13206-8_5

meaningful events (such as opportunities or threats) and respond to them as quickly as possible [2]. These events may be happening across the various layers of an organization as sales leads, orders or customer service calls. Or, they may be news items [3], text messages, social media posts, stock market feeds, traffic reports, weather reports, or other kinds of data [11].

Important application areas of CEP are the following [7]:

- "Business Activity Monitoring aims at identifying problems and opportunities in early stages by monitoring business process and other critical resources. To this end, it summarizes events into so-called key performance indicators such as, e.g., the average run time of a process";
- "Sensor Networks transmit measured data from the physical world to, e.g., Supervisory Control and Data Acquisition Systems that are used for monitoring of industrial facilities. To minimize measurement and other errors, data of multiple sensors has to be combined frequently. Further, high-level situations (e.g., fire) usually have to be derivative from raw numerical measurements (e.g., temperature, smoke, etc.)";
- "Market Data such as stock or commodity prices can also be considered as events. They have to be analysed in a continuous and timely fashion in order to recognize trends early and to react to them automatically, for example, in algorithmic trading."

The concepts of "timelines" and "flow processing" are critical to explain the need of a new class of systems. Truly, traditional Data Base Management Systems (DBMSs):

- require data to be (persistently) stored and indexed before it could be processed;
- process data only when explicitly asked by the users, i.e., asynchronously with respect to its arrival.

Both aspects contrast with the requirements of modern monitoring systems [4]. To fulfill the requirements two main concepts are used:

- The Data Stream Processing Model [1] represents the information flow processing problem as processing streams of data coming from different sources to generate new data streams as an output, and considers this problem as an extension of traditional data processing, as supported by Data Base Management Systems. Therefore, Data Stream Management Systems (DSMSs) have their roots in DBMSs but introduce essential differences. While traditional DBMSs are designed to work on persistent data, where updates are relatively infrequent, DSMSs are specialized in dealing with transient data that is continuously updated. The same way, while DBMSs run queries just once to return a complete answer, DSMSs execute standing queries, which run continuously and provide updated answers as new data arrives;
- The Complex Event Processing Model [10] considers flowing information items as notifications of events happening in the external world, which have to be filtered and combined to understand what is happening in terms of higher-level events. Therefore, the focus of this model is on detecting occurrences of

particular patterns of (low-level) events that represent the higher-level events whose occurrences has to be notified to the interested parties [4].

The situations (specified as complex events) that need to be detected in the applications mentioned above and the information associated with these situations are distributed over several events. Thus CEP can only derive such situations from a number of correlated (simple) events. To this end many different languages and formalisms for querying events, the so-called Event Query Languages (EQLs), have been developed in the past. Today the following EQL styles are used [7]:

- Composition Operators build complex event queries from simple event queries using composition operators. Complex event queries are expressed by composing single events using different composition operators. Typical operators are conjunction of events (all events must happen, possibly at different times), sequence (all events happen in the specified order), and negation within a sequence (an event does not happen in the time between two other events);
- Data Stream Query Languages are based on the database query language SQL and the following general idea: Data Streams carry events represented as tuples. Each data stream corresponds to exactly one event type. The streams are converted into relations which essentially contain (parts of) the tuples received so far. On these relations a (almost) regular SQL query is evaluated. The result (another relation) is then converted back into a data stream;
- Production Rules are very flexible and well integrated with existing programming languages. However, it entails working on a low abstraction level that is – since it is primarily state and not event oriented – somewhat different from other EQLs. Especially aggregation and negation are therefore hard to express. Production rules are considered to be less efficient then data stream query languages; this is however tied to the flexibility they add in terms of combining queries (in rule conditions) and reactions (in rule actions);
- Timed State Machines are usually used to model the behavior of a stateful system that reacts to events. The system is modelled as a directed graph. The nodes of the graph represent the possible states of the system. Directed edges are labelled with events and temporal conditions on them. The edges specify the transitions between states that occur in reaction to in-coming events. State machines are founded formally on deterministic or non-deterministic finite automata. Since states in a state machine are reached by particular sequence of multiple events occurring over time, they implicitly define complex events;
- Logic Languages express event queries in logic-style formulas. Logic languages offer a natural and convenient way to specify event queries. The main advantage of logic languages is their strong formal foundation, an issue which is neglected by many languages of other styles. Thanks to the separation of different dimensions of event processing, logic languages are highly expressive, extensible and easy to learn and use.

CEP depends on a number of techniques, [8] including: event-pattern detection; event abstraction; event filtering; event aggregation and transformation;

modelling event hierarchies; detecting relationships (such as causality, membership or timing) between events; abstracting event-driven processes.

The presented survey shows that there are a number of technical solutions and software tools for implementing the concept of CEP. But some principal problems for theoretical substantiation of the concept remain open. Among them, e.g. the existence problem for unhandled event streams, which has an important value for system design. The problem of CEP-system synthesis in accordance with its input/reaction specification is the following example of such open problems.

The main objective of this paper is to construct a mathematical model for complex event processing systems to create a theoretical background for formal analysis such systems.

The mathematical model proposed by authors is closely related with the notion of a pre-automaton, which was introduced in [6] and studied further in [12,14–16].

This paper consists of this introduction, seven sections, and conclusion. Section 2 introduces basic notion and notations. Section 3 contains the comparison of architectures for simple event processing and complex event processing.

Section 4 is devoted to constructing a mathematical model of a system for complex event processing. This model is called a CEP-machine. Elementary properties of a CEP-machine are studied in Sect. 5. The CEP-machine synthesis problem is considered in Sect. 6.

Finally, in Sect. 7 the problem of algorithmic realising for a CEP-machine is studied.

2 Basic Notions and Notations

Let X and Y be sets and f be a partial mapping from X into Y then the notation $f : X \dashrightarrow Y$ is used to specify that f is a partial mapping in contrast to the notation $X \to Y$, which is used to specify everywhere defined mappings.

Let $f : X \dashrightarrow Y$ be a partial mapping from a set X into a set Y, x be some element of X then the notation $f(x) \downarrow$ is used to indicate that x belongs to the domain of the mapping f. Moreover, if x belongs to the domain of the mapping f and it is known that $f(x) = y$ then the denotation $f(x) \downarrow y$ is used to express this fact. Similarly, the notation $f(x) \uparrow$ is used to indicate that x does not belong to the domain of the mapping f. As usually, we use the denotation $D(f)$ for the domain of f.

Let Σ be a finite alphabet. As usually, we use the notation Σ^* to refer to the free monoid generated by Σ. The unit of this monoid is denoted by ε. The semigroup of all non-empty words over an alphabet Σ is denoted by Σ^+. Hence, $\Sigma^* = \Sigma^+ \bigcup \{\varepsilon\}$.

For any $u \in \Sigma^+$ we denote by $|u|$ the length of u, i.e. a number of symbols containing in u, and set that $|\varepsilon| = 0$.

Any set L consisting of words over alphabet Σ is called prefix-free if assertions $uv \in L$ and $u \in L$ for $u, v \in \Sigma^*$ imply the equality $v = \varepsilon$. For any $L \subset \Sigma^+$ we denote by $C(L)$ the following set

$$C(L) = \{w \in L \mid w = uv \text{ and } u \in L \text{ imply } v = \varepsilon\}.$$

One can easily see that $L \subset \Sigma^+$ is prefix-free if and only if $C(L) = L$.

In addition to words over some alphabet we consider infinite sequences of alphabet symbols. Therefore we use the notation Σ^ω for the set of all one-way infinite sequences of symbols belonging to the alphabet Σ. For $\pi \in \Sigma^\omega$ and $n \in \mathbb{N}$ we use the notation $\pi_{[1..n]}$ to refer to the word that has a length n and coincides with the beginning of the sequence π. We denote also by $\pi_{(n..)}$ the sequence belonging to Σ^ω that is defined by the equality $\pi = \pi_{[1..n]}\pi_{(n..)}$.

3 Event Processing Versus Complex Event Processing

In this section we are going to consider specificity of CEP in contrast to simple event processing.

Definition 1 (Atomic and Complex Events). *We shall say that an event is atomic if it can not be represented as a complex of sub-events that are essential for the domain of our interest.*

In contrast, we shall say that an event is complex if it can be represented as a complex of sub-events that are essential for the domain of our interest.

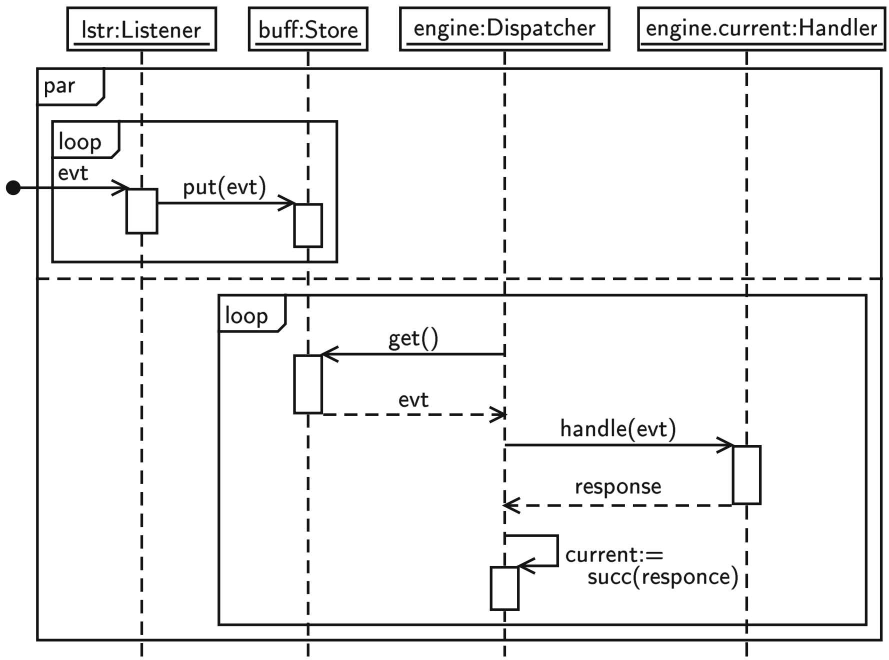


Fig. 1. Simple event processing

Example 1. Language Java distinguishes low-level events and semantic events. A keyboard, a mouse and other input devices produce the low-level events.

In contrast to low-level events semantic events are produced as a result of some sequence of low-level events, e.g. SelectItemEvent is a result of the sequence: GetFocusEvent, DownButtonEvent, and UpButtonEvent.

Definition 2 (Simple Event Processing). *Simple event processing is based on the suggestion that each event is directly related to specific, measurable changes of condition and can be processed standalone.*

In the case of simple event processing each notable happened event initiates immediately the corresponding action.

Simple event processing is commonly used to drive the real-time work-flow thereby reducing lag time and cost [13].

As shown in Fig. 1 event handler processes atomic events as soon as notifications about them being received.

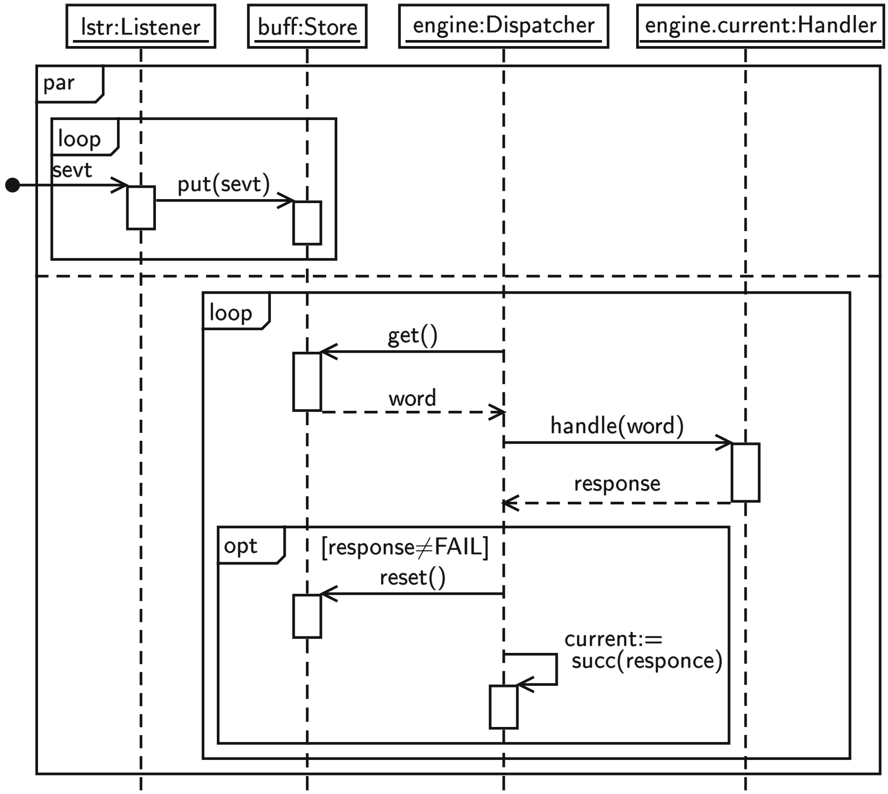


Fig. 2. Complex event processing

Definition 3 (Complex event processing). *Complex event processing is based on the suggestion that each atomic event carries too small volume of information to determine the adequate reaction.*

Complex event processing evaluates a confluence of events, determines an event pattern and then takes the corresponding action.

Complex events may have different types and may occur over a long period of time. The event dependencies may be different also and be causal, temporal, or spatial. CEP requires to use sophisticated event interpreters, event pattern definitions and recognition mechanisms [13].

As shown in Fig. 2 complex event handler is searching through the event buffer to recognise a complex event. And if the complex event was recognised then handler processes it and clears the event buffer. Thus the main difference between Simple Event Processing and Complex Event Processing is that in the first case a processing system reacts on each atomic event immediately while in the second case a processing system is trying to detect and process some meaningful complex event by analysing the buffer of atomic events. Facts mentioned above and results obtained in [12,14–16] show that concept of the Pre-Automata can be widely used in mathematical modelling of Complex Event Processing Systems.

4 Formal Model of Complex Event Processing System

In this section some mathematical model of an abstract machine for complex event processing is built. To do this we choose and fix two finite alphabets Σ and A to describe atomic events and system responses respectively.

Definition 4 (a handler). *A partially defined map* $h : \Sigma^+ \dashrightarrow \mathrm{A}$ *we shall call a handler if its domain* $D(h)$ *is a prefix-free set.*

To explain the necessity of the equation $C(D(h)) = D(h)$ note that the requirement to accumulate a sequence of atomic events until it is recognised as a processed sequence, is described by this equation.

Definition 5 (a pattern). *Let* $h : \Sigma^+ \dashrightarrow \mathrm{A}$ *be a handler and* $a \in \mathrm{A}$ *be some response then the prefix-free set* $P_h(a) = \{w \in D(h) \mid h(w) = a\}$ *is called an* h, a*-pattern.*

A definition of an abstract machine for complex event processing should answer the informal description given by Fig. 2.

Definition 6 (CEP-machine). *An abstract machine for complex event processing (below a CEP-machine) is a quadruple* $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ *where* Σ *and* A *are finite alphabets of events and responses respectively,* H *is a finite set of handlers, and* $\delta : \mathrm{A} \to H$ *is a total map, which is called a transition map.*

One can easily see that each handler h in the model corresponds to the method handle() of the class *Handler* and the map δ corresponds to the method succ() of the class Dispatcher (see Fig. 2).

To define a behaviour of a CEP-machine let us consider the set Σ^ω containing all infinite sequences of elements belonging Σ. Such sequences will be called event streams.

Let us define the partial function $T_h : \Sigma^\omega \dashrightarrow \mathbb{N}$ for any handler $h \in H$ by the following conditions

$$\begin{array}{rl} T_h(\pi) \downarrow t & \text{if } \pi_{[1..t]} \in D(h)\,; \\ T_h(\pi) \uparrow & \text{if } (\forall\, t \in \mathbb{N})\ h(\pi_{[1..t]}) \uparrow\ . \end{array} \tag{1}$$

Informally, $T_h(\pi)$ is the first response time of the handler h under processing the event stream π.

Definition 7 (an evolutionary operator of CEP-machine). *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine then the partial map $S : H \times \Sigma^\omega \dashrightarrow H \times \Sigma^\omega$ defined by the conditions*

$$\begin{array}{rl} S\langle h, \pi\rangle \downarrow \langle \delta(h(\pi_{[1..t]})), \pi_{(t..)}\rangle & \textit{if } T_h(\pi) \downarrow t \\ S\langle h, \pi\rangle \uparrow & \textit{if } T_h(\pi) \uparrow \end{array}$$

will be called an evolutionary operator of the CEP-machine.

Now to specify a correct behaviour of a CEP-machine we define its valid scenarios called work-flows.

Definition 8 (a work-flow). *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine and $\langle\langle h^{(t)}, \pi^{(t)}\rangle \mid\ t = 0, 1, \ldots\rangle$ be a sequence over $H \times \Sigma^\omega$ then it is called a work-flow if the following equality holds*

$$\langle h^{(t+1)}, \pi^{(t+1)}\rangle = S\langle h^{(t)}, \pi^{(t)}\rangle \ \textit{for all}\ t \geq 0\,.$$

5 Elementary Properties of CEP-machines

Simple, but very important properties of CEP-machines are a consequence of the existence of some natural topology on the set Σ^ω.

More precisely, for a finite alphabet Σ the family $\mathcal{B} = \{u \cdot \Sigma^\omega \mid u \in \Sigma^+\}$ of subsets over Σ^ω holds characteristic properties of a topological base. Therefore, we consider Σ^ω as a topological space and the corresponding topology is called Tychonoff topology on sequence space. It is well-known that this topological space is a bicompact.

Proposition 1. *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine then for any $h \in H$ the function $T_h : \Sigma^\omega \dashrightarrow \mathbb{N}$ defined by (1) is continuous function on the set $\{\pi \in \Sigma^\omega \mid T_h(\pi) \downarrow\}$ under the assumption that the topology on $\mathbb{N}$ is discrete.*

Proof. One can easily see that for any $t \in \mathbb{N}$ the equality

$$T_h^{-1}(t) = \bigcup_{u \in D(h) : |u| = t} u \cdot \Sigma^\omega$$

is true. But the right side of this equality is a union of sets belonging to $\mathcal{B}$, hence $T^{-1}(t)$ is an open set. ◻

Corollary 1. *The function $T_h : \Sigma^\omega \dashrightarrow \mathbb{N}$ is a piecewise constant function.*

Corollary 2. *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine, S be its evolutionary operator, and $D(S)$ be domain of S then*

$$D(S) = \bigcup_{h \in H} D_h(S)\,,$$

where each $D_h(S) \subset \Sigma^\omega$ is an open set in Tychonoff topology on Σ^ω.

Theorem 1. *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine, S be its evolutionary operator, and $h \in H$ be some handler then $S\langle h, \pi\rangle \downarrow$ for all $\pi \in \Sigma^\omega$ if and only if $D(h)$ is finite and $\Sigma^\omega = \bigcup_{u \in D(h)} u \cdot \Sigma^\omega$.*

Proof. Indeed, suppose that $S\langle h, \pi\rangle \downarrow$ for any event stream π then the statement $T_h(\pi) \downarrow$ is true for all $\pi \in \Sigma^\omega$. It means that

$$\Sigma^\omega = \bigcup_{t=1}^{\infty} T^{-1}(t) = \bigcup_{t=1}^{\infty} \left(\bigcup_{u \in D(h) : |u| = t} u \cdot \Sigma^\omega \right) .$$

Further, taking into account Proposition 1 and Weierstrass Boundedness Theorem one can conclude that there exists $M \in \mathbb{N}$ such that $T_h(\pi) \leq M$ for all $\pi \in \Sigma^\omega$. Hence, we obtain that

$$\Sigma^\omega = \bigcup_{t=1}^{M} \left(\bigcup_{u \in D(h) : |u| = t} u \cdot \Sigma^\omega \right) .$$

Thus, the set $D_0(h) = \{u \in D(h) \mid 1 \leq |u| \leq M\}$ is finite.

Let $D_0(h) = \{u_1, \ldots, u_m\}$ then

1. $u_i \in D(h)$ for all $i = 1, \ldots, m$ and
2. $\Sigma^\omega = \bigcup_{i=1}^{m} u_i \cdot \Sigma^\omega$.

We claim that $u_i \cdot \Sigma^\omega \bigcap u_j \cdot \Sigma^\omega = \emptyset$ for $1 \leq i \neq j \leq m$. Indeed, suppose that there exists some $\pi \in u_i \cdot \Sigma^\omega \bigcap u_j \cdot \Sigma^\omega$ then there are three possibilities:

- $|u_i| = |u_j|$ and in this case we have the equality $u_i = u_j$ that contradicts the assumption $i \neq j$;
- $|u_i| < |u_j|$ and u_i is a prefix of u_j that contradicts the equality $C(D(h)) = D(h)$;
- $|u_i| > |u_j|$ that is impossible too (reasoning is similar to the previous item).

Further, let u be an arbitrary element in $D(h)$ then taking into account properties 1 and 2 of $D_0(h)$ we can conclude the following statement: for each $\pi \in u \cdot \Sigma^\omega$ there exists the unique $1 \leq i(\pi) \leq m$ such that $\pi \in u_{i(\pi)} \cdot \Sigma^\omega$. As above the assumption $|u| \neq |u_{i(\pi)}|$ contradicts the equality $C(D(h)) = D(h)$ and the equality $|u| = |u_{i(\pi)}|$ means $u = u_{i(\pi)}$. Hence, $D(h) = D_0(h)$ and the direct statement of the theorem is proved.

The converse statement of the theorem is evident. □

Remark 1. Theorem 1 is not complicated but it grounds the following important alternative: either a handler is able to provide a response to an infinite number of complex events and, in such a case, it can not provide processing of any event stream, or the handler is able to provide processing of any event stream and, in such a case, its behaviour consists in responding to a finite number of complex events.

6 CEP-machines and Pre-automata

As it noted above, the concept of a pre-automaton is closely related with the concept of a CEP-machine. This section is devoted to explaining of the mentioned relation.

Definition 9. (see [6]). *A triple (X, Σ, μ), where X is a set, Σ is a finite alphabet, and $\mu : X \times \Sigma^* \dashrightarrow X$ is a partial mapping, is called a pre-automaton if the following conditions hold*

1. *the equality $\mu(x, \varepsilon) \downarrow x$ holds for all $x \in X$;*
2. *if the assertions $\mu(x, u) \downarrow$ and $\mu(\mu(x, u), v) \downarrow$ are true for some $x \in X$ and $u, v \in \Sigma^*$ then the assertion $\mu(x, uv) \downarrow \mu(\mu(x, u), v)$ is true too;*
3. *if the assertions $\mu(x, u) \downarrow$ and $\mu(x, uv) \downarrow$ are true for some $x \in X$ and $u, v \in \Sigma^*$ then the assertion $\mu(\mu(x, u), v) \downarrow \mu(x, uv)$ is true too.*

In this context μ is called a transition function.

Now we describe a manner to associate a pre-automaton with a CEP-machine. The origin point for our construction is a CEP-machine $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$. We will find the target pre-automaton as the triple $\widehat{\mathcal{M}} = (H, \Sigma, \mu)$ such that the partial mapping $\mu : H \times \Sigma^* \dashrightarrow H$ satisfies the following condition

$$\begin{array}{l} \text{for any } h \in H,\ w \in D(h), \text{ and } \pi \in \Sigma^\omega \\ \quad \text{the conjunction of } \mu(h, w) \downarrow \text{ and } S\langle h, w \cdot \pi\rangle \downarrow \langle \mu(h, w), \pi\rangle \text{ is true} \end{array} \tag{2}$$

where S is the evolutionary operator of the CEP-machine $\mathcal{M}$.

Theorem 2. *Let $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ be a CEP-machine and $\mu : H \times \Sigma^* \dashrightarrow H$ be defined in the following manner*

1. *$\mu(h, \varepsilon) = h$ for any $h \in H$;*
2. *for any $h \in H$, $u \in D(h)$, and $v \in \Sigma^*$ the truth of the assertion $\mu(\delta(h(u)), v) \downarrow$ implies that $\mu(h, uv) \downarrow$ is true and in this case $\mu(h, uv) = \mu(\delta(h(u)), v)$;*
3. *$\mu(h, w) \uparrow$ for all other cases,*

then the triple $\widehat{\mathcal{M}} = (H, \Sigma, \mu)$ is a pre-automaton and the mapping μ satisfies (2).

To prove the Theorem 2 the following lemma is needed.

Lemma 1. *Let* μ *be as in Theorem 2,* $h \in H$*, and* $w \in \Sigma^+$ *be a word such that* $\mu(h, w)\downarrow$ *then there exists the unique alternating sequence* $h_0 \in H$, $u_0 \in \Sigma^+$, $h_1 \in H$, $u_1 \in \Sigma^+$, ..., $u_{n-1} \in \Sigma^+$, $h_n \in H$ *such that*

1. $w = u_0 \dots u_{n-1}$;
2. $u_i \in D(h_i)$ *for all* $i = 0, \dots, n-1$;
3. $h_0 = h$ *and* $h_{i+1} = \mu(h_i, u_i)$ *for all* $i = 0, \dots, n-1$.

Proof. To prove the lemma let us consider the following algorithm.

1: let us assign $h_0 \leftarrow h$, $w_0 \leftarrow w$, and $i \leftarrow 0$;
2: if $w_i \in D(h_i)$ then assign $h_{i+1} \leftarrow \mu(h_i, w_i)$, $u_i \leftarrow w_i$, and halt;
3: choose $u_i \in D(h_i)$ and $w_{i+1} \in \Sigma^+$ so as to satisfy the equality $w_i = u_i w_{i+1}$;
4: let assign $h_{i+1} \leftarrow \mu(h_i, u_i)$ and $i \leftarrow i + 1$;
5: go to item 2.

Taking into account that $\mu(h, w)\downarrow$ and the definition of μ one can obtain that for any i either $w_i \in D(h_i)$ and algorithm terminates or w_i can be represented in accordance to item 3. In the last case the definition of μ guaranties that $\mu(h_i, w_i)\downarrow$. Moreover, the inequality $|w_i| < |w_{i-1}|$ holds.

Thus, an algorithm execution is terminated because each its step decreases length of the current word w_i. After such a termination we evidently obtained the required alternating sequence. □

Corollary 3. *Any word* $w \in \Sigma^+$ *such that* $\mu(h, w)\downarrow$ *can be uniquely represented in the form* $w = uv$ *where* u *is a word satisfying the condition* $\mu(h, u)\downarrow$ *and* $v \in D(\mu(h, u))$.

Proof (of Theorem 2). Primarily, let us check that μ satisfies conditions 1 – 3 of Definition 9. The validity of condition 1 is evident.

To check condition 2 of the theorem assume that $\mu(h, u)\downarrow$ and $\mu(\mu(h, u), v)\downarrow$. Using Lemma 1 one can construct the alternating sequence

$$h_0 = h, u_0, \dots, u_{m-1}, h_m = \mu(h, u), u_m, \dots, u_{m+n-1}, h_{m+n} = \mu(\mu(h, u), v)$$

such that the sequence $h_0 = h, u_0, \dots, u_{m-1}, h_m = \mu(h, u)$ is the sequence for $\mu(h, u)$ and $h_m = \mu(h, u), u_m, \dots, u_{m+n-1}, h_{m+n} = \mu(\mu(h, u), v)$ is the sequence for $\mu(\mu(h, u), v)$. Thus $u = u_0 \dots u_{m-1}$, $v = u_m \dots u_{m+n-1}$, and $uv = u_0 \dots u_{m-1} u_m \dots u_{m+n-1}$. It is evident that conditions 2 and 3 of Lemma 1 hold therefore the constructing sequence is the sequence for $\mu(uv, h)$. This means that $\mu(h, uv) \downarrow \mu(\mu(h, u), v)$.

To check condition 3 of the theorem is similarly to check condition 2.

The validity of (2) is evident. □

Definition 10 (the dual pre-automaton for a CEP-machine). *Let* $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta)$ *be a CEP-machine and* $\widehat{\mathcal{M}} = (H, \Sigma, \mu)$ *be the pre-automaton determined by Theorem 2 then* $\widehat{\mathcal{M}}$ *will be called the dual pre-automaton for the CEP-machine* $\mathcal{M}$.

We can carry out the converse construction too and associate with any pre-automaton $\mathcal{P}$ a CEP-machine $\mathcal{M}$ so as to satisfy the equality $\mathcal{P} = \widehat{\mathcal{M}}$.

Proposition 2. *Suppose that a pre-machine $\mathcal{P} = (X, \Sigma, \mu)$ is given and let us define the quadruple $\widehat{\mathcal{P}} = (\Sigma, X, H, \delta)$ such that*

$$H = \{h_x : \Sigma^+ \dashrightarrow X \mid x \in X\} \text{ where}$$

$$D(h_x) = \{w \in \Sigma^+ \mid \mu(x, w) \downarrow \text{ and if } w = uv \wedge \mu(x, u) \downarrow \text{ then } v = \varepsilon\},$$
$$h_x(w) = \mu(x, w) \text{ for } w \in D(h_x);$$

$$\delta(x) = h_x .$$

Then $\widehat{\mathcal{P}}$ is a CEP-machine and $\widehat{\widehat{\mathcal{P}}}$ is isomorphic to $\mathcal{P}$.

Proof. Let $\widehat{\mu} : H \times \Sigma^* \dashrightarrow H$ be the transition function for the pre-automaton $\widehat{\widehat{\mathcal{P}}}$. Then for $w \in D(h_x)$ we have $\widehat{\mu}(h_x, w) = \delta(h_x(w)) = h_{\mu(x,w)}$. Extending the mapping $\widehat{\mu}(x, w)$ for $w \in \Sigma^*$ in according with the conditions of Theorem 2 we obtain $\widehat{\mu}(h_x, w) \downarrow$ if and only if $\mu(x, w) \downarrow$ and in this case $\widehat{\mu}(h_x, w) = h_{\mu(x,w)}$. Hence, the mapping such that $x \mapsto h_x$ is an isomorphism of pre-automata [6]. □

7 Synthesis of CEP-machines with Specified Behaviour

The real engineering practice requires methods to provide improving of development processes, in particular, for synthesis of CEP-systems. The key notion for this synthesis problem is the notion of "a CEP-machine behaviour". In this section we propose a generalization for the notion "a behaviour", which was defined for automata, and solve the corresponding synthesis problem.

In this context the first problem is to set a method to specify a CEP-machine behaviour. We suggest that to specify the behaviour of the CEP-machine and to specify a mapping associating an input event stream with a sequence of responses of the CEP-machine are the solutions of the same problem. These considerations lead us to the following definitions.

Definition 11 (an initial CEP-machine). *A quintuple $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta, h_*)$, where $h_* \in H$, is called an initial CEP-machine if the quadruple $(\Sigma, \mathrm{A}, H, \delta)$ is a CEP-machine.*

In other words, an initial CEP-machine is a CEP-machine with the marked handler, which is called an initial handler. This handler is chosen as the active handler under initialization the CEP-machine.

Definition 12 (the response function of a CEP-machine). *For an arbitrary initial CEP-machine $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta, h_*)$ let us define the partial mapping $\beta : \Sigma^+ \dashrightarrow \mathrm{A}$ in the following manner*

1. *the domain of β is defined by the equality $D(\beta) = \{w \in \Sigma^+ \mid \mu(h_*, w) \downarrow\}$ where μ is the transition function of the pre-automaton $\widehat{\mathcal{M}}$;*

2. *for* $w \in D(\beta)$ *using Corollary 3 represent* $w = uv$ *such that* $\mu(h_*, u) \downarrow$ *and* $v \in D(\mu(h_*, u))$ *then define* $\beta(w) = \mu(h_*, u)(v)$.

The constructed mapping β *is called the response function of the CEP-machine* $\mathcal{M}$.

In the following proposition the main property of response functions is established.

Proposition 3. *Let* $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta, h_*)$ *be an initial CEP-machine and* $\beta : \Sigma^+ \dashrightarrow \mathrm{A}$ *be its response function then the following condition is true for any* $u, v \in D(\beta)$ *such that* $\beta(u) = \beta(v)$ *and any* $w \in \Sigma^*$

$$uw \in D(\beta) \text{ implies } vw \in D(\beta) \text{ and } \beta(uw) = \beta(vw)\,. \tag{3}$$

Proof. Let $u, v \in D(\beta)$ and $\beta(u) = \beta(v)$. Taking into account the last equality and representation $\mu(h_*, u) = \delta(\beta(u))$ one can conclude that $\mu(h_*, u) = \mu(h_*, v)$. If $uw \in D(\beta)$ then $\mu(h_*, uw) \downarrow$ and using condition 3 of Definition 9 we obtain that $\mu(\mu(h_*, u), w) \downarrow$ and, hence,

$$\mu(h_*, uw) = \mu(\mu(h_*, u), w) = \mu(\delta(\beta(u)), w) = \mu(\delta(\beta(v)), w) = \mu(\mu(h_*, v), w)\,.$$

Therefore, $\mu(\mu(h_*, v), w) \downarrow$ and using condition 2 of Definition 9 we obtain that $\mu(h_*, vw) \downarrow$ i.e. $vw \in D(\beta)$.

Further, $\beta(u) = \beta(v)$ implies $\mu(h_*, u) = \mu(h_*, v)$ and we can use the decomposition from Corollary 3 for $h = \mu(h_*, u) = \mu(h_*, v)$ and w. Let $w = w'w''$ be the corresponding decomposition then $\mu(h, w') \downarrow$ and $w'' \in D(\mu(h, w'))$. Definition 9 and the equality $\beta(u) = \beta(v)$ ensure that the statements $\mu(h_*, uw') \downarrow$ and $\mu(h_*, vw') \downarrow$ are true both and the equality $\mu(h_*, uw') = \mu(h_*, vw')$ hold. Therefore, we have

$$\begin{aligned}\beta(uw) = \mu(h_*, uw')(w'') = {} & \mu(\mu(h_*, u), w')(w'') = \mu(h, w')(w'') = \\ & \mu(\mu(h_*, v), w')(w'') = \mu(h_*, vw')(w'') = \beta(vw)\,.\end{aligned}$$

Thus, proof is complete. □

The converse statement to the previous proposition is true too.

Theorem 3. (about synthesis of a CEP-machine). *Let* Σ *and* A *be finite alphabets and* $\beta : \Sigma^+ \dashrightarrow \mathrm{A}$ *be a partial mapping satisfying* (3) *then there exists an initial CEP-machine* $\mathcal{M} = (\Sigma, \mathrm{A}, H, \delta, h_* \in H)$ *whose response function coincides with* β.

Proof. Let us prove the theorem in two stages. Primarily construct a CEP-machine $\mathcal{M}_\beta$ using the mapping β and then prove that its response function coincides with β.

To realise the first stage of the proof let us consider on the set $D(\beta) \subset \Sigma^*$ the binary relation $\equiv_\beta$ defined by the next way: $u \equiv_\beta v$ means that $\beta(u) = \beta(v)$ is true. It is evident that this relation is an equivalence.

Now let us consider $H = \{h_*\} \bigcup \{h_{[u]_\beta} \mid u \in D(\beta)\}$ where $[u]_\beta$ is the equivalence class of u with respect to equivalence $\equiv_\beta$. Taking into account that $\beta|[u]_\beta$ is a constant mapping for each $u \in D(\beta)$ one can conclude that $|H| \leq |\mathrm{A}| + 1$. Therefore H is a finite set.

Let us define $h_*(w) \downarrow$ if $w \in C(D(\beta))$ and in this case $h_*(w) = \beta(w)$. The definition of $C(D(\beta))$ ensure that h_* is a handler.

Further, for $u \in D(\beta)$ define that $h_{[u]_\beta}(w) \downarrow$ if $uw \in C(D(\beta))$. Condition (3) ensures that this definition does not depend on a choice of $u' \in [u]_\beta$. Moreover, in the considered case the equality $\beta(uw) = \beta(u'w)$ holds. Therefore, the formula $h_{[u]_\beta}(w) = \beta(uw)$ defines $h_{[u]_\beta}$ correctly. To check that $h_{[u]_\beta}$ is a handler let assume that $w'w'' \in D(h_{[u]_\beta})$ and $w' \in D(h_{[u]_\beta})$ then $uw'w'' \in C(D(\beta))$ and $uw' \in C(D(\beta))$ are true both. But the last conjunction means that $w'' = \varepsilon$ by the definition of $C(D(\beta))$. Hence, $h_{[u]_\beta}$ is a handler.

To define $\delta : \mathrm{A} \to H$ note that for any $a \in \beta(D(\beta))$ the equality $\beta(u) = a$ uniquely determines $[u]_\beta$, hence we can define δ on $\beta(D(\beta))$ by the next way $\delta(a) = [u]_\beta$ where $\beta(u) = a$. On the set $\mathrm{A} \setminus \beta(D(\beta))$ we can define the mapping δ in arbitrary way, e.g. setting $\delta|(\mathrm{A} \setminus \beta(D(\beta))) = h_*$.

Thus, we have constructed the CEP-machine $\mathcal{M}_\beta = (\Sigma, \mathrm{A}, H, \delta)$.

To complete proof it is need to check that the response function of the CEP-machine $\mathcal{M}_\beta$ coincides with β. Denote by $\widehat{\beta}$ the response function of the CEP-machine $\mathcal{M}_\beta$ and by μ the transition function of the pre-automaton $\widehat{\mathcal{M}_\beta}$.

By definition $D(\widehat{\beta})$ coincides with the set $\{w \in \Sigma^+ \mid \mu(h_*, w) \downarrow\}$. For such a word w we can apply Lemma 1 and construct the alternating sequence $h_0 = h_*$, u_0, $h_1 = h_{[u_0]_\beta}$, $\ldots$, $h_{n-1} = h_{[u_0 \ldots u_{n-2}]_\beta}$, u_{n-1}, $h_n = h_{[u_{n-1}]_\beta}$ such that $u_0 \ldots u_i \in D(h_i)$ for $i = 0, \ldots, n-1$ and $w = u_0 \ldots u_{n-1}$. Hence, $\widehat{\beta}(w) = \beta(u_0 \ldots u_{n-2} u_{n-1}) = \beta(w)$. □

8 Complex Event Processing and Computability

Above we were considering the general properties of CEP-machines and the theory was being constructed similar to finite automata theory. However, problems associated with CEP-machines are not automatically solvable in contrast to problems associated with finite automata. Therefore computability for CEP-machines is a special problem requiring its studying.

For use the described above approach to solve problems of designing software systems, we restrict the class of handlers, namely, we consider computable handlers only. Such a restriction is a key to provide processing of event streams using a computational system. In the most general form the schema of complex event processing is represented in Fig. 2. We should stress that there is a potential problem in this schema. It is connected with applying the method handle(). This method sequentially applies the corresponding handler to the current buffer contents. But a computable function does not identify data which does not belong to the domain of this function in general case. Therefore an attempt to execute this method can lead to its infinite running. In this section we show that the indicated anomaly can be eliminated.

Theorem 4. (about Decidability of the Halting Problem for Handlers). *Let* Σ *and* A *be finite alphabets,* $h : \Sigma^+ \dashrightarrow \mathrm{A}$ *be a computable handler, and* $\pi \in \Sigma^\omega$ *be an event stream then the predicate* $M(\pi) \cong \exists n\ h(\pi_{[1..n]}) \downarrow$ *is decidable relative to* $g(n) = \pi_{[1..n]}$.

Proof. As it is established in computation theory the domain of a computable function is a semidecidable set. Hence, the predicate $w \in D(h)$ is semidecidable. Such a predicate is represented as $\exists t\ R(w,t)$ for some decidable predicate R (see [5, p.114,Theorem6.4]). As R one can be chosen, for example, the predicate "a program to compute h executes at most t commands and is halted when the input is w".

Let us $\gamma : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ be the mapping defined by the formulae

$$\begin{aligned} \gamma_1(n) &= \mu\, k\ \left(n < \frac{k(k+1)}{2}\right) - 1 \\ \gamma_2(n) &= n - \gamma_1(n) \end{aligned},$$

where $\mu\, k\ (M(k,n))$ means the minimal value of k that satisfies the condition $M(k,n)$. It is evident that $\gamma = (\gamma_1, \gamma_2)$ is a computable bijection.

Now let us consider the following algorithm

1: let assign $n \leftarrow 1$;
2: let assign $m, t \leftarrow \gamma(n)$;
3: if $R(g(m), t)$ then return $g(m)$ and halt;
4: let assign $n \leftarrow n + 1$;
5: go to item 2.

Taking into account that the domain of h is prefix-free one can easily check the following sentence: either there exists m such that $\pi_{[1..m]} \in D(h)$ and the algorithm finds it or the algorithm is executing infinitely. To complete proof it is sufficient to transform this informal algorithm into the corresponding Turing machine. □

Corollary 4. *If all handlers of a CEP-machine are computable then this machine can be algorithmic realised.*

Proof. Indeed, we can find $\pi_{[1..m]} \in D(h)$ using the algorithm of Theorem 4 if such $\pi_{[1..m]}$ exists and then calculate $h(\pi_{[1..m]})$. □

9 Conclusion

In the paper a study of complex event processing systems has been conducted. This study has based on known examples of software solutions for such systems and results of a generalisation of these examples. Therefore some survey of complex event processing systems has been taken as a basis of the research.

Abstracting of invariants for these examples has led to the formulation of a mathematical model, which has been called a CEP-machine. This model has been described from two points of view: structural and behavioural. To specify the behavioural part of the model a number of formal notions has been introduced.

Basing on these formal objects the elementary properties of a CEP-machine have been studied. Theorem 1 expresses the principal result of such a study. It consists in that either an event handler is able to provide a response to an infinite number of complex events and, in such a case, it can not provide processing of any event stream, or the handler is able to provide processing of any event stream and, in such a case, its behaviour consists in responding to a finite number of complex events.

Duality between of CEP-machines and pre-automata (see Theorem 2 and Proposition 2) is the another important result obtained in the paper. It provides some technique for studying of a CEP-machine behaviour.

The important problem for applications is the CEP-machine synthesis problem with the given behaviour, which is specified by a response function. Theorem 3 describes and gives grounds for the method to solve this problem.

Finally, Theorem 4 gives a solution for the problem about the algorithmic realisability of a CEP-machine. This theorem has established that the computability condition of machine handlers is sufficient for the algorithmic realisability of the CEP-machine.

Unfortunately, Theorem 3 does not establish any universal properties for the constructed CEP-machine in contrast to the analogous construction for finite state machine. Therefore the question about universality of this construction for CEP-machines is open now.

The following statement "If a response function is computable then the corresponding CEP-machine is algorithmical realisable" is an example of the next open problem.

We should indicate yet another direction of analysing of CEP-machines. The necessity of this direction follows from the property of CEP-machines marked above: if a CEP-machine is non-trivial then there exist event streams that can be processed by this machine. Hence, studying of such anomalies is an important problem. This studying would be carried out in the probabilistic context that would be provide mean estimations for behavioural anomalies of CEP-machines.

References

1. Babcock, B., Datar, M., Motwani, R.: Load shedding for aggregation queries over data streams. In: ICDE'04: Proceedings of the 20th International Conference Data Engineering, pp. 350–361. IEEE CS, Washington (2004)
2. Bates, J.: Secrets Revealed: Trading Tools Uncover Hidden Opportunities. Global Trading, 14 May 2012. http://fixglobal.com/home/secrets-revealed-trading-tools-uncover-hidden-opportunities/
3. Crosman, P.: Aleri, Ravenpack to Feed News into Trading Algos. Wall Street & Technology (2009). http://www.wallstreetandtech.com/data-management/aleri-ravenpack-to-feed-news-into-tradin/217500395
4. Cugola, G., Margara, A.: Processing flows of information: from data stream to complex event processing. CSUR **44**(3), 15 (2012). ACM Press

5. Cutland, N.: Computability: an introduction to recursive function theory. Cambridge University Press, London (1980)
6. Dokuchaev, M., Novikov, B., Zholtkevych, G.: Alg. Discr. Math. 11(2), 51–63 (2011)
7. Eckert, Michael, Bry, François, Brodt, Simon, Poppe, Olga, Hausmann, Steffen: A CEP babelfish: languages for complex event processing and querying surveyed. In: Helmer, Sven, Poulovassilis, Alexandra, Xhafa, Fatos (eds.) Reasoning in Event-Based Distributed Systems. SCI, vol. 347, pp. 47–70. Springer, Heidelberg (2011)
8. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications, Stamford (2010)
9. Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation. Addison Wesley, New York (2001)
10. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley, Boston (2002)
11. Luckham, D.C.: Event Processing for Business: Organizing the Real-Time Enterprise. John Wiley & Sons Inc., Hoboken (2012)
12. Mikhailova, I., Novikov, B., Zholtkevych, G.: Protoautomata as models of systems with data accumulation. In: Ermolayev, V., et al. (eds.) Proceedings of the 9th International Conference ICTERI 2013, pp. 582–589. CEUR-WS (2013)
13. Michelson, B.: Event-Driven Architecture Overview. Patricia Seybold Group, Boston (2006)
14. Novikov, B., Perepelytsya, I., Zholtkevych, G.: Pre-automata as mathematical models of event flows recognisers. In: Ermolayev, V., et al. (eds.) Proceedings of the 7th International Conference ICTERI 2011, pp. 41–50. CEURS-WS (2011)
15. Perepelytsya, I., Zholtkevych, G.: On some class of mathematical models for static analysis of critical-mission asynchronous systems. Syst. ozbr. ta viysk. tehn. **27**(3), 60–63 (2011)
16. Perepelytsya, I., Zholtkevych, G.: Hierarchic decomposition of pre-machines as models of software system components. Syst. upravl. navig. i zv'iazku. **20**(4), 233–238 (2011)

On Algebraic Properties of Nominative Data and Functions

Volodymyr G. Skobelev[1], Mykola Nikitchenko[2], and Ievgen Ivanov[2](✉)

[1] Institute of Applied Mathematics and Mechanics of NAS of Ukraine, Donetsk, Ukraine
skbv@iamm.ac.donetsk.ua
[2] Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
nikitchenko@unicyb.kiev.ua, ivanov.eugen@gmail.com

Abstract. In the chapter basic properties of nominative data and functions over nominative data (nominative functions) are investigated from the perspective of abstract algebra. A set of all nominative data over arbitrary fixed sets of names and values together with basic operations which include naming, denaming, and overlapping is considered as an algebraic structure and its main properties are studied. Nominative data with complex names satisfy the principle of associative naming and processing. For such data a natural equivalence relation is introduced. Properties of nominative functions (mathematical models of programs over nominative data) and predicates are studied. A notion of nominative stability of nominative functions and predicates is considered. A two-sorted algebra of nominative functions and predicates which generalizes Glushkov algorithmic algebras is introduced and it is proved that the set of nominative stable functions and the set of nominative stable predicates constitute its sub-algebra. The obtained results form a mathematical basis for nominative program logic construction.

Keywords: Glushkos algorithmic algebra · Program algebra · Nominative data · Nominative set · Named set · Nominative function

1 Introduction

To increase software system reliability a number of mathematically based approaches for software (and hardware) development can be used. These approaches are usually referred to as formal methods. Such methods are grounded on different branches of mathematical logic, set theory, automata theory, universal algebra, formal language theory, and other fundamental areas of mathematics. The central idea of formal methods can be described as follows: first, a formal specification of a software system (abstract system model) is described, then this

This work was supported in part by the research project No. 11BF015-02 "Formal specifications and methods of development of reliable software systems", Taras Shevchenko National University of Kyiv, Ukraine.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 117–138, 2014.
DOI: 10.1007/978-3-319-13206-8_6

specification is successively transformed to a concrete specification (system realization), and, at last, correctness of transformations can be proved. The majority of formal methods are based on algebraic approach. This approach has the following two characteristics: (1) the formalism of many-sorted algebras is used to model software systems; (2) special logics based on such many-sorted algebras are used to reason about systems and their transformations. In the literature various kinds of such algebras and logics are described (for instance, see [1]).

In this chapter we use composition-nominative approach [2] to construct algebras of nominative data and function, considered as formal models of program, and investigate properties of these algebras. This approach is grounded on several principles [2–4] including the *Development principle* (from abstract to concrete), *Principle of integrity of intensional and extensional aspects*, *Principle of priority of semantics over syntax*, *Compositionality principle*, *Nominativity principle*.

The latter Nominativity principle states that nominative data [2] are adequate mathematical models of various forms of data that are processed and stored in computing systems. There are several types of nominative data [5,6], and all of them are based on naming relations that associate names and values. The simplest type of nominative data is a nominative set (or named set) [2,5] which is defined as a partial function from an arbitrary set of names to an arbitrary class values. Other types of nominative data represent hierarchical data organizations with simple or complex names.

On the abstract level a computing system is modeled as a partial function that maps nominative data (input data) to nominative data (output data). Such functions are called binominative. More generally, one can consider functions which map nominative data to arbitrary values. Nominative functions can be composed in various ways, e.g. by sequential composition, branching, etc. Operations that construct composed systems from constituents are called compositions.

A set of compositions available to a system developer together with a set of functions obtained from some chosen set of basic functions by applications of compositions form a certain algebraic system (program algebra) which is considered as a semantic model of a computing system development language. A syntax of this development language follows naturally from this semantic model: programs are represented as terms of the described algebra.

The relation of the above mentioned notions to semantics of programming languages can be illustrated on a simple educational programming language WHILE described in [7]. This is an imperative language in which programs are composed from statements that involve boolean and arithmetic expressions. A program state is an assignment of values to variable names. It can be modeled as a nominative set (which is defined on assigned variables and is undefined on unassigned variables). Semantics of a statement can be represented as a (partial) binominative function (a mapping from states to states) and semantics of a boolean or arithmetic expression can be represented as a function on nominative data which takes boolean or, respectively, integer values. Semantics of statements are composed to obtain semantics of a program.

The described approach allows one to investigate semantic properties of programs and develop methods of synthesis of systems with desired properties. In particular, in the works on the composition-nominative approach [2–4] various logical systems for reasoning about properties of computing systems (functions) and solving verification problems have been proposed. Some of them [4] are based on the classical Floyd-Hoare logic [8,9], but extend it to handle a larger class of pre- and post-conditions that includes partial predicates.

Although applications are important for the composition-nominative approach, it is largely a mathematical framework, so it gives a rise to various theoretical questions.

One such question is definition and investigation of algebras of programs over nominative data. This question is the topic of this chapter. A set of functions and a set of predicates over nominative data which can be obtained from basic functions and predicates using compositions such as branching, cycle, etc. [2,6,10] form a two-sorted algebraic structure [11] which generalizes Glushkov algorithmic algebras [12]. However, its main difference and advantage is that its functions are defined over nominative data which give a natural representation of data structures commonly used in programming. We will define and investigate some properties of this algebra in this chapter.

The chapter is organized in the following way: in Sect. 2 we will introduce the necessary notation; in Sect. 3 we will give an overview of the composition-nominative approach; in Sect. 4 we will introduce the formal definition and classification of nominative data; in Sect. 5 we will describe representation of various concrete data structures by nominative data; in Sect. 6 we will introduce the main operations over nominative data; in Sect. 7 we will introduce algebras of nominative data; in Sect. 8 we will consider the notion of nominative equivalence and its relation to algebras of nominative data; in Sect. 9 we will introduce the main compositions of functions and predicates over nominative data and introduce a generalization of Glushkov algorithmic algebras which we call Associative Nominative Glushkov Algorithmic Algebras; in Sect. 10 we introduce the notion of nominative stability of nominative predicates and functions and prove that the sets of nominative stable functions and predicates form a sub-algebra of an Associative Nominative Glushkov Algorithmic Algebra and discuss practical implications of this result.

2 Notation

We will use the following notation:

- 2^A is the power set of a set A;
- $|A|$ means the cardinality of a set A;
- $A \tilde{\to} B$ is the class of partial functions from A to B;
- $f(x) \downarrow$ (or $f(x) \uparrow$) means that a partial function f is defined (or respectively, undefined) on an argument x. The notation $f(x) \downarrow R y$, where R is a binary relation e.g. equality ($=$), inclusion ($\in$), etc. means that $f(x)$ is defined and the relation $f(x) R y$ holds;

– Dom f denotes the domain of a function, i.e. if $f : A \tilde{\to} B$, then $Dom f = A$;
– $dom(f)$ denotes the set of arguments on which f is defined (domain of definedness), i.e. $dom(f) = \{x \mid f(x) \downarrow\}$;
– $im(f) = \{f(x) \mid x \in dom(f)\}$ is the image of a function f;
– $f|_A$ is the restriction of a function f to a set A;
– $f^{-1}(A) = \{x \mid f(x) \downarrow \in A\}$ is the preimage of a set A under a function f;
– $f(x) \cong g(y)$ means the strong equality: if at least one of the values $f(x)$ and $g(y)$ is defined, then the another one is defined and they both are equal.

3 Composition-Nominative Approach to Program Formalization

In order to discuss role of data in programming we should construct adequate models of programs. To tackle this problem we use composition-nominative approach to program formalization [4]. This approach aims to construct a hierarchy of program models of various abstraction levels and generality and is based on the following main principles:

– *Development principle* (from abstract to concrete): program notions should be introduced as a process of their development that starts from abstract understanding and proceeds to more concrete considerations.
– *Principle of integrity of intensional and extensional aspects*: program notions should be presented in the integrity of their intensional and extensional aspects. The intensional aspects in this integrity play a leading role.
– *Principle of priority of semantics over syntax*: program semantic and syntactical aspects should be first studied separately, then in their integrity in which semantic aspects prevail over syntactical ones.
– *Compositionality principle*: programs can be constructed from simpler programs (functions) with the help of special operations, called compositions, which form a kernel of program semantics structures.
– *Nominativity principle*: nominative (naming) relations are basic ones in constructing data.

According to the composition-nominative approach program models are specified as *composition-nominative systems* (CNS). These systems are composed of simpler systems: composition, description, and denotation systems. Composition system defines semantic aspects of programs, description system defines program descriptions (syntactical aspects), and denotation system specifies meanings of descriptions.

We consider semantics of programs as partial functions over class of data processed by programs; compositions are n-ary operations over functions. Thus, a composition system can be specified as two algebras: data algebra and function algebra. Function algebra is the main semantic notion in program formalization. Terms of this algebra define syntax of programs (descriptive system), and ordinary procedure of term interpretation gives a denotation system.

CNS can be used to construct formal models of various programming, specification, and database languages [4,5]. The program models represented by CNS are mathematically simple, but specify program semantics rather adequately. They are classified in accordance with the levels of abstraction of their parameters: data, functions, and compositions. Usually abstraction levels of program models are induced by abstraction levels of data.

Data are considered on three levels: abstract, boolean, and nominative. On the abstract level data are treated as "black boxes", thus no information can be extracted. On the boolean level to abstract data new data considered as "white boxes" are added. Usually, these are logical values T (true) and F (false) from the set *Bool*. On the nominative level data are considered as "grey boxes", constructed of "black boxes" (values) and "white boxes" (names) with the help of naming relations. The last level is the most interesting for programming. Data of this level are called *nominative data*.

Concretizations of nominative data can represent various data structures, such as records, arrays, lists, relations, etc. Thus, we can formulate the following *data representativity principle*: program data can be represented as concretizations of nominative data. To support this principle later in the chapter we will give representations of various data structures using nominative data.

We start with rigorous definitions of various classes of nominative data.

4 Classification of Nominative Data

Nominative data are built over classes of names V and atoms (basic values) A with the help of naming relations. Thus in the first approximation, a nominative data d is either an atom from A, or has the form $[v_1 \mapsto d_1, ..., v_n \mapsto d_n]$, where $v_1, ..., v_n$ are distinct names from V and $d_1, d_2, ...d_n$ are atoms or possibly other nominative data.

To define nominative data let us denote by $V \xrightarrow{n} B$ the class of all partial functions from a set of names V to a set of values B which have a finite graph.

Nominative data are classified in accordance with the following parameters:

- *values* can be simple (unstructured) or complex (structured),
- *names* can be simple (unstructured) or complex (structured).

The described parameters give 4 types of nominative data. There can be different ways to define the notion of a complex name. We will follow the principle of development from abstract to concrete and consider the simplest case of name construction and processing. This means that we will consider complex names as *sequences* of simple names that satisfy the *associativity* property. In other words, names are constructed with the help of concatenation operation which is associative by definition. This allows us to formulate *the principle of associative construction and processing of complex names*: complex names are constructed from simple names using concatenation and data with complex names have to be processed by operations that take into account associativity of names. Besides, we will require that data with complex names satisfy *the principle of unambiguous associative naming* which means that one complex name should have at most one corresponding value in any given data.

The formal definitions of nominative data of different types and the corresponding examples are given below.

- $D_0 = V \xrightarrow{n} A$. This is the simplest type of nominative data. For example,

  $$[u \mapsto 1, v \mapsto 2].$$

  We will say that members of D_0 are data of the type TND_{SS} (data with simple names and simple values).
- $D_1 = ND(V, A)$, where $ND(V, A)$ is defined as

  $$ND(V, A) = \bigcup_{k \geq 0} ND_k(V, A),$$

  where

  $$ND_0(V, A) = A \cup \{\emptyset\},$$

  $$ND_{k+1}(V, A) = A \cup \left(V \xrightarrow{n} ND_k(V, A) \right), \quad k \geq 0.$$

  Note that here we denote by $\emptyset$ the empty nominative data (this notation is also used for the empty set). For the empty nominative data we will also use the notation [].
  Data from this class are hierarchically constructed, for example,

  $$[u \mapsto 1, v \mapsto [w \mapsto 2]].$$

  Such data can be represented by oriented trees (of varying arity) with arcs labelled by names and with leafs labelled by atoms. A *path* is a nonempty finite sequence $(v_1, v_2, ..., v_k)$ of names $v_1, ..., v_k \in V$. For a given data d a *value of a path* $(v_1, v_2, ..., v_k)$ in d is defined by the following expression

  $$d(v_1, v_2, ..., v_k) \cong (...((d(v_1))(v_2))...(v_k)).$$

  We say that a path $(v_1, v_2, ..., v_k)$ is *a path in a data* $d \in ND(V, A)$, if a value of $(v_1, v_2, ..., v_k)$ in d is defined, i.e. $d(v_1, v_2, ..., v_k) \downarrow$ (a path in data corresponds to a path from the root to a node in an oriented tree).
  A *terminal path* in a data $d \in ND(V, A)$ is a path in d such that its value belongs to $A \cup \{\emptyset\}$.
  The least k such that $d \in ND_k(V, A)$ is called the *rank* of a data d.
  We will say that members of D_1 are data of the type TND_{SC} (data with simple names and complex values).
- $D_2 = NDVS(V, A)$, where $NDVS(V, A)$ is the set of all elements of $A \cup (V^+ \xrightarrow{n} A)$ such that either $d \in A$, or $d \in V^+ \xrightarrow{n} A$ and all strings from $dom(d)$ are pairwise incomparable in the sense of the prefix relation (*the principle of unambiguous associative naming*).

For example

$$[uv \mapsto 1, uw \mapsto 2, w \mapsto 3].$$

Data of this class have *complex names* i.e. names that are strings in alphabet V.

We will say that members of D_2 are data of the type TND_{CS} (data with complex names and simple values).

– $D_3 = NDVC(V, A)$, where $NDVC(V, A)$ is the class of all data $d \in ND(V^+, A)$ such that for any two paths $(u_1, u_2, ..., u_k)$ and $(v_1, v_2, ..., v_l)$ in d, neither of which is a prefix of another, the words $u_1u_2...u_k$ and $v_1v_2...v_l$ are incomparable is sense of the prefix relation (*principle of unambiguous associative naming*). Such data are also called *complex-named data* [6]. For example,

$$[uv \mapsto 1, w \mapsto [uw \mapsto \emptyset]].$$

These data are hierarchically constructed data with complex names and unambiguous naming.

We will say that members of D_3 are data of the type TND_{CC} (data with complex names and complex values).

5 Examples of Representations of Data Structures by Nominative Data

In this section we present some concrete examples to support data representativity principle: program data can be adequately represented as concretizations of nominative data.

Below we describe representations of various commonly used data structures by nominative data. Representations are chosen in such way that the basic operations over data structures correspond to simple combinations of basic operations over nominative data.

– Array $(a_1, a_2, ..., a_n)$ of elements of type T_0.
Nominative data type: TND_{SS}, $V = \{1, 2, \ldots\}$, $A = T_0$.
Representation:

$$[1 \mapsto a_1, 2 \mapsto a_2, \ldots, n \mapsto a_n].$$

– Two-dimensional array $\begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} \\ \ldots & \ldots & \ldots & \ldots. \\ a_{n,1} & a_{n,2} & \ldots & a_{n,m} \end{pmatrix}$ of elements of type T_0.
Nominative data type: TND_{CS} (complex names), $V = \{1, 2, \ldots\}$, $A = T_0$.
Representation:

$$[i.j \mapsto a_{i,j} \mid i = 1, \ldots, n,\ j = 1, \ldots, m]$$

(in this notation the dot "." separates symbols in a complex name).

– Jagged array $((a_{i,j})_{j=1}^{m})_{i=1}^{n}$ of elements of type T_0.
Nominative data type: TND_{SC}, $V = \{1, 2, \ldots\}$, $A = T_0$.
Representation:

$$[i \mapsto [j \mapsto a_{i,j} \mid j = 1, \ldots, m] \mid i = 1, \ldots, n].$$

– Associative array $(a_k)_{k \in K}$ of elements of type T_0, where K is a set of keys.
Nominative data type: TND_{SS}, $V = K$, $A = T_0$.
Representation:

$$[k \mapsto a_k \mid k \in K].$$

– Table of the form (with Key being a key attribute):

Key	Attr1	Attr2
key_1	val_{11}	val_{21}
key_2	val_{12}	val_{22}

Nominative data type: TND_{SC}, $V = \{Attr_1, Attr_2\}$.
Representation:

$$[key_1 \mapsto [Attr_1 \mapsto val_{11}, Attr_2 \mapsto val_{12}],$$
$$key_2 \mapsto [Attr_1 \mapsto val_{21}, Attr_2 \mapsto val_{22}]].$$

– A list of elements $e_1, e_2, \ldots e_n$.
Nominative data type: TND_{SC}, $V = \{data, next\}$.
Representation:

$$[data \mapsto e_1, next \mapsto [data \mapsto e_2, next \mapsto [\ldots data \mapsto e_n, next \mapsto \emptyset \ldots]]].$$

– A circular list of elements $e_1, e_2, ...e_n$.
Nominative data type: TND_{SC}, $V = \{head, memory, data, next\} \cup \{loc_1, loc_2, loc_3, \ldots loc_n\}$.
Representation:

$$\begin{aligned}[head \mapsto loc_1, memory \mapsto [&\\ &loc_1 \mapsto [data \mapsto e_1, next \mapsto loc_2],\\ &loc_2 \mapsto [data \mapsto e_2, next \mapsto loc_3],\\ &\ldots\\ &loc_n \mapsto [data \mapsto e_n, next \mapsto loc_1]]].\end{aligned}$$

– A binary tree with nodes labelled by elements $e_1, e_2, e_3, ...$
Nominative data type: TND_{SC}, $V = \{data, left, right\}$.
Representation:

$$\begin{aligned}&[data \mapsto e_1,\\ &left \mapsto [data \mapsto e_2, left \mapsto [\ldots], right \mapsto [\ldots]],\\ &right \mapsto [data \mapsto e_3, left \mapsto [\ldots], right \mapsto [\ldots]\\ &]].\end{aligned}$$

– Data types based on free algebras (simple forms of inductive data types [13]). Such data are widely used in (partly) functional programming languages (e.g. ML, Haskell). They are suitable for representing list- and tree-like data. Such types can be interpreted as carrier sets of (many-sorted) free algebras or as sets of terms.

  Let S be a finite set of elements called *sorts*. Let Σ be finite *signature* $\{c_1, c_2, ..., c_n\}$, where c_i are (names of) *constructors*. Suppose that each constructor c_i has (unique) associated *type* of the form $s_1 \times \ldots \times s_m \to s$, where $s_i, s \in S$, $m \geq 0$. Let D be a free algebra with the signature Σ which is freely generated by some set A. Then the members of the carrier set of D can be considered as elements of an inductive data type. Data of such type can be processed by means of application of constructors and pattern matching (for data decomposition).

  Then one can define a mapping τ which gives a representation of data from D by nominative data of the type TND_{SC} over the classes of names $V = \{constructor, 1, 2, 3, \ldots\}$ and the basic values A:

  - $\tau(x) = x$, if $x \in A$.
  - $\tau(x) = [constructor \mapsto c, 1 \mapsto \tau(x_1), 2 \mapsto \tau(x_2), ..., n \mapsto \tau(x_n)]$, if x has a form $c(x_1, x_2, \ldots, x_n)$, where $c \in \Sigma$.

  Applications of constructors and pattern matching on inductive data types correspond to naming and denaming (and name equality checking) on nominative data.

6 Operations over Nominative Data

The main operations over nominative data are the operations of *denaming* (taking the value of a name), *naming* (assigning a new value to a name), and *overlapping*.

In this section we will define these operations for data of the types TND_{CS}, TND_{SC}, and TND_{CC}. Analogous operations can be rather straightforwardly defined on data of the type TND_{SS}, but we will not consider them in this chapter.

We will use the same symbols for denoting denaming, naming, and overlapping for each type of data TND_{CS}, TND_{SC}, and TND_{CC}. The interpretation of the operation symbols should be clear from the context.

Let V and A be fixed sets of names and values.

Definition 1 (Denaming).

(1) For nominative data of the type TND_{SC}, (associative) denaming is an unary operation $v \Rightarrow_a$ with a parameter $v \in V^+$ defined by induction by the length of v (denoted as $|v|$) as follows:
  - *if $|v| = 1$, then $v \Rightarrow_a (d) \cong d(v)$;*
  - *if $|v| = n > 1$, then $v \Rightarrow_a (d) \cong v_1 \Rightarrow_a (x \Rightarrow_a (d))$, where $v = xv_1$, $x \in V$, $v_1 \in V^{n-1}$.*

(2) For nominative data of the types TND_{CS} and TND_{CC}, (associative) denaming is an unary operation $v \Rightarrow_a$ with a parameter $v \in V^+$ defined by induction by the length of v:
– *if $|v| = 1$, then*

$$v \Rightarrow_a (d) \cong \begin{cases} d(v), & \text{if } d(v) \downarrow; \\ d/v, & \text{if } d(v) \uparrow \text{ and } d/v \neq \emptyset; \\ \text{undefined}, & \text{if } d(v) \uparrow \text{ and } d/v = \emptyset, \end{cases}$$

where

$$d/u = [v_1 \mapsto d(v) \mid d(v) \downarrow, v = uv_1, v_1 \in V^+]$$

(division of a data by a name);
– *if $|v| = n > 1$, then*

$$v \Rightarrow_a (d) \cong v_1 \Rightarrow_a (x \Rightarrow_a (d)),$$

where $v = xv_1$, $x \in V$, $v_1 \in V^+$, $|v_1| = n - 1$.

The following examples illustrate this definition:

$$u \Rightarrow_a ([u \mapsto 1, v \mapsto 2]) = 1;$$

$$(uv) \Rightarrow_a ([u \mapsto [vw \mapsto 1, u \mapsto 2]]) = [w \mapsto 1].$$

The name of this operation originates from the following property (*associativity*) [6]:

$$u \Rightarrow_a (d) \cong u_n \Rightarrow_a (u_{n-1} \Rightarrow_a (\ldots u_1 \Rightarrow_a (d)\ldots))$$

for all complex names $u, u_1, u_2, \ldots, u_n \in V^+$ such that $u = u_1 u_2 \ldots u_n$.

Definition 2 (Naming).

(1) For nominative data of the type TND_{SC}, naming is an unary operation $\Rightarrow v$ with a parameter $v \in V^+$ defined inductively by the length of v as follows:
$\Rightarrow v(d) = [v \mapsto d]$, if $v \in V$;
$\Rightarrow v(d) = [v_1 \mapsto (\Rightarrow v_2(d))]$, if $v = v_1 v_2$, $v_1 \in V$ and $v_2 \in V^+$.
(2) For nominative data of the type TND_{CS}, naming is an unary operation $\Rightarrow v$ with a parameter $v \in V^+$ defined as follows:
$\Rightarrow v(d) = [v \mapsto d]$, if $d \in A \cup \{\emptyset\}$;
$\Rightarrow v(d) = [vu \mapsto d(u) \mid u \in dom(d)]$, if $d \notin A \cup \{\emptyset\}$.
(3) For nominative data of the type TND_{CC}, naming is an unary operation $\Rightarrow v$ with a parameter $v \in V^+$ such that $\Rightarrow v(d) = [v \mapsto d]$.

Overlapping can be intuitively treated as an updating operation which updates values in the first argument with the values of the second argument taking into account their names.

For the types of nominative data with complex names and/or values different overlapping operations can be considered. We will define two kinds of overlapping: global and local overlapping. Global (associative or structural) overlapping

∇_a updates several values while the local one ∇_a^v (with a parameter name v) updates only one value with complex name v.

The global overlapping can be used for formalization of procedures calls and the local operation formalizes the assignment operator in programming languages. Intuitively, this operation joins two data and resolves name conflicts in favour of its *second* argument.

Definition 3 (Global overlapping).

(1) For nominative data of the type TND_{SC}, global overlapping is a partial binary operation ∇_a such that
$d_1 \nabla_a d_2 = d_2 \cup d_1|_{dom(d_1) \setminus dom(d_2)}$, if $d_1 \notin A$ and $d_2 \notin A$;
$d_1 \nabla_a d_2 \uparrow$, if $d_1 \in A$ or $d_2 \in A$.

(2) For nominative data of the type TND_{CS}, global overlapping is a binary operation ∇_a such that
$d_1 \nabla_a d_2 = d_2 \cup d_1|_{dom(d_1) \setminus (dom(d_2) V^*)}$;
$d_1 \nabla_a d_2 \uparrow$, if $d_1 \in A$ or $d_2 \in A$.
where $dom(d_2)V^*$ denotes the all words of the form uv, where $u \in dom(d_2)$ and $v \in V^*$ (i.e. an arbitrary, possibly empty word in the alphabet V).

(3) For nominative data of the type TND_{CC}, (global) overlapping is a binary operation ∇_a defined inductively by the rank of the first argument as follows. Let

$$NDVC_k(V, A) = NDVC(V, A) \cap ND_k(V^+, A)$$

be the data from the set $NDVC(V, A)$ the rank of which is less than or equal to k.
Induction base of the definition. If $d_1 \in NDVC_0(V, A)$, then

$$d_1 \nabla_a d_2 \cong \begin{cases} d_2, & \text{if } d_1 = \emptyset \text{ and } d_2 \in NDVC(V, A) \setminus A; \\ \text{undefined}, & \text{if } d_1 \in A \text{ or } d_2 \in A. \end{cases}$$

Induction step of the definition. Assume that the value $d_1 \nabla_a d_2$ is already defined for all d_1, d_2 such that $d_1 \in NDVC_k(V, A)$. Let

$$d_1 \in NDVC_{k+1}(V, A) \setminus NDVC_k(V, A).$$

Then $d_1 \nabla_a d_2 = d$, where d is defined by its values on names $u \in V^+$ as follows:

(1) $d(u) = d_2(u)$, if $u \in dom(d_2)$ and u does not have a proper prefix which belongs to $dom(d_1)$;
(2) $d(u) = d_1(u) \nabla_a (d_2/u)$, if $d_1(u)$ is defined and does not belong to A and u is a proper prefix of some element of $dom(d_2)$, where $d_2/u = [v_1 \mapsto d_2(v) \mid d_2(v) \downarrow, v = uv_1, v_1 \in V^+]$ is the division of a data by a name;
(3) $d(u) = d_2/u$, if $d_1(u)$ is defined and belongs to A and u is a proper prefix of some element of $dom(d_2)$;
(4) $d(u) = d_1(u)$, if $d_1(u)$ is defined and u is not comparable (in the sense of the prefix relation) with any element of $dom(d_2)$;
(5) $d(u) \uparrow$, otherwise.

The global overlapping on the data of the type TND_{CC} has the following properties [6]:

- $[u \mapsto d_1]\nabla_a[v \mapsto d_2] = [u \mapsto d_1, v \mapsto d_2], \quad u, v \in V, \quad u \neq v;$
- $[uv \mapsto d_1]\nabla_a[u \mapsto d_2] = [u \mapsto d_2]$, $u, v \in V^+$, i.e. the value under a name u in second argument overwrites the value under names in first argument, which are extensions of u;
- $[u \mapsto d_1]\nabla_a[uv \mapsto d_2] = [u \mapsto (d_1\nabla_a[v \mapsto d_2])]$, if $u, v \in V^+$, $d_1 \notin A$, i.e. the value under a name uv in second argument modifies values under prefixes of uv in first argument;
- $\emptyset\nabla_a d = d\nabla_a\emptyset = d$, if $d \notin A$;
- $d_1\nabla_a d_2 \uparrow$, if $d_1 \in A$ or $d_2 \in A$.

Definition 4 (Local overlapping).

(1) For nominative data of the type TND_{CS} local overlapping is a binary operation ∇_a^v with a parameter $v \in V^+$ defined as follows:

$$d_1\nabla_a^v d_2 \cong d_1\nabla_a(\Rightarrow v(d_2)).$$

(2) For nominative data of the type TND_{SC} local overlapping is a binary operation ∇_a^v with a parameter $v \in V^+$ defined inductively by the length of v as follows:

- *if $v \in V$, then*

$$d_1\nabla_a^v d_2 \cong d_1\nabla_a[v \mapsto d_2];$$

- *if $v = v_1v_2$, where $v_1 \in V$ and $v_2 \in V^+$, and $d_1(v_1)\downarrow$ and $d_1(v_1) \notin A$, then*

$$d_1\nabla_a^v d_2 \cong d_1\nabla_a[v_1 \mapsto d_1(v_1)\nabla_a^{v_2} d_2];$$

- *if $v = v_1v_2v_3...v_n$, where $v_i \in V$, and $d_1(v_1)\uparrow$ or $d_1(v_1) \in A$, then*

$$d_1\nabla_a^v d_2 \cong d_1\nabla_a[v_1 \mapsto [v_2 \mapsto ... \mapsto [v_n \mapsto d_2]...].$$

(3) For nominative data of the type TND_{CC} local overlapping is a binary operation ∇_a^v with a parameter $v \in V^+$ defined as follows:

$$d_1\nabla_a^v d_2 \cong d_1\nabla_a(\Rightarrow v(d_2)).$$

7 Algebras of Nominative Data

The operations on nominative data of different classes defined earlier give a rise to algebraic structures defined below. We will focus on the types TND_{CS}, TND_{SC}, TND_{CC}, because the type TND_{SS} is rather simple and it is straightforward to define an algebra on it.

Let V and A be fixed sets of names and basic values.

Definition 5. *An algebra of nominative data of the type TND_{SC} is an algebra*

$$NDA_{SC}(V, A) =< ND(V, A); (v \Rightarrow_a)_{v \in V^+}, (\Rightarrow v)_{v \in V^+}, (\nabla_a^v)_{v \in V^+} >$$

with the carrier set $ND(V, A)$ (data) and the following operations:

- *a family of partial unary associative denaming operations* $v \Rightarrow_a: ND(V, A) \tilde{\rightarrow} ND(V, A)$, $v \in V^+$;
- *a family of unary complex naming operations* $\Rightarrow v : ND(V, A) \rightarrow ND(V, A)$, $v \in V^+$;
- *a family of partial binary local overlapping operations* $\nabla_a^v : ND(V, A) \times ND(V, A) \tilde{\rightarrow} ND(V, A)$, $v \in V^+$.

Definition 6. *An algebra of nominative data of the type TND_{CS} is an algebra*

$$NDA_{CS}(V, A) =< NDVS(V, A); (v \Rightarrow_a)_{v \in V^+}, (\Rightarrow v)_{v \in V^+}, (\nabla_a^v)_{v \in V^+} >$$

with the carrier set $NDVS(V, A)$ (data) and the following operations:

- *a family of partial unary associative denaming operations* $v \Rightarrow_a: NDVS(V, A) \tilde{\rightarrow} NDVS(V, A)$, $v \in V^+$;
- *a family of unary complex naming operations* $\Rightarrow v : NDVS(V, A) \rightarrow NDVS(V, A)$, $v \in V^+$;
- *a family of partial binary local overlapping operations* $\nabla_a^v : NDVS(V, A) \times NDVS(V, A) \tilde{\rightarrow} NDVS(V, A)$, $v \in V^+$.

Definition 7. *An algebra of nominative data of the type TND_{CC} is an algebra*

$$NDA_{CC}(V, A) =< NDVC(V, A); (v \Rightarrow_a)_{v \in V^+}, (\Rightarrow v)_{v \in V^+}, (\nabla_a^v)_{v \in V^+} >$$

with the carrier set $NDVC(V, A)$ (data) and the following operations:

- *a family of partial unary associative denaming operations* $v \Rightarrow_a: NDVC(V, A) \tilde{\rightarrow} NDVC(V, A)$, $v \in V^+$;
- *a family of unary complex naming operations* $\Rightarrow v : NDVC(V, A) \rightarrow NDVC(V, A)$, $v \in V^+$;
- *a family of partial binary local overlapping operations* $\nabla_a^v : NDVC(V, A) \times NDVC(V, A) \tilde{\rightarrow} NDVC(V, A)$, $v \in V^+$.

8 Nominative Equivalence

Complex-named data have an inherent associated hierarchical naming structure. However, from the viewpoint of their information content that can be obtained from such data using the operation of associative denaming, some pairs of complex-named data like $[v_1 \mapsto [v_2 \mapsto [v_3 \mapsto 1]]]$ and $[v_1 v_2 v_3 \mapsto 1]$ that have different hierarchical naming structure are essentially equivalent.

The following equivalence relation on complex-named data called nominative equivalences formalizes this observation.

Definition 8 (Paths and terminal paths in data).

(1) A path in a complex-named data $d \in NDVC(V,A)$ is a nonempty sequence $(v_1, v_2, ..., v_n)$ of words from V^+ such that $((d(v_1))(v_2)...)(v_n)$ is defined. The value $((d(v_1))(v_2)...)(v_n)$ is called the value of the path $(v_1, v_2, ..., v_n)$ in d.
(2) A path in a complex-named data $d \in NDVC(V,A)$ is called a terminal path, if its value in d belongs to $A \cup \{\emptyset\}$.

Definition 9 (Nominative inclusion and equivalence).

(1) A complex-named data $d_1 \in NDVC(V,A)$ is nominatively included in a complex-named data $d_2 \in NDVC(V,A)$, if either $d_1, d_2 \in A$ and $d_1 = d_2$, or $d_1, d_2 \notin A$ and for each terminal path $(v_1, v_2, ..., v_n)$ in d_1 there exists a terminal path $(v'_1, v'_2, ..., v'_m)$ in d_2 such that $v_1 v_2 ... v_n = v'_1 v'_2 ... v'_m$ and the values of $(v_1, v_2, ..., v_n)$ in d_1 and $(v'_1, v'_2, ..., v'_m)$ in d_2 coincide.
(2) Two complex-named data $d_1, d_2 \in NDVC(V,A)$ are nominative equivalent (denoted as $d_1 \approx d_2$), if d_1 is nominatively included in d_2 and d_2 is nominatively included in d_1.

It is known [5] that nominative inclusion is a preorder on $NDVC(V,A)$ and nominative equivalence is an equivalence relation on $NDVC(V,A)$.

Nominative equivalent data may have different hierarchical naming structures, but they turn into the same TND_{CS} data when they are flattened, e.g.:

- $[v_1 \mapsto [v_2 v_3 \mapsto 1, v_2 v_4 \mapsto 2]] \approx [v_1 v_2 v_3 \mapsto 1, v_1 v_2 v_4 \mapsto 2]$;
- $[v_1 v_2 \mapsto [v_3 \mapsto 1, v_4 \mapsto 2]] \approx [v_1 v_2 v_3 \mapsto 1, v_1 v_2 v_4 \mapsto 2]$.

Theorem 1. *Nominative equivalence is a congruence on $NDA_{CC}(V,A)$.*

Proof. From [5] it follows that for each pair of $d_1, d_2 \in NDVC(V,A)$ the operations of naming and associative denaming satisfy the following property (called *nominative stability*): if $d_1 \approx d_2$ and an operation is defined on d_1, then it is defined on d_2 and its values on d_1 and d_2 are nominative equivalent. Similarly, from [5] it follows that if $v \in V^+$, $d_1, d_2 \notin A$, $d_1 \approx d'_1$, $d_2 \approx d'_2$, then $d_1 \nabla_a^v d_2 \approx d'_1 \nabla_a^v d'_2$. This implies that nominative equivalence is a congruence on $NDA_{CC}(V,A)$. □

This result allows us to construct a factor (quotient) algebra of $NDA_{CC}(V,A)$.

Let us denote by $NDA_{CC}^{\approx}(V,A)$ the factor algebra of $NDA_{CC}(V,A)$ by the nominative equivalence $\approx$.

Lemma 1. *$NDA_{SC}(V,A)$ is isomorphic to $NDA_{CC}^{\approx}(V,A)$.*

Proof (Sketch). Let us define a function $\iota_{13}^{\approx} : ND(V,A) \to 2^{NDVC(V,A)}$ such that $\iota_{13}^{\approx}(d) = \{d' \in NDVC(V,A) \mid d' \approx d\}$ (note that $ND(V,A) \subset NDVC(V,A)$). It is easy to show that each class of nominative equivalence has a unique member from $ND(V,A)$ and the operations on equivalence classes correspond to the operations on TND_{CS} data under this mapping. Then $\iota_{13}^{\approx}$ is an isomorphism from $NDA_{CS}(V,A)$ to $NDA_{CC}^{\approx}(V,A)$. □

Lemma 2. *$NDA_{CS}(V, A)$ is isomorphic to $NDA^{\approx}_{CC}(V, A)$.*

Proof (Sketch). Let us define a function $\iota^{\approx}_{23} : NDVS(V, A) \to 2^{NDVC(V,A)}$ as follows: $\iota^{\approx}_{23}(d) = \{d' \in NDVC(V, A) \mid d' \approx d\}$ (note that $NDVS(V, A) \subset NDVC(V, A)$). It is easy to show that each class of nominative equivalence has a unique member from $NDVS(V, A)$ and the operations on equivalence classes correspond to the operations on TND_{SC} data under this mapping. Then $\iota^{\approx}_{23}$ is an isomorphism from $NDA_{SC}(V, A)$ to $NDA^{\approx}_{CC}(V, A)$. □

Theorem 2. *Algebras $NDA_{SC}(V, A)$, $NDA_{CS}(V, A)$, and $NDA^{\approx}_{CC}(V, A)$ are isomorphic.*

Proof. Follows immediately from Lemmas 1 and 2. □

Let us denote by $D_3/{\approx}$ the carrier set of $NDA^{\approx}_{CC}(V, A)$.

Let $\iota^{\approx}_{13}$ and $\iota^{\approx}_{23}$ be the isomorphisms defined in the proofs of Lemmas 1 and 2. We will call $(\iota^{\approx}_{13})^{-1}$ the *layering* operation and $(\iota^{\approx}_{23})^{-1}$ the *flatting* operation.

For each $d \in D_3$ let $\iota^{\approx}_{3}(d)$ be the $\approx$-equivalence class of d.

Let us introduce the following inclusion maps for classes of nominative data: $\iota_{01} : D_0 \hookrightarrow D_1$, $\iota_{02} : D_0 \hookrightarrow D_2$, $\iota_{13} : D_1 \hookrightarrow D_3$, $\iota_{23} : D_2 \hookrightarrow D_3$.

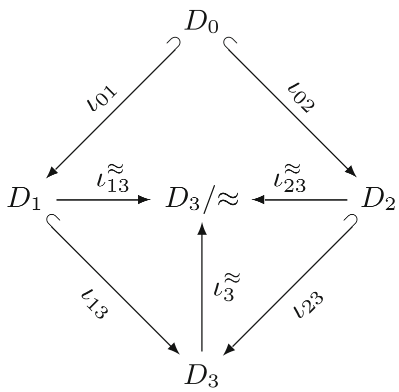


Fig. 1. Classes of nominative data and mappings between them.

Then it is easy to check that the diagram shown in Fig. 1 is commutative. This diagram illustrates the types of nominative data and mappings between them.

9 Compositions of Functions and Predicates over Nominative Data

Let V and A be fixed sets of basic names and values.

Denote

$$Pr_{CC}(V, A) = NDVC(V, A) \tilde{\to} \{T, F\},$$

$$Fn_{CC}(V, A) = NDVC(V, A) \tilde{\to} NDVC(V, A),$$

where T and F denote logical constants (true and false). We will assume that they do not belong to $NDVC(V, A)$.

We will call the elements of $Pr_{CC}(V, A)$ (*partial nominative*) *predicates* and the elements of $Fn_{CC}(V, A)$ *(partial binominative) functions*.

Similarly, let us denote the following sets of functions and predicates on data of the types TND_{SC} and TND_{CS}:

$$Pr_{SC}(V, A) = ND(V, A)\tilde{\to}\{T, F\},$$

$$Fn_{SC}(V, A) = ND(V, A)\tilde{\to}ND(V, A),$$

$$Pr_{CS}(V, A) = NDVS(V, A)\tilde{\to}\{T, F\},$$

$$Fn_{CS}(V, A) = NDVS(V, A)\tilde{\to}NDVS(V, A).$$

Let us consider conventional logical and programming compositions of predicates and functions. These compositions are defined analogously for functions and predicates over each type of nominative data, so below we give a common definition in which the symbol Fn should be understood as one of Fn_{CC}, Fn_{CS}, or Fn_{SC}, and the symbol Pr should be understood as one of Pr_{CC}, Pr_{CS}, Pr_{SC} However, within each definition of a composition below all occurrences of the symbols Fn, Pr refer to sets of functions and predicates over nominative data of the same type, i.e. either TND_{CC}, or TND_{CS}, or TND_{SC}.

Let us denote by $\bar{U}$ the set of all tuples $(u_1, u_2, ..., u_n)$, $n \geq 1$ of complex names from V^+ such that whenever $i \neq j$, u_i and u_j are incomparable in the sense of the prefix relation.

- Sequential composition of functions (denoted using the infix notation)

$$\bullet : Fn(V, A) \times Fn(V, A) \to Fn(V, A)$$

  is defined as follows: for all $f, g \in Fn(V, A)$ and data d

$$(f \bullet g)(d) \cong g(f(d)).$$

- Prediction composition [12] (denoted using the infix notation)

$$\cdot : Fn(V, A) \times Pr(V, A) \to Pr(V, A)$$

  is defined as follows: for all $f \in Fn(V, A)$, $p \in Pr(V, A)$, and data d

$$(f \cdot p)(d) \cong p(f(d)).$$

- Assignment composition $Asg^u : Fn(V, A) \to Fn(V, A)$ with a parameter $u \in V^+$ is defined as follows: for each $f \in Fn(V, A)$ and data d,

$$(As^u(f))(d) \cong d\nabla_a^u f(d)$$

  (where ∇_a^u refers to the local overlapping on the corresponding type of nominative data).

– The composition of superposition into a function

$$S_F^{u_1,u_2,\dots,u_n} : Fn(V,A) \times (Fn(V,A))^n \to Fn(V,A)$$

with parameters $n \geq 1$ and $u_1,...,u_n \in V^+$ such that $(u_1,...,u_n) \in \bar{U}$ is defined as follows:

$$S_F^{u_1,u_2,\dots,u_n}(f,f_1,...,f_n)(d) \cong f(...((d\nabla_a^{u_1} f_1(d))\nabla_a^{u_2} f_2(d))...\nabla_a^{u_n} f_n(d))...).$$

We will also use the following notation for this composition: for each tuple $\bar{u} = (u_1,u_2,...,u_n) \in \bar{U}$, $S_F^{\bar{u}}$ denotes $S_F^{u_1,u_2,\dots,u_n}$.

– The composition of superposition into a predicate

$$S_P^{u_1,u_2,\dots,u_n} : Pr(V,A) \times (Fn(V,A))^n \to Pr(V,A)$$

with parameters $n \geq 1$ and $u_1,...,u_n \in V^+$ such that $(u_1,...,u_n) \in \bar{U}$ is defined as follows:

$$S_P^{u_1,u_2,\dots,u_n}(p,f_1,...,f_n)(d) \cong p(...((d\nabla_a^{u_1} f_1(d))\nabla_a^{u_2} f_2(d))...\nabla_a^{u_n} f_n(d))...).$$

We will also use the following notation for this composition: for each tuple $\bar{u} = (u_1,u_2,...,u_n) \in \bar{U}$, $S_P^{\bar{u}}$ denotes $S_P^{u_1,u_2,\dots,u_n}$.

– Branching composition $IF : Pr(V,A) \times Fn(V,A) \times Fn(V,A) \to Fn(V,A)$ is defined as follows: for each $p \in Pr(V,A)$ and $f,g \in Fn(V,A)$:

$$IF(p,f,g)(d) \cong \begin{cases} f(d), & \text{if } p(d) \downarrow= T; \\ g(d), & \text{if } p(d) \downarrow= F; \\ \text{undefined}, & \text{if } p(d) \uparrow . \end{cases}$$

– Cycle composition $WH : Pr(V,A) \times Fn(V,A) \to Fn(V,A)$ is defined as follows: for each $p \in Pr(V,A)$, $f \in Fn(V,A)$, and data d:
$WH(p,f)(d) \downarrow= f^{(n)}(d)$, if there exists $n \geq 0$ such that $(f^{(i)} \cdot p)(d) \downarrow= T$ for all $i \in \{0,1,...,n-1\}$ and $(f^{(n)} \cdot p)(d) \downarrow= F$, where $f^{(n)}$ denotes a n-times sequential composition of f with itself (assuming that $f^{(0)}$ is the identity function), and $WH(p,f)(d)$ is undefined otherwise.

– Negation $\neg : Pr(V,A) \to Pr(V,A)$ is a composition such that for each $p \in Pr(V,A)$ and data d:

$$(\neg p)(d) \cong \begin{cases} T, & \text{if } p(d) \downarrow= F; \\ F, & \text{if } p(d) \downarrow= T; \\ \text{undefined}, & \text{if } p(d) \uparrow . \end{cases}$$

– Disjunction $\vee : Pr(V,A) \times Pr(V,A) \to Pr(V,A)$ is a composition defined as follows: for each $p_1,p_2 \in Pr(V,A)$ and data d:

$$(p_1 \vee p_2)(d) \cong \begin{cases} T, & \text{if } p_1(d) \downarrow= T \text{ or } p_2(d) \downarrow= T; \\ F, & \text{if } p_1(d) \downarrow= F \text{ and } p_2(d) \downarrow= F; \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

– Identity composition $Id : Fn(V, A) \rightarrow Fn(V, A)$ is defined as follows: $Id(f) = f$ for all $f \in Fn(V, A)$.
– True constant predicate (null-ary composition) $True \in Pr(V, A)$ is defined as follows: $True(d) \downarrow = T$ for all data d.
– Bottom function (null-ary composition) $\perp_F \in Fn(V, A)$ is defined as follows: $\perp_F (d) \uparrow$ for all data d.
– Bottom predicate (null-ary composition) $\perp_P \in Pr(V, A)$ is defined as follows: $\perp_P (d) \uparrow$ for all data d.
– Name checking predicate (null-ary composition) $u! \in Pr(V, A)$ with a parameter $u \in V^+$ is defined as follows:

$$u!(d) = \begin{cases} T, & \text{if } u \Rightarrow_a (d) \downarrow; \\ F, & \text{if } u \Rightarrow_a (d) \uparrow . \end{cases}$$

– Empty constant function (null-ary composition) $Empty \in Fn(V, A)$ is defined as follows: $Empty(d) = \emptyset$ for all data d.
– Emptiness checking predicate (null-ary composition) $IsEmpty \in Fn(V, A)$ is defined as follows:

$$IsEmpty(d) = \begin{cases} T, & \text{if } d = \emptyset; \\ F, & \text{if } d \neq \emptyset. \end{cases}$$

The compositions defined above allow us to specify a rather expressive program language. This language generalizes several simple program algebras used in the approaches such as Glushkov Algorithmic Algebras [12], Floyd-Hoare logics [8,9], algorithmic logics [14], dynamic logics [15], etc. In Glushkov Algorithmic Algebras predicates and functions are considered as partial as we do here. Therefore we consider our approach as a generalization of Glushkov Algorithmic Algebras.

Definition 10. *An Associative Nominative Glushkov Algorithmic Algebra of predicates and functions over the nominative data of the type TND_{CC} is a two-sorted algebra*

$$NGA^a_{CC}(V, A) =< Pr_{CC}(V, A), Fn_{CC}(V, A); \vee, \neg, \bullet, IF, WH, \cdot, (Asg^u)_{u \in V^+},$$

$$(S^{\bar{u}}_F)_{\bar{u} \in \bar{U}}, (S^{\bar{u}}_P)_{\bar{u} \in \bar{U}}, Id, True, \perp_F, \perp_P, (u!)_{u \in V^+}, Empty, IsEmpty >$$

with the carrier sets $Pr_{CC}(V, A)$ (predicates) and $Fn_{CC}(V, A)$ (functions) for some V and A with the following operations: the disjunction $\vee$ and the negation $\neg$ compositions on predicates, the sequential composition of functions $\bullet$, the branching composition IF, the cycle composition WH, the prediction composition $\cdot$, the family of assignment compositions Asg^u, $u \in V^+$, the families of superposition compositions $S^{\bar{u}}_P$, $S^{\bar{u}}_P$ for $\bar{u} \in \bar{U}$, the identity composition on functions Id, the true constant predicate $True$, and the following constant elements of the carrier sets (null-ary compositions): the bottom predicate $\perp_P$, the bottom function $\perp_F$, the family of name checking predicates $u!$, $u \in V^+$, the empty constant function $Empty$, and the emptiness checking predicate $IsEmpty$.

Similarly, we can define algebras of functions and predicates over nominative data of the types TND_{CS} and TND_{SC}:

Definition 11.

(1) An Associative Nominative Glushkov Algorithmic Algebra of predicates and functions over the nominative data of the type TND_{SC} is a two-sorted algebra

$$NGA^a_{SC}(V,A) =< Pr_{SC}(V,A), Fn_{SC}(V,A); \vee, \neg, \bullet, IF, WH, \cdot, (Asg^u)_{u\in V^+},$$

$$(S^{\bar{u}}_F)_{\bar{u}\in\bar{U}}, (S^{\bar{u}}_P)_{\bar{u}\in\bar{U}}, Id, True, \bot_F, \bot_P, (u!)_{u\in V^+}, Empty, IsEmpty >$$

(2) An Associative Nominative Glushkov Algorithmic Algebra of predicates and functions over the nominative data of the type TND_{CS} is a two-sorted algebra

$$NGA^a_{CS}(V,A) =< Pr_{CS}(V,A), Fn_{CS}(V,A); \vee, \neg, \bullet, IF, WH, \cdot, (Asg^u)_{u\in V^+},$$

$$(S^{\bar{u}}_F)_{\bar{u}\in\bar{U}}, (S^{\bar{u}}_P)_{\bar{u}\in\bar{U}}, Id, True, \bot_F, \bot_P, (u!)_{u\in V^+}, Empty, IsEmpty >$$

10 Nominative Stability

The type TND_{CC} of nominative data is the most rich and interesting among other types of nominative data, so we will focus on it.

The binary relation of *nominative stability* is a formalization of the idea that a program's behavior does not change, if the hierarchical naming structure of its data changes. It can be illustrated by the following feature of the Pascal programming language: the two-dimensional array definitions `var A: array [1..n, 1..m] of real` and `var A: array [1..n] of array [1..m] of real` are equivalent and both the `A[i,j]` and `A[i][j]` syntax can be used to access the array elements regardless of the form of its definition. However, it should be noted that many practical programming languages like C++ and Java do not have this feature. This gives a rise to the following informal question: *which properties a programming language must satisfy in order to guarantee that its programs behave correctly regardless of the hierarchical naming structure of their data*?

We formalize such a notion of independence of a program behavior from the hierarchical naming structure of data as nominative stability of functions and predicates.

Definition 12. *A predicate $p \in Pr_{CC}(V,A)$ is called nominative stable, if for each $d_1, d_2 \in NDVC(V,A)$, if $p(d_1)\downarrow$ and $d_1 \approx d_2$, then $p(d_2)\downarrow$ and $p(d_1) = p(d_2)$.*

Definition 13. *A function $f \in Fn_{CC}(V,A)$ is called nominative stable, if for each $d_1, d_2 \in NDVC(V,A)$, if $f(d_1)\downarrow$ and $d_1 \approx d_2$, then $f(d_2)\downarrow$ and $f(d_1) \approx f(d_2)$.*

Let us denote by $PrNS(V, A)$ the set of all nominative stable predicates $p \in Pr_{CC}(V, A)$ and by $FnNS(V, A)$ the set of all nominative stable functions $f \in Fn_{CC}(V, A)$.

Theorem 3. *$PrNS(V, A)$ and $FnNS(V, A)$ form a sub-algebra of the Associative Nominative Glushkov Algorithmic Algebra $NGA^a_{CC}(V, A)$.*

Proof (Sketch). Using the same methods which were used in [5] to show monotonicity of program compositions it is straightforward to show that $FnNS(V, A)$ is closed under the sequential composition $\bullet$ and the identity composition Id, and if $p \in PrNS(V, A)$ and $f, g \in FnNS(V, A)$, then $f \cdot p \in PrNS(V, A)$, $IF(p, f, g)$ and $WH(p, f) \in FnNS(V, A)$. Also, it is trivial to show that $PrNS(V, A)$ is closed under the negation and disjunction compositions.

From [5, Lemmas 7.3 and 7.4] it follows that the assignment composition preserves nominative stability, i.e. if a function $f \in FnN(V, A)$ is nominative stable, then $Asg^u(f)$ is nominative stable, so if $f \in FnNS(V, A)$, then $Asg^u(f) \in FnNS(V, A)$.

Let us show that superposition into a function preserves nominative stability. Let $n \geq 1$, $f, f_1, ..., f_n \in FnN(V, A)$, and $(u_1, u_2, ..., u_n) \in \bar{U}$. Let us show that $S_F^{u_1,...,u_n}(f, f_1, ..., f_n) \in FnNS(V, A)$. Let $d_1, d_2 \in NDVC(V, W)$, $S_F^{u_1,...,u_n}(f, f_1, ..., f_n)(d_1) \downarrow$, and $d_1 \approx d_2$. Then $d_1 \notin A$ for all $i \in \{1, 2, ..., n\}$. Thus $d_2 \notin A$ and for all $i \in \{1, 2, ..., n\}$, $f_i(d_2) \approx f_i(d_1)$. Then

$$(...((d_2 \nabla_a^{u_1} f_1(d_2)) \nabla_a^{u_2} f_2(d_2))...\nabla_a^{u_n} f_n(d_2))...) \downarrow$$

and

$$(...((d_1 \nabla_a^{u_1} f_1(d_1)) \nabla_a^{u_2} f_2(d_1))...\nabla_a^{u_n} f_n(d_1))...) \approx$$
$$\approx (...((d_2 \nabla_a^{u_1} f_1(d_2)) \nabla_a^{u_2} f_2(d_2))...\nabla_a^{u_n} f_n(d_2))...),$$

because $\approx$ is a congruence in $NDA_{CC}(V, A)$. Taking into account that $f \in FnNS(V, A)$, we conclude that

$$S_F^{u_1,u_2,...,u_n}(f, f_1, ..., f_n)(d_2) \downarrow$$

and

$$S_F^{u_1,u_2,...,u_n}(f, f_1, ..., f_n)(d_2) \approx S_F^{u_1,u_2,...,u_n}(f, f_1, ..., f_n)(d_1).$$

Thus $S_F^{u_1,u_2,...,u_n}(f, f_1, ..., f_n) \in FnNS(V, A)$.

Similarly it is straightforward to show that superposition into a predicate preserves nominative stability.

Besides, it is easy to check that

$$\{True, \perp_P, IsEmpty\} \cup \{u! \mid u \in V^+\} \subset PrNS(V, A)$$

and

$$\{\perp_F, Empty\} \subset FnNS(V, A).$$

We conclude that $PrNS(V, A)$ and $FnNS(V, A)$ are closed under all compositions of $NGA^a_{CC}(V, A)$, so they form a sub-algebra. $\square$

Let us define the following equivalence relation on $FnNS(V, A)$:

$$f \equiv_F g$$

(where $f, g \in FnNS(V, A)$), if $d_1 \approx d_2$ and $f(d_1) \downarrow$ implies $f(d_1) \approx g(d_2)$.

Also, let us define the following equivalence relation on $PrNS(V, A)$:

$$p_1 \equiv_P p_2$$

(where $p_1, p_2 \in PrNS(V, A)$), if $d_1 \approx d_2$ and $p(d_1) \downarrow$ implies $p_1(d_1) = p_2(d_2)$.

Let

$$NGANS^a(V, A) =$$

$$< PrNS(V, A), FnNS(V, A); \vee, \neg, \bullet, IF, WH, \cdot, (Asg^u)_{u \in V^+},$$

$$(S_F^{\bar{u}})_{\bar{u} \in \bar{U}}, (S_P^{\bar{u}})_{\bar{u} \in \bar{U}}, Id, True, \bot_F, \bot_P, (u!)_{u \in V^+}, Empty, IsEmpty >$$

be the sub-algebra mentioned in Theorem 3.

Theorem 4. *The relations $\equiv_F$, $\equiv_P$ are congruences on $NGANS^a(V, A)$.*

We omit the proof, as it can be easily obtained from Theorem 1.

This theorem allows us to construct a factor algebra.

Let $NGANS^a_{\equiv}(V, A)$ be the factor algebra of $NGANS^a(V, A)$ by the congruences $\equiv_F$ and $\equiv_P$.

Theorem 5. *The algebras $NGA^a_{CS}(V, A)$, $NGA^a_{SC}(V, A)$, and $NGANS^a_{\equiv}(V, A)$ are isomorphic.*

Proof (Sketch). Using Lemmas 1 and 2 it is straightforward to show that the algebra $NGANS^a_{\equiv}(V, A)$ is isomorphic to $NGA^a_{CS}(V, A)$ and $NGA^a_{SC}(V, A)$. Thus $NGA^a_{CS}(V, A)$ and $NGA^a_{SC}(V, A)$ are isomorphic. □

The obtained result has several interpretations and possible applications. Firstly, a programmer can construct a nominative stable program oriented on a certain hierarchical naming structure of input data, but this program would give equivalent results, if input data were changed to equivalent data. Such stability simplifies programming with complex data making it "softer" because the programmer should not remember the current structure of data. Secondly, the class of such programs can be correctly implemented using nominative data from the class TND_{SC} (or TND_{CS}) only. Third, the obtained result gives a perspective for reduction of formulas of a logic over hierarchical nominative data to formulas of a logic over "flat" data, which is closer to classical logic.

11 Conclusions and Future Work

We have investigated basic properties of nominative functions and predicates from the perspective of the abstract algebra. We have defined basic compositions of such functions and predicates and introduced a two-sorted algebra which generalizes Glushkov algorithmic algebras. An advantage of this generalization is that its functions are defined over nominative data which give a natural and adequate representation of data structures commonly used in programming (as we have demonstrated in Sect. 6). We have considered an equivalence relation on nominative data with complex names and the notions of nominative stability of nominative functions and predicates and proved the set of a nominative stable functions and the set of nominative stable predicates form its sub-algebra.

In the forthcoming papers we plan to continue our study of theoretical aspects of composition-nominative approach and describe practical applications of this approach.

References

1. Sannella, D., Tarlecki, A.: Foundations of Algebraic Specification and Formal Software Development. Springer, Heidelberg (2012)
2. Nikitchenko, N.S.: A composition-nominative approach to program semantics. Technical report, IT-TR 1998–020, Technical University of Denmark (1998)
3. Nikitchenko, M., Tymofieiev, V.: Satisfiability in composition-nominative logics. Cent. Eur. J. Comput. Sci. **2**, 194–213 (2012)
4. Kryvolap, A., Nikitchenko, M., Schreiner, W.: Extending Floyd-Hoare logic for partial pre- and postconditions. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2013. CCIS, vol. 412, pp. 355–378. Springer, Heidelberg (2013)
5. Nikitchenko, M., Ivanov, I.: Programming with nominative data. In: Proceedings of CSE'2010 International Scientific Conference on Computer Science and Engineering, Kosice, Slovakia, 20–22 September 2010, pp. 30–39 (2010)
6. Nikitchenko, M., Ivanov, I.: Composition-nominative languages of programs with associative denaming. Visnyk (Bulletin) of the Lviv University Ser. Appl. Math. Inform. **16**, 124–139 (2010)
7. Nielson, H.R., Nielson, F.: Semantics with Applications: A Formal Introduction. John Wiley & Sons Inc., New York (1992)
8. Floyd, R.: Assigning meanings to programs. Proc. Am. Math. Soc. Symp. Appl. Math. **19**, 19–31 (1967)
9. Hoare, C.: An axiomatic basis for computer programming. Commun. ACM **576–580**, 583 (1969)
10. Skobelev, V.: Local algorithms on graphs. Publishing house of Institute of Applied Mathematics and Mechanics of NAS of Ukraine (in Russian) (2003)
11. Roman, S.: Lattices and Ordered Sets. Springer, New York (2008)
12. Glushkov, V.: Automata theory and formal transformations of microprograms. Cybernetics (in Russian) **5**, 3–10 (1965)
13. Coquand, T., Paulin, C.: Inductively defined types. In: Martin-Löf, P., Mints, G. (eds.) COLOG 1988. LNCS, vol. 417, pp. 50–66. Springer, Heidelberg (1990)
14. Mirkowska, G., Salwicki, A.: Algorithmic Logic. Springer, New York (1987)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. The MIT Press, Cambridge (2000)

TuningGenie: Auto-Tuning Framework Based on Rewriting Rules

Pavlo A. Ivanenko(✉), Anatoliy Y. Doroshenko, and Kostiantyn A. Zhereb

Institute of Software Systems of National Academy of Sciences of Ukraine, Glushkov prosp. 40, Kyiv 03187, Ukraine
paiv@ukr.net, {doroshenkoanatoliy2, zhereb}@gmail.com

Abstract. This paper presents results on development of the auto-tuning framework named TuningGenie aimed at automating adjustment of parallel tasks to target platform. The framework works with source code of parallel software and performs source-to-source transformations by using facilities of a rule-based rewriting system. This approach significantly raises flexibility of proposed solution and enables changing computation structure of a tuned program in a declarative way. The framework's architecture, lifecycle, offered toolset for optimization, demo examples and results of tuning of a computationally complex task of short-term meteorological forecasting are presented.

Keywords: Auto-tuning · Rewriting rule systems · Parallel software optimization

1 Introduction

Current and emerging scientific and industrial applications require significant computation power provided by parallel platforms such as multicore, clusters, Grids, cloud computing and GPGPU. Therefore, optimizing application code for a given parallel platform is a significant part of development process. The problem is that powerful computation platform does not guarantee high performance and efficiency of application: developer efforts are required to achieve it.

Programming efficient algorithm has always been a challenging task. It is made even harder by widespread use of multicore processors. The performance growth of a multithreaded program is limited by Amdahl's law [1], so efficient programs should minimize the fraction of sequential computation as well as synchronization and communication overhead. Such optimizations are usually hardware platform-dependent, and the program that was optimized for one platform is likely not optimal on another platform. Therefore there is a need of tools that could perform such adaptation to concrete platform in automated mode.

One traditional solution of this problem is provided by parallelizing compilers [29, 32]. Among existing well known implementations of this methodology it's worth mentioning Intel® Parallel Compilers [2, 30] and Oracle® Solaris Studio [3]. The quality of automated parallelization has increased recently with OpenMP [31] support, however such tools are still most efficient when applied to programs with simple

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 139–158, 2014.
DOI: 10.1007/978-3-319-13206-8_7

structure of computation (loops without data dependency between iterations) [2, 35]. The reasons include the complexity of static analysis of large codebases, increasing complexity of parallel architectures, as well as their diversity.

An alternative to parallelizing compilers is provided by auto-tuning [4] – a methodology that has already proven its efficiency and versatility. It can automate the search for the optimal program version out of set of provided possibilities by running each candidate and measuring its performance on a given parallel architecture. Its main benefit is high level of abstraction – program is optimized without explicit knowledge of hardware implementation details, such as number of cores, cache size or memory access speed on various levels. Instead, parallel programs use subject domain concepts such as number and size of independent tasks, or algorithm details such as data traversal methods.

The drawbacks of auto-tuning approach include significant one-time costs of optimization process itself: if the number of program versions is large enough, the optimization process may run for many hours and even days. Also embedding tuning facilities into program's logic makes its architecture more complex. In most cases it is reasonable to extract tuning functionality into a separate application and leave optimized software unaware about optimization. In any case, there is an additional development overhead – creation of auto-tuner. Since auto-tuner generally does not depend on software domain and performs generic actions it is possible to generate it automatically and save developer from writing large amount of boilerplate code.

This paper considers a universal framework for automated generation of auto-tuner applications from source code. In general proposed solution is independent from programming language and applied frameworks. Proposed approach solves many of the described drawbacks and is versatile enough to be applied in any subject domain.

2 Rewriting Rule-Based Approach

TuningGenie framework optimizes parallel programs by generating a stand-alone tuner application. This approach allows encapsulating the optimization logic in auto-tuner, simplifying development and maintenance for the target application.

As mentioned before, manually creating auto-tuner requires writing significant amount of boilerplate code and modifying program architecture to provide communication with auto-tuner. To avoid developing a separate tuning application from scratch the TuningGenie framework extracts all needed information from program source code and generates auto-tuner automatically. Developers add some metadata to the source code in form of special comments – pragmas. Such expert knowledge defines number of program variants to be evaluated, which is expected to be as small as possible. Also such approach allows TuningGenie to provide domain-independent optimization.

The software implementation of the auto-tuning system is based on the rewriting rules system TermWare, [5, 6]. TermWare is an open-source implementation of rewriting rules engine written in Java. It provides a language for describing *rewriting rules* that operate on data structures that are called "*terms*", and also a rule engine that interprets rules to transform terms. Informally, terms are tree-like structures of the form $f(t_1, t_2, \ldots, t_n)$, where *subterms* $t_1, t_2, \ldots, t_n$ are either tree nodes themselves or leaf nodes, corresponding to constants or variables. In this section, we provide a brief formal description of syntax and semantics of TermWare language, sufficient to understand the next sections. More detailed description can be found in [5, 6].

TermWare syntax is described by the following grammar rules

$$\begin{aligned}
&Term ::= Const\,|Var\,|\,Identifier\,(Term\,[,Term]^{*})\\
&Const ::= String\,|\,Number\,|\,true\,|false\,|\,nil\\
&Var ::= \$Identifier
\end{aligned}$$

Here *String*, *Number* and *Identifier* have their usual meaning, e.g. like in Java language. In this section, we denote constants as c_i, variables as $\$x_i$ and arbitrary terms as t_i. For a term $t = f(t_1, t_2, \ldots, t_n)$, we define

$$\begin{aligned}
head(t) &= f\\
arity(t) &= n
\end{aligned}$$

For constants and variables, we consider

$$\begin{aligned}
head(c) &= head(\$x) = nil\\
arity(c) &= arity(\$x) = -\infty
\end{aligned}$$

Some terms can also be entered using shortcut infix notation, such as

$$\begin{aligned}
&a + b = plus(a, b)\\
&pattern[cond] \rightarrow target[act] = rule(pattern, cond, target, act)
\end{aligned}$$

In particular, *rule* terms are very important – they represent a “program” written in TermWare, i.e. a transformation of terms.

To describe the semantics of TermWare rules, we use the following additional concepts and operations. A *variable substitution* is a mapping from some set of variables to terms:

$$s = \{\$x_1 \rightarrow t_1, \ldots, \$x_n \rightarrow t_n\}$$

Match operation is used to match arbitrary term with a given pattern. This operation takes as inputs a term that does not contain variables, a pattern (a term that can contain variables) and current substitution, and outputs true/false value that denotes match success/failure, and in case of success, an updated substitution. Match is performed according to the following rules:

1. $match(c, c, s) = (true, s)$
2. $t \neq c \Rightarrow match(t, c, s) = (false, \emptyset)$
3. $x \notin s \Rightarrow match(t, x, s) = (true, s \cup \{x \rightarrow t\})$
4. $s(x) = t \Rightarrow match(t, x, s) = (true, s)$
5. $s(x) \neq t \Rightarrow match(t, x, s) = (false, \emptyset)$
6. $head(t_1) \neq head(t_2) \Rightarrow match(t_1, t_2, s) = (false, \emptyset)$
7. $arity(t_1) \neq arity(t_2) \Rightarrow match(t_1, t_2, s) = (false, \emptyset)$
8. $match(t_i, p_i, s_{i-1}) = (v_i, s_i) \Rightarrow match(f(t_i), f(p_i), s_0) = \left(\bigcup_{i=1}^{n} v_i, s_n \right)$

Rules 1–2 mean that constant patterns match only themselves. Rules 3–5 describe matching of variables: if variable is missing from current substitution, it captures the value of any term; if variable is in substitution, it will only match the same term as it has captured. Rules 6–7 specify that terms with different heads or arities cannot match. Finally, rule 8 defines recursive matching of terms with same heads and arities, in which case subterms are matched sequentially.

Subst operation applies a given variable substitution to a given target. It is defined by the following rules:

1. $subst(c, s) = c$
2. $subst(x, s) = s(x)$
3. $subst(t_i, s) = r_i \Rightarrow subst(f(t_i), s) = f(r_i)$

Again we consider cases of constants (rule 1) that ignore variable substitution and return themselves, variables (rule 2) that are replaced with a term from variable substitution, and tree nodes (rule 3) that are replaced recursively.

Now we can define how TermWare rule $pattern[cond] \rightarrow target[act]$ is applied to a given term t:

1. Enumerate all subterms of t, starting with t itself and using depth-first search.
2. For each subterm t_i, calculate $match(t_i, pattern, \emptyset) = (v, s)$
3. If $= false$, rule application has failed, move to next subterm.
4. If $v = true$, evaluate conditions represented by *cond*. Technically, we calculate $subst(cond, s)$ to obtain conditions with concrete values of variables, then evaluate this expression as an imperative Java code. If evaluation results in false value, rule application has failed
5. Calculate $subst(target, s) = r$ and replace subterm t_i with the term r
6. Evaluate actions represented by *act*, in the same way as in step 4. Now we are not interested in evaluation result: actions are supposed to execute imperative code that has some useful side-effects (such as saving results to a database, or recording messages into a log).

Steps 4 and 6 are optional: *cond* and *act* components of a rule can be omitted, in this case corresponding step is skipped. Using these components allows TermWare rules to take full advantage of arbitrary Java facilities. In particular, certain details can be saved when applying one rule and then used in the next rule, even if it operates with unrelated part of term.

TermWare can work with arbitrary data encoded as terms. In this paper, we represent Java source code in form of abstract syntax trees (AST) as a term that is transformed with TermWare rules. Notice that in this case, there is a a difference between TermWare constants (like 5) and Java constants (represented as Literal(5)), and also between TermWare variables (like $x) and Java variables (represented as Identifier("x")). However, it is usually clear from context which case is assumed.

TuningGenie uses TermWare to extract expert knowledge from program source code and generate a new version of program on each tuning iteration and knowledge base is used as interlink between pre-tuning and tuning phases. By manipulating terms representing AST, TuningGenie can perform structural changes in program computations using declarative style (without explicit changes to source code).

Current version of TermWare contains components for interaction with Java and C# languages. To support other languages, a parser to translate source code into TermWare language and a pretty-printer should be implemented. Adding support of a new language to TuningGenie also requires extending it by means for building and launching optimized application. API for measuring time and searching for optimal configuration among evaluated is generic and written in Java. So in general TuningGenie's architecture supports any imperative programming language but current version works only with Java programs.

Simple examples of using rewriting rules for different tasks can be found at [5].

As already mentioned, the main benefit of TuningGenie is gained from utilization of term rewriting engine facilities. Representing program code as a term allows modifying program structure in a declarative way without modifications in source code. This feature becomes very useful when you are working with "legacy" code and want to try out slightly different variants without refactoring it. An example of such transformation is displayed below in Sect. 5 "TuningGenie facilities".

3 A Model for Parallel Programs Optimization

Consider the following simple abstract model that illustrates the optimization approaches for parallel programs. Our model is based on a well-known abstract parallel machine with random memory access (PRAM [7]). We add the following changes to bring the model closer to modern parallel systems:

- Besides global memory, we consider a local memory which is another level of memory hierarchy. This local memory represents CPU caches of various levels (shared between CPU cores), local application caches, etc. The access speed for the local memory is significantly faster compared with the global memory: $T_{loc} \ll T_{shar}$. Another significant difference is limited size of the local memory, denoted as M_{loc}. If the data is missing from the local memory (e.g. because of its size which is too large for the local memory), it is read from slower global memory. In real computing systems there are multiple levels of memory, but for our purposes a single additional level will be sufficient.
- We consider our system as containing homogeneous processors. As modern CPUs are usually multicore, our model "processor" corresponds to a single core of

real CPU. The number of processors is considered fixed and denoted by *N*. Notice that we apply auto-tuning approach to adapt application to a concrete computing system with *fixed* parameters.
- Concurrent access to the shared memory in our model is limited to a single strategy Concurrent Read Exclusive Write (CREW). Therefore, while all read operations are concurrent, write operations are exclusive. Denote $T_{N_{ew}}$ a time that processors wait for write access to the shared memory.

Parallel programs usually contain critical sections, i.e. regions of code that cannot be executed in parallel. When such section is executed by a single processor, other processors cannot execute the same section, therefore the computations inside a critical section can be modeled by a sequential RAM model [7]. Also additional time is needed to synchronize the results of critical sections execution on different processors. Denote $T_{N_{seq}}$ execution time of all critical sections. The time of parallel execution (i.e. non-critical sections of code) is denoted as $T_{N_{par}}$.

In our model each instruction is executed in three phases:

1. Read the data from global memory to local memory – if the data is not already present in local memory, and if it is needed to for instruction execution.
2. Execute the instruction using the data from the local memory.
3. Write instruction results to the global memory (optional)

We separate the time spent on memory access by a processor with number *i*. This time includes first and third phases of instruction execution: $T_{i_{mem}} = T_{i_{loc}} + T_{i_{shar}}$.

Let T_N be a total time of parallel program execution. It consists of the following parts:

$$T_N = T_{N_{par}} + T_{N_{seq}} + T_{N_{ew}}$$

Therefore, optimizing a parallel algorithm can be aimed at one of three targets:

- reducing time of parallel execution $T_{N_{par}}$ (without increasing $T_{N_{seq}}$).
- reducing synchronization overhead $T_{N_{ew}}$.
- reducing sizes of critical sections and therefore the time of sequential execution $T_{N_{seq}}$

Parallel execution time can be reduced by optimizing the usage of the local memory. To this end, we should partition computation between the processors in such a way that the relevant data fits into the local memory with significantly faster access compared to global memory. In ideal case, for each processor $i: T_{i_{shar}} \to 0$. Our framework facilitates the design of the algorithms that use cache efficiently, and does not require an explicit knowledge about hardware architecture.

As an example, consider a modification of QuickSort algorithm [8]. The modified algorithm contains a parameter responsible for switching to another algorithm for "inner" sorting. This parameter should be selected based on cache efficiency concerns. In our approach, the optimal value of this parameter is selected empirically by the auto-tuner. This example is considered in detail in Sect. 5 «TuningGenie facilities». For now we note that such algorithms belong to the area of Cache-aware

Algorithms [9]. On the other hand, the algorithm is abstract enough: it does not depend on hardware architecture of execution environment, as this parameter is a part of subject domain and not a parameter of cache implementation.

The synchronization overhead $T_{N_{ew}}$ can be reduced by optimal partitioning of computation tasks between processors. As an example, consider a classical coarse-grained block parallelization scheme, which provides uniform processor load using as large blocks of data as possible. In such case, the number of data exchanges between processors and the synchronization overhead is minimal. Coarse-grained block parallelization scheme provides the optimal performance for many parallel problems [10]. Another approach considers structural modification of computations performed by a program. In Sect. 7 “Asynchronous cycles for iterative algorithms” we present an example of such transformation for a class of iterative algorithms using barrier synchronization at the end of each iteration.

Unfortunately, the approaches that target (1) optimizing parallel section of computation and (2) reducing synchronization overhead, are usually at conflict: they require different strategies for distributing tasks between processors. The most efficient strategy depends both on a given computation problem and on parameters of hardware platform, such as shared memory access speed, cache access speed and cache size. The optimal strategy must be defined for each execution environment. The power of auto-tuning methods is that the selection of strategy is performed empirically and in automated mode.

4 Tuning Lifecycle

General lifecycle of auto-tuning process consist of three steps:

1. Apply modification that might affect software performance
2. Estimate actual performance empirically (in target environment)
3. Analyze estimation results. If efficiency criteria are met – optimization is finished (exit). Go to step 1 otherwise.

The process is visualized in Fig. 1:

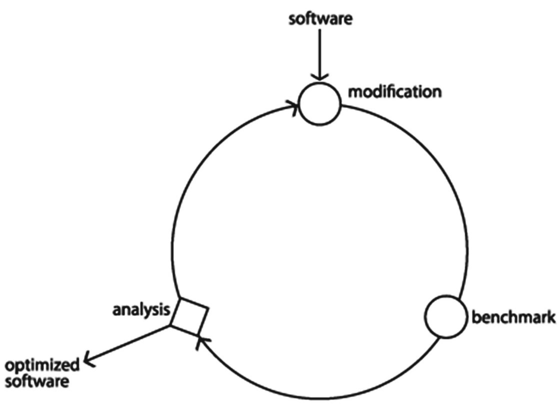


Fig. 1. Generic tuning workflow

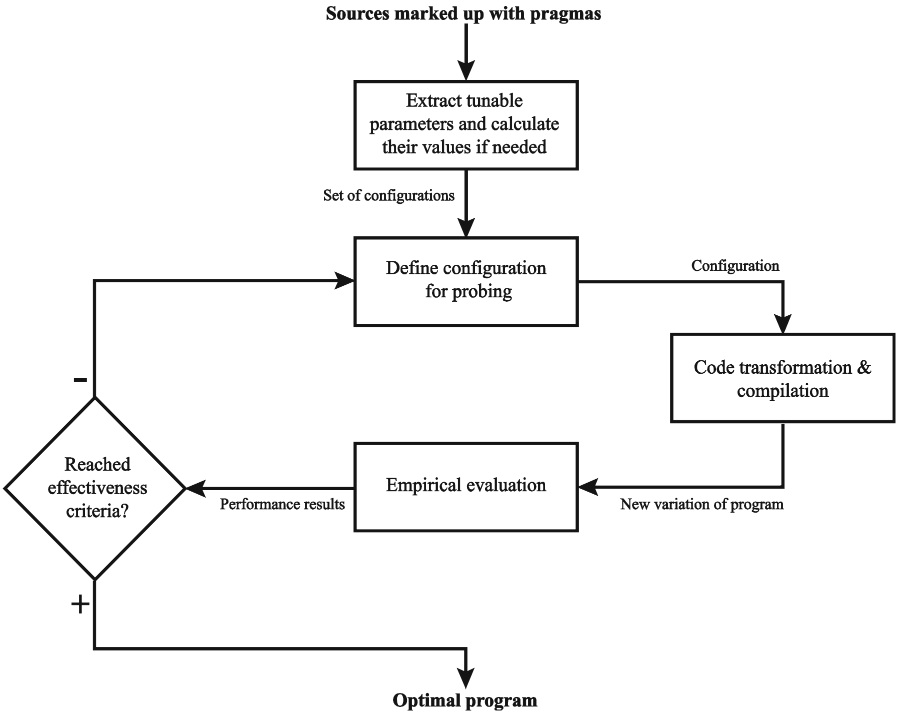


Fig. 2. TuningGenie workflow

Such lifecycle is the same for all *static* auto-tuners – tuners that do not affect runtime execution of software. Optimization is performed via iterative evaluation of software versions until a version with acceptable performance is found, all possible variants are tried or time limit is reached.

TuningGenie implements the static auto-tuner paradigm and fits the described lifecycle. The following section describes how TuningGenie modifies, evaluates and chooses optimal variant of program in detail (graphically presented in Fig. 2).

TuningGenie accepts as an input Java source code marked with so-called "pragmas". Pragmas describe configurations and transformations of program that affect its performance. Pragmas are specified manually by program developers, using Java comments of a special form.

During the parsing of source code into a term representing AST, the auto-tuner builds a set of configurations based on these expert data. Then these configurations are translated into rewriting rules. Also on this "preliminary" stage some values of program parameters are calculated (see the "*calculatedValue*" pragma in Sect. 5). The results of this stage include a program term, a set of parameterized rewriting rules and a set of rule configurations C that specify the concrete parameter values. Each of these configurations specifies a unique version of the input program.

Then TuningGenie searches for the most efficient configuration $C_{opt} \in C$, by iteratively performing the following steps:

1. Select a configuration to test $C^* \in C$. Configurations are selected sequentially from set C. Size of C depends on values from pragmas and developers are expected to use their expert knowledge to narrow the region of search (group pragmas only if they are really related, narrow boundary values for each pragma). Additionally time limit can be used for cases when it's hard to estimate overall optimization time (step 4).
2. Generate and compile the corresponding program variant. The parameters from a given configuration are substituted into the rules; then these rules are executed on the source term of the parallel program. After the transformation is complete, the term is transformed into Java source code using TermWare facilities. The code is compiled using common JDK tools. The result of this step is a new version of the program ready for performance measurement.
3. Execute the program and evaluate its performance. A small launcher class is generated automatically. It runs the program and measures its execution time, therefore computing $f(C^*) = t$. Each configuration is executed several times (at least 3). If the time consumed for each run differs a lot (more than 10 %) – warning message is logged.
4. If all configurations from the set C have been evaluated, or if the optimization process has used up all the allotted time – go to step 5. Else go to step 1.
5. The optimal configuration is selected from a set of all configuration that have been evaluated: $C_{opt} = C^*{:}f(C^*) = f_{min}$. For this configuration, an optimal program version is generated, as in step 2.

The program obtained as a result of steps 1–5 is saved and executed in a target environment. This program is considered optimal for a given architecture.

5 TuningGenie Facilities

As already mentioned in Sect. 3, the expert knowledge about subject domain and implementation is saved in source code as special directives called "pragmas" (by analogy with C language). Pragmas are actually the comments of a special form. Therefore they are ignored by the Java compiler. Adding pragmas does not change the structure of computation, and such instrumented program can be compiled by any compiler without additional libraries.

Currently the TuningGenie framework supports three kinds of pragmas:

- *tunableParam* – specifies a search domain for an optimal value of a numeric variable. E.g. the following pragma sets the possible values for a `threadCount` variable as a range [50..100] with a step 10:

```
//tunableParam name=threadCount start=50 stop=100 step=10
int threadCount = 1;
```

The *tunableParam* pragma is applicable to the algorithms that use geometrical (data) parallelization: it allows to find the optimal decomposition of computation by estimating the size of a block that executes on a single processor. It also can be applied when we need to estimate the optimal number of some limited resources like size of caches or number of used threads to be used in a program. Another use of the *tunable-Param* pragma is to find an optimal threshold value to switch to a different algorithm. Consider a sort algorithm that uses QuickSort or MergeSort for large arrays and applies the same sort recursively to sub-arrays. For some small size of sub-array, it is more efficient to switch to another sorting algorithm, such as InsertionSort. The optimal threshold value depends on processor architecture and can be determined experimentally. TuningGenie framework allows finding this value with minimal changes to source code:

```
//tunableParam name=threshold start=10 stop=100 step=5
int threshold = 1;
if (high - low < threshold) {
    insertionsort(array, low, high);
} else {
    addPartitionForQuickSort(array, low, high)
}
```

- *calculatedValue* pragma evaluates some function in a target environment and assigns the resulting value to a given variable:

```
//calculatedValue name=hdReadSpeed method=
//"org.tuning.EnvironmUtils.getHDReadSpeed()"
int hdKbPerSec= 1;
```

This pragma can be used if some algorithm parameters depend on execution environment details, such as execution times of arithmetic operations or memory access on various levels. All such values are calculated before auto-tuning process starts and saved in TermWare knowledge base. The data stored in knowledge base is accessible from rewriting rules, so calculatedValue can specify the value range for the tunable-Param pragma. In this way, the developer can run a small test program to reduce the search domain and improve auto-tuning speed. The similar approach for GPU computing is described in [11].

- *bidirectionalCycle* – can be used to specify loops were the iterations can be run in any order. In the following example TuningGenie will evaluate both increment (`0` to `SIZE-1`) and decrement (`SIZE-1` to `0`) versions:

```
int[] data = new int[SIZE];
//bidirectionalCycle
for (int i=0; i <SIZE; i++) {
 doSomethingWith(data[i]);
}
```

This pragma is targeted at cases when multi-dimensional data is linearized into single-dimensional. Sequential iteration from multi-dimensional point of view is not always sequential for linearized data, hence not always cache-efficient. So if order of iteration is not important tuner might pick up better variant. In our case study (Sect. 5), for a meteorological program the difference in running time due to this change was about 15 %.

It's worth mentioning that by default all pragmas (and corresponding parameters) are treated as independent. It means that for each parameter's value from a single pragma a separate version of program is generated and measured; parameters from other pragmas are ignored (i.e. take values specified in source code). So if we have n pragmas, with number of configurations contributed by each pragma denoted as N_i, then total size of search space will be $\sum N_i$ and not $\prod N_i$. To mark parameters that need to be tested in correlation optional *group* option must be specified in related pragmas.

Other important fact is that at the moment all pragma's variables have "local" scope and they are tied to demarked instruction. That's why it's impossible to gather all pragmas in one place – they have to be spread among source code and duplicated for each section of duplicated code (if such is present).

6 Translating Pragmas to Rewriting Rules

As it is already discussed in previous sections we use the rewriting rules technique to transform source code of the program. For each pragma described in the previous section a corresponding rewriting rule is generated. Then all generated rules are applied to the term representing the source code of the program. In this section, we describe in detail the process of rule generation for each pragma.

For *calculatedValue* and *tunableParam* pragmas, the rewriting rules are completely identical. Both pragmas initialize some variable with a value taken from the currently tested configuration. The only difference is in syntax of pragmas: for *calculatedValue* the value to be substituted is calculated before performing tuning iteration, and for *tunableParam* it is specified in parameters.

Consider the following code fragment containing tunableParam pragma

```
//tunableParam name=threshold start=1 stop=10 step=1
int threshold =1;
```

The rule generated from this pragma (slightly simplified for readability) has the following form:

```
VariableDeclarator(Id("threshold"), Literal($value))
[$value!="newValue"]        ->
VariableDeclarator(Id("threshold"), Literal("newValue"))
```

Notice the condition [$value!="newValue"] that was added to prevent infinite rule application. Rewriting rules are applied repeatedly until the moment when none of them is applicable. In this example, we need to prevent the infinite rule application in case when the substitution value newValue is equal to the initial $value. Actual value of "newValue" is taken from defined by pragma range of values. In this particular case newValue $\in [1, 2, 3 \ldots 10]$ and each of values would be automatically sequentially probed and estimated by tuner.

Now consider bidirectionalCycle pragma. Assume that the following loop does not depend on iteration order (i.e. it can be run both in increment and decrement order):

```
//bidirectionalCycle
for (int loopVar = 0; loopVar <42; loopVar ++){
.....................
}
```

The pragma is translated into a rewriting rule that uses propositional variables [12] to invert iteration order:

```
ForStatement(
    LoopHead(
          ForInit(VariableDeclarator(Id("loopVar"),Lite
    ral($initialValue))),
          RelationalExpression(Identifier("loopVar"),"<",
       Literal($endValue)),
          ForUpdate(StatementExpression(Identifier
   ("loopVar "),"++"))),
    Block(.....)
)       ->
ForStatement(
    LoopHead(
          ForInit(VariableDeclarator(Id("loopVar"),Lite
    ral($endValue))),
          RelationalExpressision(Identifier("loopVar"),">",
    Literal($initialValue)),
          ForUpdate(StatementExpression(Identifier
   ("loopVar "),"--"))),
    Block(.....)
)
```

This rule swaps initial and final values of loop counter variable, changes the operator that modifies this variable on each iteration (from increment to decrement), and changes the loop exit condition.

All rewriting rules, together with the strategy for their application and the knowledge base (facts base) [12], are combined into a single term system. The system performs reduction of the initial term and obtains a new term. This term is translated back into a program source code using target language (in our case it’s Java).

7 Asynchronous Cycles for Iterative Algorithms

There is a wide class of parallel programs that use iterative computation, where the outputs of a previous iteration become the inputs for the next one. Inside iterations the computations can be parallelized, e.g. using geometrical decomposition [10]. Figure 3 shows the computations performed within a single iteration:

In Fig. 3 t_{split}, $t_{parallel}$ and t_{merge} denote the time spent on distributing data, parallel computations and collecting results respectively. In such computation scheme we cannot change iteration order or run them in parallel. Now consider iteration concatenation from Fig. 4:

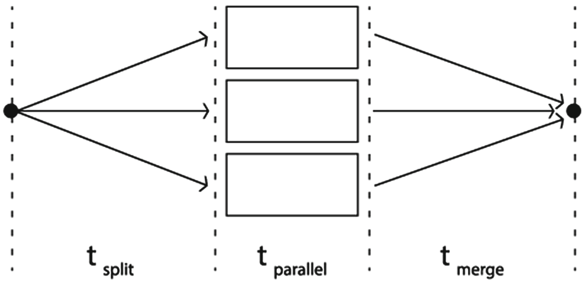


Fig. 3. Iteration phases

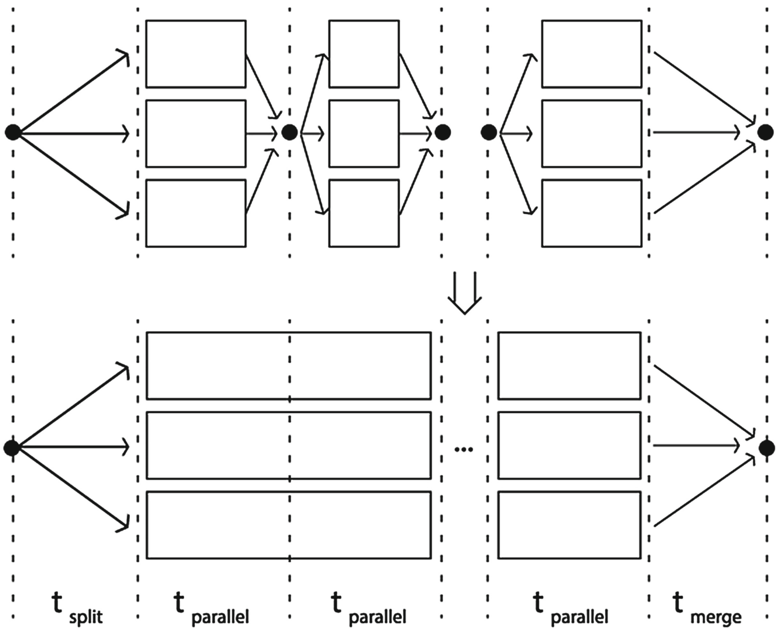


Fig. 4. Iterations transformation scheme

The number of merged iterations is denoted as m. A class of problems that allow such concatenation will be called problems of asynchronous cycle with degree of freedom m. Notice that such modification in computation structure can lead to slight changes in calculation result. It is out of scope to explain details of how such inaccuracy can be estimated and analyzed but interested reader can find details in papers [13, 14]. Here we only state that the main idea behind such transformation is sacrificing accuracy of result to gain additional performance boost. And the mission of tuner is to find acceptable trade-off.

It's easy to see that the performance of the parallel algorithm is improved by modifying computation scheme to use asynchronous cycle. This effect is achieved because of reduced overhead on data decomposition and synchronization of intermediate iteration results.

The total execution time of initial parallel program can be described by the following expression:

$$T_p = n \times (t_{split} + t_{exchange}),$$

where n is total number of iterations; $t_{exchange} = t_{split} + t_{merge}$.

For simplicity, let's assume that m divides n and $\in [1, n]$. Let's estimate the program execution time after transforming program to use the asynchronous cycle:

$$T_{pm} = n \times t_{parallel} + \frac{n}{m} \times t_{echange}$$

Now estimate the bounds on algorithm speedup S^{tuned}:

$$S_{tuned} = \frac{T_p}{T_{pm}} = \frac{t_{parallel} + t_{echang\mathit{e}}}{\frac{1}{m} \times t_{echange} + t_{parallel}}$$

$$\lim_{m=n\to\infty} S_{tuned} = \frac{t_{parallel} + t_{exchange}}{t_{parallel}} = 1 + \frac{t_{exchange}}{t_{parallel}}$$

The minimum value of this expression is achieved when $m = 1$, meaning that computation scheme coincides completely with initial one:

$$S_{tuned} = \frac{t_{parallel} + t_{echange}}{t_{parallel} + t_{echange}} = 1$$

Therefore speedup lies in the following bounds:

$$S_{tuned} \in [1, 1 + \frac{t_{exchange}}{t_{parallel}}]$$

Obviously this optimization approach is the most efficient for the problems that have a large number of iterations and also time of parallel computations and of data exchange is of the same order of magnitude. As said before, for most problems the speedup S_{tuned} increases with the increase of m along with increase of inaccuracy of result. If there is no need to preserve the intermediate results (e.g. for visualization purposes), the optimal value of m is determined by the accuracy requirements. The next section «Case Study» considers in detail an example of using asynchronous cycle approach in a complex computation problem.

8 Case Study

This section presents the case study on the performance tuning of an application for short-term atmospheric circulation modelling (from hours to several days).

Mathematical definition of the model and method for numerical solution can be found in [13]. Here we use a simplified problem definition to give a general idea.

Considered model describes macro-scale processes. Their usual horizontal extent is of the order of thousands of kilometers. Typical examples of such processes are equatorial and monsoonal currents, long Rossby waves [15], cyclones and anticyclones, ridges etc.

Model is based on spherical coordinate system. This work considers two-dimensional simplified case that models wind characteristics (direction and force). Hence model consists of two evolution equations to calculate horizontal components of wind force.

Model takes state of atmosphere $State(T_0)$ at some moment T_0 as an input. Suppose we want to get prediction of state at some moment in future - $State(T_m), T_m > T_0$. To calculate this program performs series of smaller predictions with time step $\Delta t \ll T_m - T_0$. So we have recurrence relation between iterations: $State(T_m) = Prediction(State(T_{m-1})), T_m = T_{m-1} + \Delta t$. Note that iteration step Δt should be chosen from the interval $[5, 20]$ minutes to provide acceptable accuracy of the result.

For the experiment initial conditions for model were taken from an electronic archive of one of European regional forecasting center [16].

Method for numerical solution of described model allows to parallelize computation on three levels:

- evolution equation level – each of equation can be solved independently in scope of one iteration;
- geometric decomposition – input data can be decomposed according to data parallel paradigm;
- by utilizing a modified additive-averaged operator splitting algorithm [14] it is possible to expand independent processing of subtasks to several calculation iterations without significant loss in accuracy of result.

Overall performance of parallel program mostly depends on granularity of data decomposition since it defines amount of calculation per subtask hence size of plain parallel section. It's reasonable to assume that "coarse-grained" decomposition will be the most efficient one. Let's identify number of subdomains as S and number of available processors as N. Search area can be narrowed to values of $S \approx N$, for instance $S \in [N\, div\, 4, 4 * N]$.

An experiment was performed in environment with two quad-core 2xIntel® Xeon® processors. Size of input grid was $600 * 600$ points. Experiment calculated weather forecast for 24 h. Single run lasted 5–10 min and whole optimization took about two hours. All measured times were quite stable – deviation of execution time was 2.6 % at most for same configuration. Search for optimal value of S was carried by auto-tuner utilizing "*tunableParam*" pragma. During experiment 14 configurations were tried – 6 for different values of S and 8 configurations were derived from '*bidirectionalCycle*' pragma. Optimized program showed multiprocessor speedup $Sp(8) = 3.82$ compared to original sequential program and efficiency$E(8) = 47.75\,\%$. Following chart displays how overall calculation time depends on number of sub-domains S:

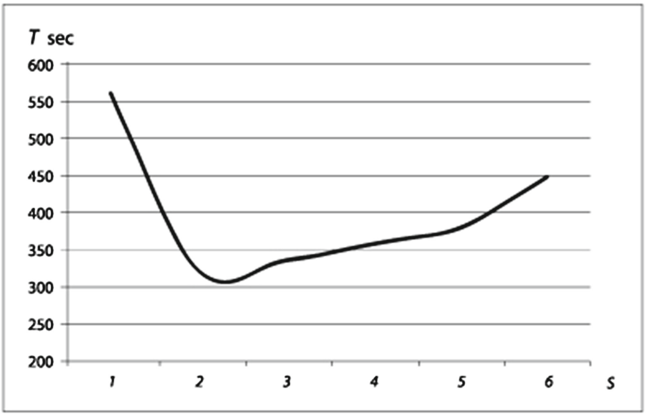


Fig. 5. Experiment results

The experiment has validated assumption about decomposition S = N to be the most efficient one but also revealed other bending point of multiprocessor speedup function:

$$Sp(N) = \frac{T_N}{T_0}$$

This point represented decomposition of input data into very small subdomains (about a hundred elements). Such result can be explained by suggestion that all data entirely fit processor caches providing additional acceleration of computation. It is worth mentioning that our framework simplifies so-called program's "behavior analysis". Since it empirically evaluates all considered program variations it is easy to discover tuned parameter's impact on overall performance (Fig. 5).

9 Related Work

The problem of creating efficient parallel programs for diverse parallel platforms has been studied extensively and many approaches have been proposed.

Parallelizing compilers include both commercial tools, such as Intel compilers [30] and open-source projects, such as OpenUH [33] and Omni OpenMP [34] compilers. Such tools don't require any changes to source code and generate a parallel executable automatically. However, they may generate inefficient code [2, 35].

Automated source-to-source transformation tools, such as Par4All [36], work in automated way like compilers, but produce source code in high-level language instead of executable binaries produced by compilers. Typically it is easier to implement such transformation than a complete parallelizing compiler. Therefore this approach can also be used in implementation of parallelizing compilers [34] and combined with advanced techniques, such as machine learning [37] and dynamic analysis [38]. Source-to-source tools have an advantage of producing code that can be manually analyzed and improved by a developer. However, for complex code structure they still produce inefficient code [37].

Semi-automated parallelization tools provide some automated optimizations, but developer can guide the system by suggesting which optimizations should be applied for each program. Examples include interactive tools like SUIF Explorer [39] and iPat/OMP [40], and also scriptable tools like PIPS [20]. Such tools require more input from the developer compared to fully automated solutions, but can produce more efficient parallel code. Auto-tuning approaches can be also described as semi-authomated tools: developer provides a set of parameters that are automatically tried to select the best option.

Auto-tuners usually are distributed among following categories [17] by their implementation details:

1. Independent libraries.
2. Stand-alone applications.
3. Part of operating Systems.
4. Compiler extensions.

Well-known examples of specialized libraries like ATLAS [18] and FFTW [19] introduce high-performance implementation of some specific functions. Unlike representatives of other categories they are tied to domain and language.

TuningGenie falls into the second category. We won't provide an extensive overview of state of art in auto-tuning. The well-known frameworks and tools that represent all categories of mentioned classification can be found in [4]. Here we'll consider two the most similar solutions.

Fist one is an ABCLibScript [21] – a Computer Language for automatic performance tuning. It defines a language in which developers can define tuning details. Similar to our pragma-based approach, additional meta-data is inserted into source code. ABCLibScript has quite powerful capabilities that even allow to perform algorithm selection based on input parameters in runtime. At the moment TuningGenie supports only pre-execution optimization. On the other hand, our pragmas are more compact and can change program's structure in a declarative manner through capabilities of rewriting rules.

Second one is Atune-IL [22] – a language extension for auto-tuning. It also uses pragmas and is not tied to some specific programming language. The main difference is again in a way how code transformation is performed. Atune-IL uses different set of templates for processing it's pragmas in different languages by employing StringTemptlate [23, 24]. It's much easier to add support of new language into Atune-IL but term rewriting engine seems more flexible. As it is said before, representing program code as a term allows easily modifying program structure. Also via knowledge base TuningGenie can exchange data between pre-tuning and tuning phases so it can initialize program variables with values that were calculated in target environment (micro-benchmarking [11]).

10 Conclusion

Development of efficient parallel software is a complex task. World-known thesis states that program is symbiosis of algorithms and data structures [25]. Parallel programming model considerably complicates both components and the most efficient programs are derived from their best combination. Variety of modern hardware architectures leads to different combinations being optimal for various computational environments. Thus optimization phase consumes sizeable amount of time and efforts in development of parallel programs. Auto-tuning methodology helps to automate process but in general requires creation of additional software that performs tuning (tuner) or significant changes in software being optimized. This paper introduces a framework for automated generation of such tuner-apps. It was designed with emphasis on minimization of required source code changes in optimized program. Mostly such changes are declarative and do not affect initial sources. Such flexibility is reached by utilizing rule-rewriting approach for code transformation which significantly differs from related solutions.

Current TuningGenie version contains quite an extensive toolset so reducing time costs of optimization is a main priority in further development. Partial execution of optimized program would dramatically reduce tuning time since current implementation of the framework fully executes target program at each tuning cycle which may not be necessary.

Adding means for software instrumentation in *runtime* is one of the most promising but quite challenging directions of further development. For instance, parallel applications would benefit from ability to switch to better-performing sub-routine based on input data characteristics as is presented for example in ABCLibScript [21].

At the moment TuningGenie is used as a separate module that needs to be included and configured to tune your application. So introducing integration with project management tools like Ivy [26], Maven [27] or Buildr [28] would significantly simplify usage of this framework.

References

1. Amdahl, G.M.: Validity of the single processor approach to achieving large-scale computing capabilities. In: AFIPS Conference Proceedings, pp. 483–485. ACM, New York (1967)
2. Intel® Parallel Compilers. http://software.intel.com/en-us/articles/automatic-parallelization-with-intel-compilers
3. Oracle® Solaris Studio. http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html
4. Naono, K., Teranishi, K., Cavazos, J., Suda, R.: Software Automatic Tuning From Concepts to State-of-the-Art Results. Springer, Berlin (2010)
5. TermWare Framework. http://www.gradsoft.ua/products/termware_end.html
6. Doroshenko, A., Shevchenko, R.: A rewriting framework for rule-based programming dynamic applications. Fundamenta Informaticae **72**(1), 95–108 (2006)
7. Eppstein, D., Galil, Z.: Parallel algorithmic techniques for combinatorial computation. Ann. Rev. Comput. Sci **3**, 233–283 (1988)
8. Sedgewick, R., Wayne, K.: Algorithms, 4th edn. Addison-Wesley Professional, Boston (2011)
9. Prokop, H.: Cache-oblivious algorithms. In: Proceedings of 40th Annual Symposium on Foundations of Computer Science (FOCS ‘99), pp. 285–299. IEEE Press, New York (1999)

10. Andon, F.I., Tseitlin, G.E., Yatsenko, E.A.: Algebraic-Algorithmic Models and Methods of Parallel Programming. Akademperiodika, Kyiv (2007). (in Russian)
11. Ma, W., Krishnamoorthy, S., Agrawal, G.: Parameterized micro-benchmarking: an auto-tuning approach for complex applications. In: Proceedings of 9th Conference on Computing Frontiers, pp. 213–222. ACM, New York (2012)
12. TermWare Tutorial. http://www.gradsoft.ua/rus/Products/termware/docs/tutorial_eng.html
13. Prusov, V.A., Doroshenko, A.E., Chernysh, R.I., Guk, L.N.: Theoretical study of a numerical method to solve a diffusion-convection problem. Cybern. Syst. Anal. **44**(2), 283–291 (2008)
14. Prusov, V.A., Doroshenko, A.E., Chernysh, R.I.: A method for numerical solution of a multidimensional convection-diffusion problem. Cybern. Syst. Anal. **45**(1), 89–95 (2009)
15. Belov, P.N., Borisenkov, E.P., Panin, B.D.: Numerical Methods of Weather Prediction. Gidrometeoizdat, Leningrad (1989). (in Russian)
16. Database of Regional Meteorological Center Offenbach. ftp://ftp-outgoing.dwd.de
17. Karcher, T., Schaefer, C., Pankratius, V.: Auto-tuning support for manycore applications - perspectives for operating systems and compilers. Technical report, University of Karlsruhe, Karlsruhe, Germany (2009)
18. Whaley, R., Petitet, A., Dongarra, J.J.: Automated empirical optimizations of software and the ATLAS project. Parallel Comput. **27**(1–2), 3–35 (2001)
19. Frigo, M., Johnson, S.: FFTW: an adaptive software architecture for the FF. Acoust. Speech Signal Process. **3**, 1381–1384 (1998)
20. Irigoin, F., Jouvelot, P., Triolet, R.: Semantical interprocedural parallelization: an overview of the PIPS project. In: 5th International Conference on Supercomputing, pp. 244–251. ACM, New York (1991)
21. Katagiri, T., Kise, K., Honda, H., Yuba, T.: ABCLibScript: a directive to support specification of an auto-tuning facility for numerical software. J. Parallel Comput. **32**(1), 92–112 (2006)
22. Schaefer, C.A., Pankratius, V., Tichy, W.F.: Atune-IL: an instrumentation language for auto-tuning parallel applications. In: Sips, H., Epema, D., Lin, H.-X. (eds.) Euro-Par 2009. LNCS, vol. 5704, pp. 9–20. Springer, Heidelberg (2009)
23. Parr, T.: A Functional language for generating structured text, May 2004
24. Parr, T.: The StringTemplate Homepage. http://www.stringtemplate.org/
25. Wirth, N.: Algorithms + Data Structures = Programs. Prentice Hall PTR, Upper Saddle River (1978)
26. Apache Ivy. http://ant.apache.org/ivy/
27. Apache Maven. http://maven.apache.org/
28. Apache Buildr. http://buildr.apache.org/
29. Kennedy, K., Allen, J.R.: Optimizing Compilers for Modern Architectures: a Dependence-based Approach. Morgan Kaufmann, San Francisco (2002)
30. Blair-Chappell, S., Stokes, A.: Parallel Programming with Intel Parallel Studio XE. Wiley, Hoboken (2012)
31. Chapman, B., Jost, G., Van Der Pas, R.: Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press, Cambridge (2007)
32. Amarasinghe, S.P., Anderson, J.M., Lam, M.S., Tseng, C.W.: An overview of the suif compiler for scalable parallel machines. In: PPSC (1995)
33. Huang, L., Eachempati, D., Hervey, M.W., Chapman, B.: Exploiting global optimizations for OpenMP programs in the OpenUH compiler. SIGPLAN Not. **44**(4), 289–290 (2009)
34. Kusano, K., Satoh, S., Sato, Mitsuhisa: Performance evaluation of the omni openmp compiler. In: Valero, Mateo, Joe, Kazuyuki, Kitsuregawa, Masaru, Tanaka, Hatsukazu (eds.) ISHPC 2000. LNCS, vol. 1940, pp. 403–414. Springer, Heidelberg (2000)

35. Tinetti, F.G., Méndez, M.: Fortran legacy software: source code update and possible parallelisation issues. SIGPLAN Fortran Forum **31**(1), 5–22 (2012)
36. Ventroux, N. et al.: SESAM/Par4All: a tool for joint exploration of mpsoc architectures and dynamic dataflow code generation. In: Proceedings of Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools, pp. 9–16. ACM, New York (2012)
37. Tournavitis, G., Wang, Z., Franke, B., O'Boyle, M.F.P.: Towards a holistic approach to auto-parallelization: integrating profile-driven parallelism detection and machine-learning based mapping. ACM SIGPLAN Not. **44**(6), 177–187 (2009)
38. Oancea, C.E., Rauchwerger, L.: Logical inference techniques for loop parallelization. In: 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 509–520. ACM, New York (2012)
39. Liao, S.-W. et al.: SUIF Explorer: an interactive and interprocedural parallelizer. In: 7th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 37–48. ACM, New York (1999)
40. Ishihara, M., Honda, H.: Development and implementation of an interactive parallelization assistance tool for OpenMP: iPat/OMP. IEICE Trans. Inf. Syst. **89**(2), 399–407 (2006)

ICT in Teaching and Learning

A Demo Processor as an Educational Tool

Leo van Moergestel[1](✉), Lukas Vermond[1], Kasper van Nieuwland[1], and John-Jules Meyer[2]

[1] HU Utrecht University of Applied Sciences, Utrecht, The Netherlands
leo.vanmoergestel@hu.nl
http://www.hu.nl
[2] Utrecht University, Utrecht, The Netherlands

Abstract. Explaining the workings of a processor can be done in several ways. Just a written explanation, some pictures, a simulator program or a real hardware demo. The research presented here is based on the idea that a slowly working hardware demo could be a nice tool to explain to IT students the inner workings of a processor.

Keywords: Processor hardware · IT training

1 Introduction

At our institute of Information and Communication Technology (ICT) at the HU Utrecht University of Applied Sciences, an introductory course on computer hardware and computer architecture is obligatory for all first year students of all IT specialisations ranging from Computer Science, Software Engineering, System and Network Engineering to Business and Information Management. For most students the inner workings of a processor and its operation at machine level remains a mystery even after reading written documentation and attending verbal explanations during the course. A simulator program can help to explain more about the details of the inner workings, however a real hardware model with the possibility to slow down the operation to the human pace of perception might also contribute to the understanding of the inner workings of a computer system. One of the problems in understanding a processor is its high speed, normally an advantage, that blurs the vision of what is actually happening. It would be nice if a device was available that shows the inner workings of a processor at low speed, but is also capable to work at a moderately high speed.

In this paper we study the implementation of a simple but useful processor, how it is used in our education and its effect on understanding the inner workings of the processor for students of different IT specialisations.

This paper will explain in the next section details about the hardware itself and the motivation of choices that have been made in the design, but first the goals of the research are formulated.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 161–177, 2014.
DOI: 10.1007/978-3-319-13206-8_8

2 Research Project Goals and Questions

The goal of this research is to build a simple, yet powerful processor that is easy to understand. This goal can be expressed is a list of requirements:

1. Students should have a look into the internal workings of such a system.
2. The processor should be easy to understand for students with a beginners level background in digital logic and hardware.
3. The working of a bus should be demonstrated in the model.
4. The system should work at a reasonable speed to make it possible to run real programs, but also single stepping by instruction or even clock-cycle should be possible.
5. The link from a higher level programming language e.g. C to the actual machine instruction should be demonstrated.

The following list of research questions was the basis for this research.

- How should a processor be built, that is easy to understand but powerful enough to run real programs written in C?
- How can we use this processor in our education?
- What is the impact of using a demo-processor for the different IT specialisations?

3 Global Description of the Hardware

First a description of the demo-system will be given. The processor has been built using medium scale integration components. The processor has the following characteristics:

- 16 bits address bus
- 16 bits data bus
- 8 internal registers
- RISC architecture [2]
- Von Neumann architecture

The processor is connected to 64 K of RAM. This memory is organised as 32768 locations of 16 bits. Figure 1 shows a picture of the hardware that has been built. The processor has been built using so-called breadboards in combination with printed circuit boards (PCB). The latter are used to implement the registers. Each register has its own PCB.

To work with the system, it should be connected to a personal computer using a special piece of hardware that connects to the USB-interface and that uses a micro-controller to control the processor system. This micro-controller can fill memory locations with instructions and data as well as controlling the mode the processor is running in, being halted, running instruction-by-instruction, running clocktick-by-clocktick or running continuously. Control of the working and clock speed of the processor can be done by using a command-line interface on the personal computer connected to the demo-system. The clockspeed can range from 0.5 Hz to 1 MHz.

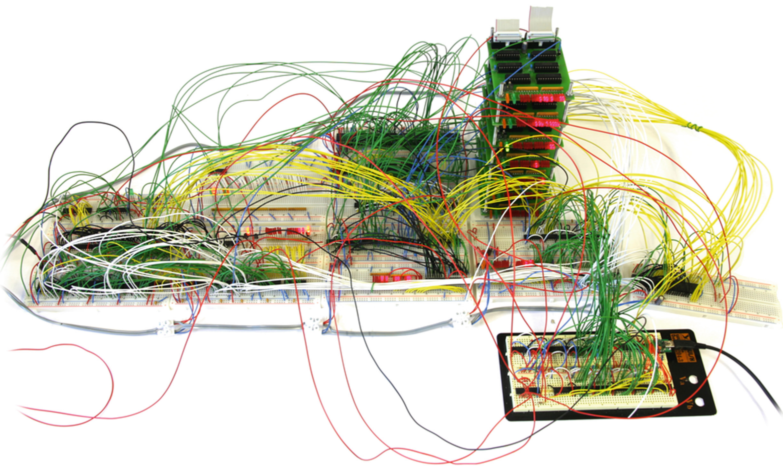


Fig. 1. Demo processor

3.1 Architecture

In Fig. 2 a schematic diagram of the system is presented. The buses are 16 bits wide as well as the registers and memory locations. This 16-bit architecture has been chosen to simplify the architecture as well as the instruction set. Yet using 16 bits makes the processor powerful enough to run programs at an affordable speed. Several concepts of modern processor architecture have deliberately been left out. Interrupts are not implemented. This will certainly be a future enhancement, but because I/O is not yet the topic of the demonstration this is not a problem. A memory management unit is not implemented in the processor. Also, special hardware for floating point arithmetic has been skipped. The reason to leave out these parts is that including them adds a huge amount of extra hardware to the system without actually contributing to the insights of

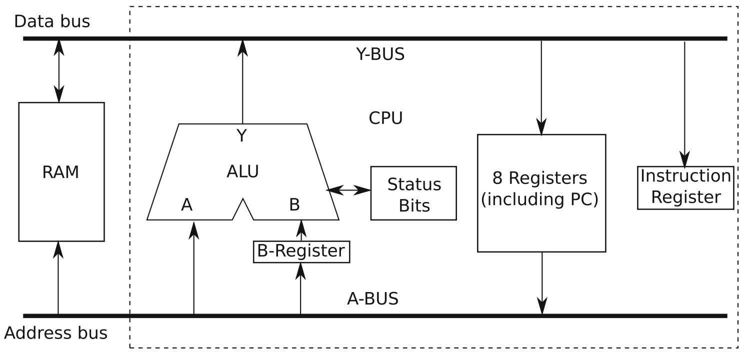


Fig. 2. Architecture.

the inner workings of a processor. Once the inner workings are understood, it seems quite acceptable that complicated instructions for floating point operation could be accomplished by using extra hardware. A memory management unit (MMU) is useful in case of implementing a multitasking operating system running several programs by using a task switcher. The MMU, being a translator of virtual addresses to physical addresses, is not necessary to be included for the understanding of the internal workings of a processor. For the same reason, the notion of user and supervisor mode, where in user mode some operations are inhibited to prevent user programs from directly accessing operating systems controlled resources [8], is also not implemented. What also has been left out in the design are speed optimizing concepts like pipelining, superscalar-design [9] and multicore systems [4]. These approaches are commonly used nowadays, but can be explained after understanding the basic details of the internal workings of a processor.

3.2 Software

From the point of view of the programmer, only the internal registers and the condition codes (flags) are important next to of course the random access memory (RAM) where the actual instructions and data are stored. Table 1 shows the set of 8 registers, numbered from 0 to 7. Register R7 is used as the program counter, in some architectures also called instruction pointer. Register R6 is the Link-Register and will be used to store the value of the return address in case of a subroutine call. This address can then be pushed onto the stack. Register R5 can be used as the stack-pointer. Every instruction is 16 or 32 bits. The 32-bit variant is only used to load 16-bit data in a specific register.

Table 1. Register set

Name	Usage
R0	General purpose
R1	General purpose
R2	General purpose
R3	General purpose
R4	General purpose
R5	Stack Pointer
R6	Link Register
R7	Program Counter

A 16-bit instruction consists of a 7-bits opcode and three fields of 3 bit each to select a register from the register set (see Fig. 3). For arithmetic and logical operations, op0 selects the destination of the operation, while op1 and op2 are used to select sources for the operations.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode							op2			op1			op0		
op3 (optional)															

Fig. 3. Instruction format

Instruction set. The instruction set is very compact and consists of the following instruction types:

Arithmetic addition: ADD, ADC (includes Carry), substraction: SUB, SBC, decrement: DEC, increment: INC and compare: CMP.
Logical bitwise AND, OR, XOR, NOT.
Shift shift left: SHL and shift right: SHR.
Load and Store load a register using an address present in another register: LOAD, STORE.
Stack PUSH and POP a register onto or from the stack.
Move and conditional move several move variants including conditional moves, that can result in a branch when the program counter is used as destination.
Subroutine calling a subroutine: CALL, a return from subroutine is implemented as a POP where the destination is the program counter.
Miscellaneous HALT stops the processor.

The processor has only two addressing modes. The immediate mode is used to load a target register with a 16-bit value that is part of the instruction itself. In this case the instruction consists of two 16-bit words. This mode can be used to put a memory address in a target register that will have the role of a pointer to a specific memory address. This results in the second possible addressing mode, where the memory address is contained in a register. Having a simple scheme for memory addressing, is common in RISC-architectures. Direct operations on memory locations as used by CISC-architectures do not fit in this approach.

3.3 Implementation

The system is built using medium scale integration (MSI) components. This results in a hardware design where all important signals are available to be displayed on LEDs. In the implementation all registers, including the instruction register and B-register, have visible contents. Every register has 16 red LEDs to indicate the contents in binary format. Two more LEDs are added to the register hardware to indicate whether a register is driving the bus or receiving data from the bus. This way the working of a bus-based system is demonstrated for internal and external operations. An orange LED indicates the fact that a register will put its data onto the bus, while a green LED indicates that a register will receive data from the bus.

3.4 C-compiler

When the inner workings of a processor are understood by students, normally the question arises: how can such a piece of hardware run a program written in a high-level language? To answer this question, the demo-processor should be programmable in a high level language. Selecting C as language is not surprising, because of its small footprint and low overhead due to its simplicity.

The C-compiler is based on LCC [6]. LCC is an open source licence C-compiler that has been developed as an educational example, thus having the same background as our processor. To make the compiler work with a specific processor, a so-called backend [1] should be written. In our implementation floating point has been skipped and integers are limited to 16-bit instead of the more commonly used 32-bit. The implementation is actually a cross-compiler, because it runs on a personal computer and generates code for the demo-processor.

LCC is compact and well documented, but a disadvantage is its poor optimisation, for example smart allocation of registers is not implemented. For our purposes this does not matter that much.

3.5 Additional Software

Two pieces of additional software have been developed. The first one is based on hardware. It is a Verilog description of the processor. In such a description, the working of the components as well as their interconnection is used to build a detailed simulation of the hardware. This description can be used in a simulator running on a personal computer that visualises the logical levels in the system for some time period.

Another piece of software is a software implementation of the instruction set. This results in a virtual machine that can be used to simulate the processor at the instruction set architecture (ISA) level [7]. Programs developed for the processor can be tested on the virtual machine. This virtual machine runs on an ordinary personal computer. Because modern personal computers use a powerful processor, the speed of the virtual machine is in our case even higher than the real hardware.

4 Usage in the Course

A simple demo program adding two integers is introduced to the students who also had a introductory course in programming. First we start with the statement at a high level:

```
a = b + c;
```

Next it is explained that variables in a program are actually memory addresses in the system where the content of that address will be the actual value of the variable. The next step is to explain that to perform arithmetic or other actions on the contents of a memory location, the content should be transferred to an internal register in the processor. This should be done for all variables involved and next the processor will perform the steps required to do the arithmetic. The result will appear in a register and should be transferred to the memory location belonging to the variable, where the result is expected to be stored. After explaining this whole story, the demo is run step by step and every instruction will show relevant changes in the contents of the register set.

The next demo is to step clock-cycle by clock-cycle through an ADD instruction. In this case the working of the internal buses Y-bus, A-bus and the usage of the B-register connected to the ALU becomes clear. The actual operation is done by adding the content of register R2, being 132, to the contents of R1, being 2000. The result will be stored in R0. First one operand is transferred from source register R2 to the B-register. When this is done, it is visible by an orange coloured LED that source register R2 will put its data on the A-bus. The receiving register, in this case the B-register, will have a green LED indicating that it will grab data from the bus. The situation is depicted in Fig. 4. Next the contents of the second operand register is put on the A-bus. This is again

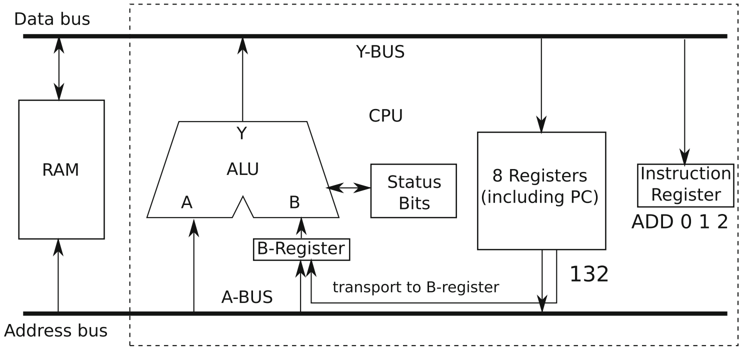


Fig. 4. Transfering 132 to register B

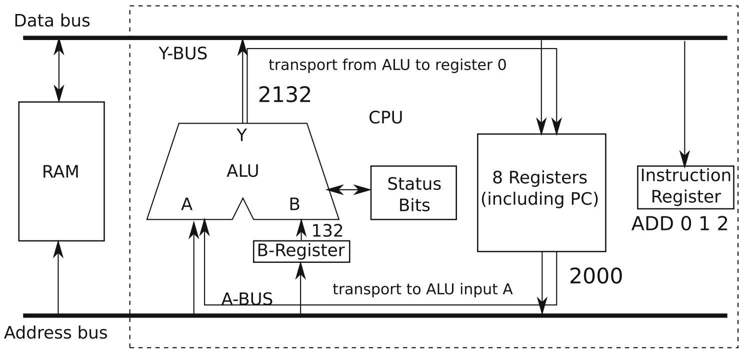


Fig. 5. Second part of add instruction

visible by an orange LED connected to the hardware of register R1. In the next clock-step the output of the ALU, being the addition of the contents of R1 and the contents of the B-register, will be transferred to the destination register R0. Here again the orange LED indicates the entity, being the ALU, that is putting data on the Y-bus and a green LED indicates R0 being the entity that will receive the data. Figure 5 shows what is happening.

5 Research and Results

The demo-system as described has been used in a course. First the internal workings of a processor was explained. Next the demos described in the previous section were given, taking 45 min including explanation of the hardware. All participants were offered a list of questions they had to fill in. The survey was conducted anonymously to prevent a situation where students might give answers to please the staff. This way they were given complete freedom to express their ideas and findings. Some questions were more relevant to the current research than other questions. The results were evaluated among the different groups of students. Table 2 shows the different groups and the number of participants in each group. Though Computer Science is represented in this table by the largest number, it is in size comparable with the number of students at Business and Information Management, while Software Engineering is actually the largest group within our institute. Only a subset of the Software Engineering group has participated in the research. In the next subsections a research question is presented and the results are discussed.

Table 2. Participants

Education	Abrev.	**N**
Business and information management	BIM	27
Software engineering	SE	39
Computer science	CS	72
Total		138

5.1 How Helpful Was the Demo?

On the inquiry list was a statement saying: *the demo helped increase my insight in the internal workings of a computer.* A set of 5 answers was possible, ranging from strongly disagree, disagree, neutral, agree and strongly agree. In Fig. 6 the results are shown where the notation $--$ has been used for strongly disagree, $-$ for disagree, 0 for neutral or do not know, $+$ for agree and $++$ for strongly agree. From the figure it becomes clear that the demo was considered to be helpful indeed. It is also interesting to see if there is a big difference between the

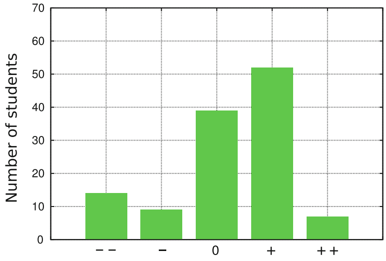


Fig. 6. Result 1, N = 138

different type of students. Taking the average of answers for the three specific groups, the difference turned out to be less than 2 %, so there is no significant difference between the three groups in this respect.

5.2 How Important is This Demo for My Study?

In the first year of their study, students did chose a particular specialisation as mentioned in Table 2 as was already introduced in the beginning of this section. The statement students had to react on was: *understanding a computer at the level presented in the demo is important for my specialisation*. Because the answers are specialisation-related and the differences per specialisation are significant, the results are presented here per specialisation. In Fig. 7 the results are shown for the Business and Information Management specialisation. Most of the students strongly disagree with the statement. This is also true in reality, because most students of this specialisation will never get involved in problems related to the details shown here. However we still believe that insight at this level will make them good participants in project groups having a strong emphasis on computer science. For students of the Computer Science specialisation, understanding at this level is in our view important. Fortunately most students of this specialisation agree with this vision as can be seen in Fig. 8.

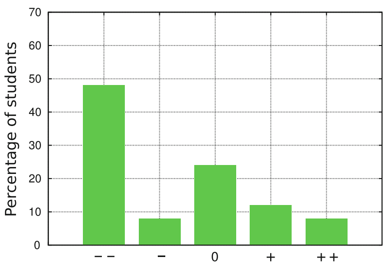


Fig. 7. Importance for BIM, N = 27

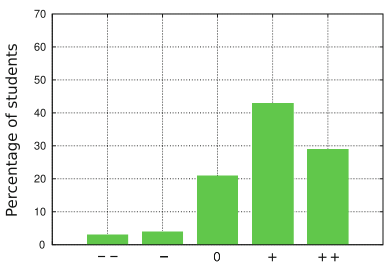


Fig. 8. Importance for CS, N = 72

Finally Fig. 9 for Software Engineering students shows results somewhere in between the Computer Science and the Business and Information Management group. From the graph in Fig. 9 it is clear that the statement that knowledge at this level is important seems to be supported by almost the same amount of students as the amount that disagree with the statement.

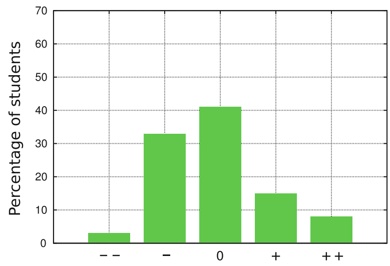


Fig. 9. Importance for Software Engineering, N = 39

5.3 Is a Student Interested in Compilers?

To investigate the eagerness of the different groups to know the relationship between high-level languages and the actual machine instructions, the next statement was presented to the students: *I want to know how a program in a high-level language will be translated to instructions for this system.* Students from the group Business and Information Management almost unanimously disagreed with the statement (see Fig. 10), while students involved with software development like the Software Engineering group (Fig. 12) and the Computer Science group (Fig. 11) clearly showed more interest. This could lead to the decision to offer these two groups the information they are interested in, while leaving the uninvolved group of students outside this particular topic.

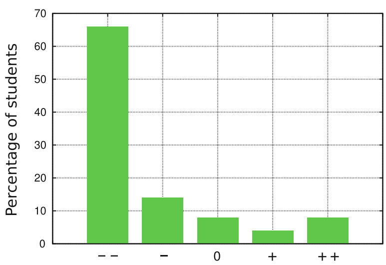


Fig. 10. Compiler BIM, N = 23

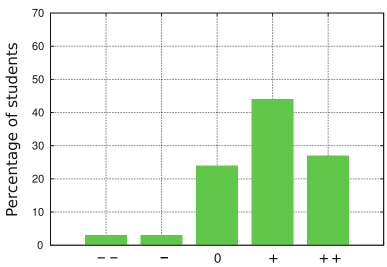


Fig. 11. Compiler CS, N = 71

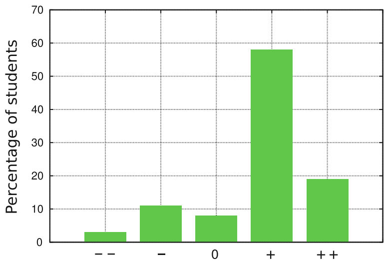


Fig. 12. Interested in compiler Software Engineering, N = 36

6 Additional Demos

Based on the results of the previous demo and comments from students, additional demos have been developed. Two aspects were elaborated:

1. The use of a high level language on the system.
2. The aspect of clockspeed.

6.1 Using C for the Processor

A simple program to discover the prime numbers starting at 7 is shown to the students.

```
int is_prime(int number){
  int divisor, remainder;
  for(divisor = 3; divisor < number; divisor +=2){
    remainder = number
      if(remainder == 0){
        return 0; /* not a prime number */
      }
  }
  return 1; /* number is prime */
}

extern void show_in_r0(int number);

int main(void){
  int number;
  for(number = 7; number < 0xFFF0; number +=2;){
    if(is_prime(number)){
      show_in_r0(number);
    }
  }
  return 0;
}
```

This program is not meant to be the most sophisticated prime discovery software, but to show to students how such a program could look in a programming language like C. Next the assembler output from the compiler is presented.

```
 .global is_prime:
    push    r1
    push    r2
    push    r3
    push    __bp
    mov     __bp, sp
    mov     __tempreg, 4
    sub     sp, sp, __tempreg

    mov     r3, 3
    sto16(__bp, -2, r3)
    jmp     __L5
__L2:
    ldo16(r0, __bp, 10)
    ldo16(r1, __bp, -2)
    call    __sdiv16
    sto(__bp, -4, r1)
    ldo16(r3, __bp, -4)
    mov     r2, 0
    scmp    r3, r2
    jne     __L6
    mov     r0, 0
    jmp     __L1
__L6:
__L3:
    ldo16(r3, __bp, -2)
    mov     r2, 2
    add     r3, r3, r2
    sto16(__bp, -2, r3)
__L5:
    ldo16(r3, __bp, -2)
    ldo16(r2, __bp, 10)
    scmp    r3, r2
    jlt     __L2
    mov     r0, 1
__L1:
    mov     sp, __bp
    pop     __bp
    pop     r3
    pop     r2
    pop     r1
    ret
```

Even though only the is_prime function is displayed, this piece of assembly language needs a lot of explanation. To understand the mapping of the high level instructions to assembly language, insight is needed in:

- the concept of a label like __L1.
- standard assembler instructions like mov.
- macro constructions like ldo16(r2, __bp, 10)

Next the concept of using the stackpointer sp in combination with the so-called basepointer __bp should be explained. In Fig. 13 the stack frame is shown when the function is_prime has been called. Using this information, students are encouraged to add comments to the assembly code. This way the mapping of a function in C to a subroutine in assembly language becomes clear.

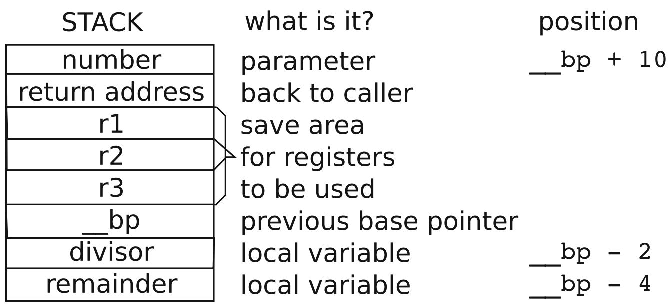


Fig. 13. Stack frame for the function is_prime

Subroutine Calling Convention. For a function to be called, at assembly level, the parameters should be available. The stack mechanism is used for this purpose. After pushing the parameters onto the stack, the function is called as a subroutine. This will also push the return address onto the stack. After returning from the subroutine, the stackpointer is adjusted to the value it had before the parameters were pushed on the stack. Consider the call of a function sum:

```
int sum(int a, int b, int c)
```

The parameters are pushed onto the stack:

```
push c
push b
push a
```

Next the function is called:

```
call sum
```

After returning the stackpointer is adjusted:

```
mov  r4, 6 ;6 bytes have been used for the parameters
add  sp, sp, r4 ;stackpointer = stackpointer + 6
```

This calling convention is a stack-based calling convention and it heavily uses the stack as its name suggests. For RISC architectures alternative approaches exist that rely on using internal registers [5]. The processor introduced here is actually a RISC processor, but it lacks a huge amount of internal registers. So the stack-based approach is used here.

C Runtime Startoff. The next piece of magic that should be solved and showed to the students is the C runtime startoff. This piece of code will initialise the stackpointer and clear the memory locations used by uninitialised global variables. After doing this, the processor is led to the entry point of `main`. After a reset the processor always starts at memory location 0. The C compiler is actually a crosscompiler, running on a standard PC. After successfully compiling the C program, the resulting machine code is uploaded to the demo system. The processor starts at location 0 where the runtime startoff code is situated. For convenience multiply, division, modulo and shift routines have also been included in the C runtime startoff.

Lessons to be Learned. The demo described here will explain a lot of aspects to the students:

– The need for high level languages. From the example it becomes clear that programming at the assembly level is complicated, error-prone and the code is difficult to maintain. Normally this is told to students without actually showing why. This demo does exactly that.
– The mapping of high level language constructs to actual machine instructions. The control of flow constructions are not available at the assembler level. Students will understand that jumping to labels, that might also be used in high level languages, makes code difficult to read and difficult to maintain. This was already noted by Dijkstra in his famous article "Go To Considered Harmful" [3].
– For programmers using languages like C and C++, it becomes clear, why local variables in a function should be initialised before use and why these variables are not available any more after the function retuned. From the listing it becomes clear that room for local variables is allocated on the stack. These locations contain bits from previous actions or just random values.
– For C programmers, it is shown that only a single return value is possible and this value is put in a register.

6.2 The Aspect of Speed

The first demo was specially designed to show the workings of the processor at low speed. To understand the power of a computer a feeling for the speed they are working at helps to understand why a set of relatively simple instructions can result in complex problem solving in a short time. The prime generating program of the previous subsection has been used to build this demo. First the processor runs at 5 Hz and will pause for a few seconds when a prime number has been found. This same program was then run at a clockspeed of 50 Hz, 200 Hz and finally the maximum speed, being around 1 MHz for this hardware.

7 Related Work

In the late seventies of the previous century, so called developments kits to learn the concepts of microprocessors were available. Nowadays these kits are replaced by software simulations of conceptual processor models.

What is still available are training kits for processor designs. However, these kits do not focus on students, but on professionals who should be trained for this specific piece of hardware. These professionals do already have a firm understanding of the internal workings of a processor and use these kits for prototyping embedded systems or other processor-based applications. To discover the inner workings of a processor, expensive hardware tools can be used like logic analysers, in combination with JTAG scanning to read internal registers in the processor. An alternative is presented here, but the complexity has been scaled down. For educational purposes this is an advantage.

A trend in modern computer science training is the introduction of cheap single board systems like the Raspberry Pi or the Arduino. These systems can certainly play an important role in the training and education of computer science students, but they are not comparable with the system presented here. The Raspberry Pi is based on a powerful and complex ARM processor, even though this processor is actually a RISC-architecture. To understand the architecture and the instruction set takes much more time. The same is true for the Arduino. The system that we have built should act like a stepping stone to give students a first basis of understanding. The fact that single stepping is easy to accomplish in combination with the continuous display of the internal registers contents makes our demo processor a powerful tool for educational purposes. The next step should be the introduction of processors as used in the aforementioned single board systems.

Working with a software simulator is also useful in training, but students are intrigued by the demo processor built in hardware. A special feature that this real hardware-based demo processor can show is the effect that the clock speed for hardware is limited. If the clock speed is too high, the processor will not function properly.

8 Discussion

In this paper the focus was on implementing a hardware demo-model for a simple but useful processor. We studied the effects on different groups of IT students. It turned out that all groups agreed on the fact that the understanding of the internal workings of a processor had been increased. There were however differences between the groups. The Computer Science group showed more interest and also were more eager to grasp the working of a processor because they know, it is essential for their specialisation. This eagerness is not triggered in the Business and Information Management group. However this group should not necessary have the in-depth insight in the internal workings of a processor.

9 Future Work

Though this demo-processor turned out to be a useful tool in our education of IT students, there are still improvements possible. A printed circuit board should be developed, so the system is not as vulnerable as it is now. Apart from the fact, that a higher clock-rate might be possible, this also opens the possibility to give the system to students for a hands-on training. This opens the possibility for much more experiments that in this case will be done by the students and not just a demo shown by the instructor in a relatively short time. Some professionally made demonstrations should be made available, so students can replay the lesson taught.

The results of the survey so far are sometimes based on a relatively small number of students. By repeating the demo for students that will start next year, the reliability of the results will become better.

10 Conclusion

In this paper a demo-system has been presented. This demo was tested on different groups of students to investigate its usefulness. The conclusion is that students state that a hardware demo helps to improve the insights in the inner workings of a processor. To make the system even more useful future enhancements should be worked on like:

- let students from the Software Engineering and Computer Science group develop simple programs in C and run them on the system;
- show optimisation techniques used by the compiler;
- let students work with the already available software simulator, based on the same hardware as the real demo-processor;
- let students work with the real hardware that is built on a printed circuit board;
- add serial I/O to the system including interrupts, so there is a more direct interaction possible and interrupt handling can be demonstrated.

References

1. Aho, A., Lam, M., Sethi, S., Ullman, J.: Compilers: Principles, Techniques and Tools, 2nd edn. Pearson, New York (2006)
2. Dandamudi, S.: Guide to RISC Processors: For Programmers and Engineers. Springer, New York (2005)

3. Dijkstra, E.: Go to considered harmful. Commun. ACM **11**(3), 147–148 (1968)
4. Flynn, M.J.: Computer Architecture: Pipelined and Parallel Processor Design. Jones & Bartlett, Boston (1995)
5. Frantzen, M., Shuey, M.: Stackghost: Hardware facilitated stack protection. In: Proceedings of the 10th Usenix Security Symposium, USENIX 10, pp. 55–66 (2001)
6. Hanson, D., Fraser, C.: A Retargetable C Compiler: Design and Implementation. Addison-Wessley, Reading (1995)
7. Smith, J., Nair, R.: Virtual Machines. Elsevier Inc., Amsterdam (2005)
8. Tanenbaum, A., Austin, T.: Structured Computer Organisation. Prentice Hall, Englewood Cliffs (2012)
9. Zomaya, A., Katevenis, M.: RISC Architectures, Chap 20 in The Book: Parallel and Distributed Computing Handbook. McGraw-Hill, New York (1995)

Challenges and Confusions in Learning Version Control with Git

Ville Isomöttönen(✉) and Michael Cochez

Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
{ville.isomottonen,michael.cochez}@jyu.fi

Abstract. Scholars agree on the importance of incorporating use of version control systems (VCSs) into computing curricula, so as to be able to prepare students for today's distributed and collaborative work places. One of the present-day distributed version control systems (DVCSs) is *Git*, the system we have used on several courses. In this paper, we report on the challenges for learning and using the system based on a survey data collected from a project-based course and our own teaching experiences during several different kinds of computing courses. The results of this analysis are discussed and recommendations are made.

Keywords: Version control systems · Git · Computer Science Education

Introduction

Version control systems (VCSs) have been actively used since the '70s. The older tools like Source Code Control System (SCCS) [22] and Revision Control System (RCS) [23] only supported storage of the versions on a locally mounted file system. Later systems were characterized as centralized systems, the well-known representatives being the Concurrent Versions System (CVS) [2] and Subversion (SVN) [6], the latter of which is still extensively used. In the present day, many users have adopted distributed version control systems (DVCS) (see [7]) where each user has a local copy of the repository that can be synchronized with other repositories, introducing more flexibility, and perhaps more complexity, to the system. The decentralized technology allow for change tracking, reversibility, and manageable collaborative work. In this paper, we are concerned with learning and using DVCSs in computing education context. The system we have used is Git[1].

Previous research suggests that there is an agreement that learning and using VCSs would be important for computer science students (see, e.g., [16]). Version control systems have been used in various computing courses, while there is

[1] http://git-scm.com/

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 178–193, 2014.
DOI: 10.1007/978-3-319-13206-8_9

still a call for more empirical research on the student usage of the systems (see, e.g., [19]). We have previously studied students' use of Git by analyzing repositories with various metrics [4,5]. In the present paper, we continue this research by focusing on student and teacher (authors) experiences. The former is based on a survey targeted to students of a project-based course, while our own experiences originate from several computing courses. Our main research perspective are the challenges the students report and we have identifed through teaching.

1 On Version Control Systems in Education

Version control systems (VCS) have been adopted in a variety of educational settings. When this kind of system is used in a course, teachers try to use the tool to support their traditional teaching practices. Liu et al. [15], for instance, used the system to monitor and visualize the contributions of the members of student teams. Clifton et al. [3] used a VCS for course management including materials, assignments, and learning processes. Others, like Hartness [10], adopted version control to promote realistic software development experiences. Further, Robles and González-Barahona [20] used the version control system to detect plagiarism by students, which was made possible by the fact that the system tracks versions for all students. Besides the automated plagiarism detection, they adopted black-box testing to automatically assess the functional requirements the software has to meet every time students submitted code into the repository. Lastly, they used the code submitted to the version control system to automatically generate personalized exams. More specific to programming education, we find a work by Glassy [9]. In his work, he tries to use the VCS not only as a tool to monitor students' progress, but also to understand *how* students develop code. In general, project-based courses are perhaps the most natural habitat for VCSs that naturally support team work and can help a teacher estimate student activity for assessment and grading [25]. We also notice that use of VCSs has been required of non-CS majors [13], and that these systems are employed outside the computer education domain. For instance, Lee et al. [14] used it to manage creative writing work.

VCSs are not specifically intended for use in educational settings, and several difficulties and confusions may emerge. As noted in earlier work by Reid and Wilson [19], who used the CVS system, there might be confusion in judging which of the students' assignment versions was the final one. This can be addressed if the students adopt the tagging feature of the system [13]. Other problems that Reid and Wilson noticed was that students mix up different functionalities of the CVS, namely the check out and update commands. Also, a well developed background knowledge for the teaching assistants was deemed to be necessary. Most issues were considered to be due to the lack of a proper mental model of the VCS system used. Time and resources needed for setting up and managing repositories are yet another challenge raised by the same authors. More recently, Lawrence et al. [13] addressed this management overhead issue by using cloud services. Glassy [9] found out that VCSs do not urge students to adapt to a more iterative work progress. Instead, students tended to postpone working on the assignments. We have earlier concluded that if a VCS is given to students as a submission tool, they might not adopt the professional use of the system but only use it for submission purposes. In general, student difficulties are reflected in the study by Robles and González-Barahona [20] who noted that when students commited code to Git, they had to ask them to upload the code to a website "just in case they did not use git in a proper way."

Perhaps the main reason for incorporating VCSs into computing curricula, as agreed by many, is that current globalized workflows require tools which support distributed collaboration [16]. Despite the challenges summarized above, there are clear indications that students are able to adopt VCSs. With support in place, even non-CS majors have successfully used the basic functions of the system [13]. In this connection, it has been observed that Git is not a tool with a flat learning curve; Xu [25] points out that it might take a long time for students to familiarize with a VCS. Familiarity seems also to be the main reason why Rocco and Lloyd [21] found that commit frequency is increasing during a course. The authors designed two assignments. For the first one, they set a minimum commit frequency while the second one had no requirements. Still, only 75 % obtained a acceptable commit frequency for the first exercise compared to 81 % for the latter one. The authors note that not only were the students able to grasp the basics of the VCS (Mercurial), but they tended to continue to take advantage of the tool later on. A similar observation can be found in Milentijevic et al. [17]. Their students considered a VCS useful, but only after they got sufficiently familiar with it. In accord, as we argued in our earlier research [5], there is a need to include VCSs as an integral part of a computing curriculum. Milentijevic et al. [17] take the idea further by proposing a generalized model. This model suggests the adoption of VCSs as a support tool in project-based learning scenarios.

In the present article, we aim to contribute to this body of literature by surveying student opinions about using Git through a preliminary qualitative study. We also report detailed challenges and confusions which we have observed during our daily teaching practice across several courses. In our view, this approach adds to the studies exploring usage patterns from repository data.

2 Study

This paper is concerned with challenges that students encounter when they start learning and using Git. We addressed this concern by (1) analyzing data that were collected from a Bachelor level project course (Project) with a survey and (2) by reporting our own teaching experiences in terms of what kinds of issues can confuse students when they begin to learn the system. Our teaching experiences originate from the Project course, specific VCS supervision sessions provided during an introductory second-year software engineering course (SE Course), and two master's level practice-oriented courses where the use of Git has been encouraged as a group work tool and all the assignments have been managed using the system. The supervision sessions during SE Course have been run during two course instances with altogether 170 passed students, providing the principal experiential source for the present paper.

The survey of the Project course was a voluntary research questionnaire issued to students at the end of the course (in Spring 2013). It requested the opinion of the students on the questions below:

- Q1 (open-ended): Do you think that using the VCS was useful?
- Q2 (open-ended): What difficulties did you encounter regarding version control?
- Q3 (5-point Likert scale): Do you think that your group used the VCS in an efficient manner?
- Q4 (5-point Likert scale): Evaluate how actively you used the system.
- Q5 (open-ended): Described how you used the VCS?
- Q6 (open-ended): Did you read the commit messages of others?
- Q7 (open-ended): Did you find the messages beneficial?

The survey was answered by 21 out of 26 students, all of whom granted a research permission. The data were tabulated into a spreadsheet file, which enabled us to have all the data available in a structured manner, while extracting the themes under particular open-ended questions and making observations about the Likert scale data. Of the open-ended questions, the data from questions Q1, Q2, Q5, and Q7 were analyzed by raising the interesting points and regularities in the respective data, which is a usual step in coding qualitative data (see [18]). With these small data, numbers (quantification) are added to the presentation only if we observed some particularly dominant aspect in the data. Question Q6 yielded quite short comments which we divided into few categories and present with frequencies in a table, similar to the Likert scale data from questions Q3 and Q4. Q7 was analyzed and is presented in connection to the closely related question Q6. The first author conducted the analysis of the survey data, while the results of the analysis were challenged and discussed together with the second author for an agreement.

We argue that reporting teaching experiences as an exploration of our own observations and experiences (our second research instrument) is a beneficial approach, as DVCSs have not yet been studied extensively in an educational setting. Then, rather than basing a study on external hypotheses sourced from the literature, it should be reasonable to report observations about teaching and learning obstacles, which can inform subsequent studies. In general, there are research approaches such as self-ethnography, which characterize our role as researchers: we are authentic participants of the research setting (educational organization) returning to remembered challenges in our daily practices (see [1]).

3 Course Settings

The *Project* is a 5-credit bachelor level course whose main goal is to develop students' understanding of group processes and software processes. The course was conducted under the topical theme of Open Data. Students innovated, designed, and implemented prototypes of Open Data software products in small groups. Work rooms were provided to students during the course to support autonomous and independent work. All the groups were guided to use a VCS system for their group work. All groups selected to use Git, perhaps because many of the students had at least tried the system during their earlier courses. Majority of the groups managed their repositories in YouSource service[2] of the university, while some groups used Git Hub[3] and Bitbucket[4]. The course is targeted to second-year students, while the participants were in different stages of their bachelor studies. It is, however, a teacher observation that most of the students had relatively little previous experience with Git, that is, experimentation with the basic use case through fetch and merge, add and commit, and push commands.

SE Course is a 3-credit lectured course for second-year students. A course assignment done in groups and an end-of-course individual exam are required for completion. The lectures focus on project management and the phases of a software life-cycle. The mandatory course assignment is the preparation of a project plan, which is done in small groups. Mandatory supervision sessions on version control were arranged at the beginning of the course in order to encourage all the students to use the distributed version control system Git for the group assignment. During 2012 autumn, there were 88 passed attendees and regarding the autumn 2013 the number is now around 90 (at the time of writing this paper; a few groups still working on the assignment).

The supervision sessions begin with a demonstration on Git usage, after which students continue to practice the topics demonstrated. We use the command line Git client for this teaching. The supervisors are available during practicing and actively guide the students. A discussion is prompted throughout the session to be able to help all the attendees to go through the basics. The sessions have provided a good opportunity to identify the aspects that need to be emphasized in teaching, as it has turned out that one and the same aspects have emerged as difficulties in these sessions.

Master's level courses referred to here are *Service oriented architectures and cloud computing for developers* (SOA&CC) and *Design of Agent-Based Systems* (Agent). The SOA&CC course acquaints students to the use of digital services and the concept of cloud computing while the Agent course aims at introducing agent systems as a novel software development paradigm.

Both courses are worth 5 credits, but students of the SOA&CC course can make an optional individual assignment which can lead to an additional 5 credits. The courses have a somewhat similar structure: during the courses students work in small groups and all of their study time is devoted to programming the exercises. Two or three weekly sessions are arranged for the group work, during which the teacher assists the students in their work. Further, mandatory contact sessions focusing on reflective program review are arranged when a part of the

[2] https://yousource.it.jyu.fi
[3] https://github.com
[4] https://bitbucket.org

course is completed. Both courses emphasize students' self-direction. The aim is to engage students without having a traditional grading system. Instead, the task of the teacher has to generate genuine interest from the students towards the topic. In [12], the authors analyze how the teaching model used in this course motivates students.

The version control system Git is suggested to students as a group work tool. The teacher does also use it to receive submissions from the students. In the SOA&CC course, Git is also used to deploy code to Platform as a Service (PaaS) providers. For the development of the agent platform in the Agent course, students use Git as a collaboration tool; both in the group and with the course teacher. At the beginning of the Agent course, each students had to perform a preliminary task with the Git version control system in a temporary repository. Concretely, there was a simulated scenario where two development branches (their own and the teacher's) had diverged from each other. The students' task was then to merge these two lines of development into one.

4 Survey Results

4.1 Q1: Usefulness

All respondents (100 %) reported that they experienced the use of VCS (Git) as beneficial, as they saw it as a development tool for shared work and as a group work tool for sharing and communicating. From this premise, it clearly seems that an authentic project-based, group-based, assignment makes students aware of the usefulness and necessity of a VCS. Authentic group assignment makes them learn VCS more profoundly as compared to 'copying and pasting' the Git commands during some previous courses:

> During this course, I finally learned to use Git little better, as I earlier just copied the commands and used them. Now I am also able to resolve upcoming problems with it [translated from Finnish by the authors].

Students also comment that they experienced using the system as useful even though they did not take advantage of its full potential. A point where the students especially pay attention to the usefulness of the system appears to be their first attempts to integrate software functionalities advanced in their group setting.

Only negative comment in the answers to this question relates to the difficulties in using the system and an experience of the lost of work in the system (Git).

4.2 Q2: Difficulties

Many students reported difficulties with the system as they worked on the same parts of the software under development. That is, they experienced conflicts that they encountered as something difficult and problematic (32 %). Some comments hint that conflicts were resolved by cloning a clean repository from the remote Git machine:

Table 1. Students' view of how efficiently they used the Git system

Do you think that your group used the VCS in an efficient manner?					
scale:	1 = No	2	3	4	5 = Yes
f:	2	3	7	7	2
Evaluate how actively you used the system.					
scale:	1 = low	2	3	4	5 = high
f:	1	3	9	6	2

> The version control system looks like a simple software, but it has very complicated functions that are difficult to grasp. This I have encountered at the time of conflicts, when I am committing, and usually the fastest solution is to clone the repository again... [translated by us from Finnish to English]

Perceiving conflicts as problems is likely to indicate lack of routine in resolving them; Repeated cloning is a very unfortunate solution and likely to cause frustration.

Another difficulty reported is, unsurprisingly, the lack of routine and experience. Some students reported that they deliberately used only the basic features of the system to avoid problems (e.g. conflicts), while others report that learning to use the system takes time. Overall, it seems that the students hesitated to use some features of the system under the project course where their goal was to produce working software during a fixed time period.

Finding help in a self-directed manner for more advanced features was also reported to take time and be difficult—our interpretation is that reading Git documentation may turn out to be difficult if a mental model of the system and the logic of its commands is unclear. Some students also had difficulties in learning to use the Git clients they chose to use.

Yet another interesting point in the data was that students noticed that a VCS does not replace project management. That is, they reported that, by using this system, they did not yet have sufficient knowledge of whether other members in the group were advancing their tasks.

4.3 Q3 & Q4: Efficiency

Student answers to Likert scale questions Q3 and Q4 are displayed in Table 1. We find the answers to accord with the above aspects drawn from the qualitative data. That is, the students found the system to be highly beneficial and used it, while there were some hesitation and restrictions for efficient use.

4.4 Q5: Usage Patterns

Students reported that they mainly adopted a usage pattern where they "pulled" at the beginning of a work session and then "pushed" in the end. Some students

Table 2. Activity in reading commit messages

Did you read the commit messages of others?		
response	f	reasons
NO	6	direct communication or tools such as IRC used, no need to
YES	11	of which 3 mentioned this to be due to the tool, i.e., commits reviewed in the web page of the Git service system or in the graphical client UI
ROUGHLY	4	of which two refer to rather accidental activity enabled by the tool used

commented that they reverted their staged changes if they noticed that they did not produce working solutions during their work session or someone appeared to have worked on the same parts of the software. Again, we would see these kinds of solutions as an unfortunate usage pattern—here, the students could have relied on branching instead of reverting their work in progress. Accordingly, some answers seem to indicate that the students dared to push their changes to a remote machine only if they considered their work to be a working solution.

We also observe that there was variation in the work style of the students as some students reported that they frequently checked for changes in the remote machine on top of executing the pulls and pushes at the ends of their work session. Overall, the students would have benefited from discussion on the good routine, e.g., concerning the flexibility and value of small commits.

4.5 Q6 & Q7: Commit Messages

Table 2 presents student activity in reading the commit messages. Little over half of the respondents (52 %) reported that they read the messages, while three of these students noted that this was caused by the tool used. Almost the same number of the respondents reported that they did not read at all the messages or skimmed them through accidentally (categories NO and ROUGHLY sum up to 47 %). It seems to us that as student groups were provided with work rooms during the project course, they relied quite a lot on direct face to face communication which has lessened the experienced need for reading the messages. On the other hand, as a side note, many groups struggled with communication, making us to conclude that in particular in this project work setting students would have benefited from better routine with reading the messages.

By the analysis of the answers to Q7, we can point out more reasons under the categories of previous table. With respect to students who answered NO to reading activity (see Table 2), we now find confirmation that some of these students regarded the communication going on in their project room as their commit messages. Students in this category also not that it was better to directly look at the code. Of the students who replied YES or ROUGHLY to the question about reading activity, some refer to only small benefits due to small group size, and also preferred looking directly at the code. Some students valued the commit messages as they clarified, located, time-stamped, and hinted about the changes made by others.

One interesting aspect emerging from the data is that commit messages are considered more useful towards the end of the projects. It may seem natural for the students to write more useful messages when the software product has grown, as it is then reasonable to provide explanation for peers about the changes committed. This is an aspect that could be studied among the professional software developers and compared with educational setting: how do a professional developer attend to commit message writing when the complexity of the product is still small?

Yet another aspects in the data were that commit messages provided a snapshot describing the project state for the students and that, altogether, the messages were difficult to be trusted without deeper looks at the code.

4.6 Summary

Taken together:

- A VCS is experienced useful and necessary, but the limited resources of a project assignment may discourage students to experiment with and take advantage of the advanced features of a VCS.
- On the other hand, authentic project assignment makes students to experience benefits to the degree that they see using the system advantageous and are willing to use it.
- The basic routine in terms of requesting the changes in the remote machine at the beginning of a work session and sharing changes in the remote machine at the end of the session appears to be well absorbed, while there seems to be variation in how frequently students interact with the remote machine. In our interpretation, students need not only conceptual explanation of the inner workings of Git, but also of what a distributed VCS implies and enables in terms of usage patterns. These students would have benefited from encouragement towards and information on the benefits of frequent commits and that branching can be useful for committing the work in progress.
- In our view, the most interesting aspect emerging from the data on commit message writing, constituting a future research question on the writing activity and informativeness, is that this writing may naturally become dependent on group size, communication distance, and the current complexity of the shared product development. The finding that students considered their presence in the work room removing a need for informative commit messages is likely to explain our previous observation of nonsensical commit messages [5], while this appears to be less prevailing in a project-based course as compared to exercise-based courses [4].

5 Teaching Experiences

Compared to the survey results, this section provides a much more detailed view to the challenges we have encountered. Based on our own experiences as teachers, we raise several aspects that confuse or constrain student learning of Git DVCS. Notice that many of the reported challenges relate to the use of Git command line client in teaching.

5.1 Mixing Git and Bash

Student with little previous command line (unix) experience easily confuse Git commands with Bash commands, and the other way around. Especially, the commands used for file manipulation, like removing, copying, etc. cause problems. For instance, when learners are asked to remove a folder from the file system, they start writing *git ...*.

5.2 Branches as Folders or Files

Manipulating branches, in particular listing them with the *git branch* command, causes learners to regard branches as folders or files in the file system (see Fig. 1), and a newly initiated branch as a copy of what it was initiated from. While these conceptualizations may not be overly harmful, they seem to complicate the learners' Git usage. For instance, if asked to checkout a particular branch, learners might try to move into a branch with the `cd` command. In order to avoid this confusion, the topic should be explicitly discussed. This challenge can be in part attributed to the use of the command line client—hence, in such a setting we stress the importance of showing visual diagrams of branching and explicitly explaining branches as pointers in the commit history.

We have also noticed learners' preference for commands like

- git add.
- git add –all
- git commit -a

which make their lives 'easier' since the repository and their directory are perfectly 'synchronized'. However, these easy-way-out commands can turn into a habit by which student omit the questions of which of their files are reasonable

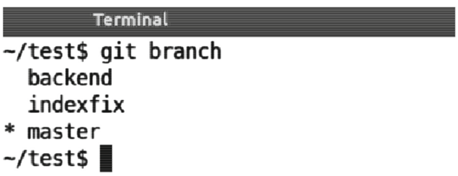


Fig. 1. Output of `git branch` command resembles a listing of folder contents

to be managed with VCS and why they should also consider small and frequent commits. For instance, in our SE Course supervision sessions, we try to avoid this to occur by using more specific forms of the commands during the first demonstrations of the basic commands.

5.3 'Git Add File' Creates a New File

In Git system, changes are "staged" for the upcoming commit using `git add` command, thus there is a lot of flexibility regarding which of the current changes the user likes to commit. We have noticed a tendency where the command `git add` *file* is, however, associated with creating a new file, while it stages the file for the commit. We speculate that the confusion with the `git add` command simply arises from it being associated with the natural language. Considered together with the `git commit` command, which usually ensues the `add` commands, it is admittedly easy to think that "one creates a new file and then commits it". Further, in our SE Course setting, where the command line Git client is used, we often ask learners to create arbitrary test files to practice with, which may have further encouraged this misconception. In order that learners would readily grasp the actual meaning of the `git add` command, we would include the bringing of existing files into the local Git working directory in addition to creating new ones, for instance.

5.4 Origin/Master Versus Origin Master

In the Git system, the branches fetched from a remote machine are marked with origin/*branch-name*, where origin is an alias for a remote repository url[5] and *branch-name* is the name of the remote branch[6]. This presentation for remote branches is used by Git when listing and merging the remote branches, for instance. However, when pushing the state of the local repository and its branch to a remote machine the 'origin/' prefix is not used, as the user gives the origin in any event in the command:

```
git push origin master:(no prefix here)master
```

A shorter and more frequently used version of this command is

`git push origin master`,

where the currently checked-out local branch is pushed (imagined in front of the semicolon) to a master branch in the origin.

In referring to remote branches, this 'with and without prefix' conventions confuses learners until they have used the system long enough and grasped the

[5] *origin* is a default alias for a remote repository created automatically when cloning a repository. User have the ability to assign another term as the alias or add several remotes due to the distributed nature of the system.

[6] By default the name of the remote branch is the same as the name of the local branch, one could however choose a different name for the local branch.

need for specifying the remote context with a prefix in doing and showing certain tasks. In our view, this is an issue to be just repeatedly explained to learners to mitigate their potential irritableness, untile the inner logics of Git commands are internalized.

5.5 No Default Branch on a Remote Machine

When creating a repository on our faculty's Git server the repository does not contain a default master branch to start with, but a branch that the remote service needs for managing the projects and repositories internally. If the user clones the repository and starts working, the command Git push will not initilize a master brancher in the remote machine if not directly specified. This results in students working unintentionally with a meta-file branch in the remote machine. Here, beginners who do not yet possess knowledge of branches may encounter odd situations making Git appear a very forbidding system. Thus, the point we address here is that configurations of the Git service systems might cause great difficulties for beginners.

5.6 Accidental Sub-repositories

Making sub-repositories lead to confusion and problems (most of the time by accident). For instance, students begin to experiment with `git init` command, and execute it with and without a name for the repository with the result that they end up with several repositories arranged hierarchically in their home folder tree. Students start pulling and committing but do not see changes correctly. This happens even more easily when different or several tools/terminal windows are used at the same time. In our master's level practical courses, for instance, IDE is used for committing and terminal to check what has changed.

5.7 Blind-Testing Effect

The absence of a proper mental model seems to cause the blind-testing effect known in the field of the automatic assessment of programming exercises. The effect describes the situation where a student aims to make progress with arbitrary changes into a programming code, opportunistically submitting the code into the assessment system. Similarly, learners do not think about the Git commands, but "just try them out", which makes it very difficult to interpret the responses and error messages by the Git client.

Similar to Bash commands, VIM as the default editor imposes additional cognitive load on learners, as they practice Git in the command line environment. Already the different modes of the editor cause accidental actions and editor behavior, which occasionally irritates learners and makes them attempt to escape the editor with arbitrary key combinations. For instance, if the saving of a commit message is interrupted in VIM, an erroneous state in versioning emerges, one that requires removing an index lock in the local .git folder.

5.8 Absence of Authentic Use Cases

When Git is taught as an intensive course, separate from realistic group assignments (SE Course supervision sessions), or on courses where assignments are momentary exercises (the master's level courses we refer to in this paper), there is a chance that learners associate issues such as branching with rather unorthodox use cases. For instance, they might consider that a new branch should be created for each required deliverable of the course. Moreover, typical weekly lab exercises in engineering courses communicated through systems such as a Git characterize VCS as a medium for short term storage. We have concluded earlier that using a VCS as a submission management tool, while useful, may cause learners to exactly use it as such a tool instead of a one for distributed group work [5]. In the same vein, momentary repositories created for weekly lab session run a risk of advertising systems such as a Git unfavorably. The reason is that creating a repository always involves a small amount of overhead which is not significant if the benefits of the system are clear. We also noticed in our earlier research that students who are working in a project course produce better commit messages as when they use Git for weekly course work [4] (in the present paper we analyzed the students' own view of the messages). We recommend teachers to pay attention to how to communicate course management use cases and more realistic VCS use cases to learners, and even to use another kinds of tools for pure submission returns.

5.9 Attitudinal Barriers

Due to initial apparent complexity of a DVCS tool, the learners make comparison to the systems where real-time shared writing is possible in a WYSIWYG environment. We have instructed Git using the command line client, which in part may explain learners' dislikes, as graphical UI/interface is and would be a much more typical environment for many present-day learners. Further, if Git is practiced in short intensive sessions using small arbitrary test files, student are not yet shown the capacities of the system, and comparisons to cloud services such as Google Docs are very likely to occur. We have tried to overcome this barrier by discussing software engineering-oriented use cases with the learners, e.g., managing large amount of code in a distributed project. In our experiences, this discussion should occur and can remove the attitudinal barrier. In the agent based systems course (master's level), students were asked during the first week to push their answers for certain programming tasks to the repository. The teacher also created a repository which contained possible answers for the tasks. The students were asked to merge the two 'development branches' during the second week. This task was received very well and students understood the usefulness of multiple remotes and branches.

6 Conclusions

In this paper, we have presented challenges in using Git DVCS in a project-based course based on a research survey targeted to students. Through the analysis of the primarily qualitative data, several viewpoints describing challenges experienced by the students were raised. We complemented this analysis with various detailed issues based on our teaching experiences collected from several different courses.

From the research survey, we concluded that while a realistic project course makes students to value the use of a VCS, there are also constraints such as the limited resources available during a product-oriented course, which may hamper independent learning about a version control tool. Regarding the commit messages, a future research question emerged regarding to what extent writing explanatory commit messages is dependent on group size, communication distance, and complexity of the product.

Our teaching experiences have made us consider whether viable mental models can arise from practice-first approach. To some extent, we have taught Git using theory and practice together, the approach of which is also echoed in recent pedagogy, e.g., in integrative pedagogy [24]. During SE Course supervision sessions, for instance, we have illustrated several computers on a whiteboard and connected basic Git commands to such a picture, and yet demonstrated this scheme to students with two teachers and computers. It seems to us, through the challenges reported in this paper, that the further one goes from basic features and use cases, the more important it is to communicate to student what really takes place in Git internally. For example, we have explained branches to students conceptually through the ideas such as ‘switching’ the view in the commit history. In retrospective, using this particular example, we would explicitly speak of pointers in the commit history to avoid delivering a view of branches as folders or local copies of folders. In general, the challenges we found while teaching Git could be compared to those observed when teaching other VCSs. This could reveal whether the challenges are specific to Git, its distributed nature, the command line interface, or version controlling in general.

Several small challenges we reported in the section for teaching experiences make us compare the research on learning and using DVCSs with the research on beginner programming difficulties where the mental models held by students has been studied to identify and react to misunderstandings among learners. Thus, we consider that similar methods could be used to advance the research reported here, e.g., think-aloud problem-solving [11] and phenomenography [8].

We have studied student use of DVCS (Git) through multiple courses and iterations of those courses, and, overall, are confident that students are able to adopt skills needed early, while challenges at different levels tend to remain without a constant use. Consider here, as an example, our observation of student hesitation to use the advanced features of Git under limited resources of a product-oriented project-based course. We opinion that VCS tools need to be integrated throughout the curriculum, which implies that attention should be paid to awareness and education of department personnel and teaching assistants, so as to promote this professional skill in a continuous and gradual manner among the students.

References

1. Alvesson, M.: Methodology for close up studies – struggling with closeness and closure. High. Educ. **46**(2), 167–193 (2003)
2. Cederqvist, P.: Version management with CVS. Technical report, Signum Support AB, Linkoping, Sweden (1993)
3. Clifton, C., Kaczmarczyk, L.C., Mrozek, M.: Subverting the fundamentals sequence: using version control to enhance course management. SIGCSE Bull. **39**(1), 86–90 (2007)
4. Cochez, M., Isomöttönen, V., Tirronen, V., Itkonen, J.: How do computer science students use distributed version control systems? In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2013. CCIS, vol. 412, pp. 210–228. Springer, Heidelberg (2013)
5. Cochez, M., Isomöttönen, V., Tirronen, V., Itkonen, J.: The use of distributed version control systems in advanced programming courses. In: Proceedings of the 9th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, vol. 1000, pp. 221–235. CEUR (2013)
6. Collins-Sussman, B., Fitzpatrick, B.W., Pilato, C.M.: Version Control with Subversion. O'Reilly Media, Sebastopol (2004)
7. de Alwis, B., Sillito, J.: Why are software projects moving from centralized to decentralized version control systems? In: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. CHASE '09, pp. 36–39. IEEE Computer Society, Washington, DC (2009)
8. Eckerdal, A., Thuné, M., Berglund, A.: What does it take to learn 'programming thinking'? In: Proceedings of the First International Workshop on Computing Education Research. ICER '05, pp. 135–142. ACM, New York (2005)
9. Glassy, L.: Using version control to observe student software development processes. J. Comput. Sci. Coll. **21**(3), 99–106 (2006)
10. Hartness, K.T.N.: Eclipse and CVS for group projects. J. Comput. Sci. Coll. **21**(4), 217–222 (2006)
11. Herman, G.L., Kaczmarczyk, L., Loui, M.C., Zilles, C.: Proof by incomplete enumeration and other logical misconceptions. In: ICER '08: Proceeding of the Fourth International Workshop on Computing Education Research, pp. 59–70. ACM, New York (2008)
12. Isomöttönen, V., Tirronen, V., Cochez, M.: Issues with a course that emphasizes self-direction. In: Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education. ITiCSE '13, pp. 111–116. ACM, New York (2013)
13. Lawrance, J., Jung, S., Wiseman, C.: Git on the cloud in the classroom. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education. SIGCSE '13, pp. 639–644. ACM, New York (2013)
14. Lee, B.G., Chang, K.H., Narayanan, N.H.: An integrated approach to version control management in computer supported collaborative writing. In: Proceedings of the 36th Annual Southeast Regional Conference. ACM-SE 36, pp. 34–43. ACM, New York (1998)
15. Liu, Y., Stroulia, E., Wong, K., German, D.: Using CVS historical information to understand how students develop software. In: MRS 2004: International Workshop on Mining Software Repositories (2004)

16. Meneely, A., Williams, L.: On preparing students for distributed software development with a synchronous, collaborative development platform. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education. SIGCSE '09, pp. 529–533. ACM, New York (2009)
17. Milentijevic, I., Ciric, V., Vojinovic, O.: Version control in project-based learning. Comput. Educ. **50**(4), 1331–1338 (2008)
18. Miles, M.B., Huberman, A.M.: Qualitative Data Analysis: A Sourcebook of New Methods. Sage, Beverly Hills (1984)
19. Reid, K.L., Wilson, G.V.: Learning by doing: introducing version control as a way to manage student assignments. In: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. SIGCSE '05, pp. 272–276. ACM, New York (2005)
20. Robles, G., Gonzalez-Barahona, J.: Mining student repositories to gain learning analytics. An experience report. In: Global Engineering Education Conference (EDUCON), pp. 1249–1254. IEEE, March 2013
21. Rocco, D., Lloyd, W.: Distributed version control in the classroom. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. SIGCSE '11, pp. 637–642. ACM, New York (2011)
22. Rochkind, M.: The source code control system. IEEE Trans. Softw. Eng. **SE-1**(4), 364–370 (1975)
23. Tichy, W.F.: RCS – a system for version control. Softw.: Pract. Exp. **15**(7), 637–654 (1985)
24. Tynjälä, P.: Perspectives into learning at the workplace. Educ. Res. Rev. **3**, 130–154 (2008)
25. Xu, Z.: Using Git to manage capstone software projects. In: ICCGI 2012: The Seventh International Multi-conference on Computing in the Global Information Technology, pp. 159–164. IARIA (2012)

Information Competence of University Students in Ukraine: Development Status and Prospects

Eugenia Spivakovska, Nataliya Osipova, Maksim Vinnik(✉), and Yulia Tarasich

Kherson State University, 27, 40 rokiv Zhovtnya St., Kherson 73000, Ukraine
{Spivakovska, Natalie, Vinnik, YuTarasich}@kspu.edu

Abstract. The paper describes the methods of information-competence formation for Ukrainian university students. The Information-Technology Competences Model has been built. Pedagogical conditions for the effective formation of student's information competency are identified and implemented. The results of the EU-funded project TEMPUS TACIS "ECDL for Ukrainian Administrators" are presented. The survey also incorporates the results from the Ministry of Education and Science of Ukraine's project, "Creating an Internet Portal for ECDL Distance Learning for Higher Educational Establishments". For basic information competences level formation of students, it is suggested to use the ECDL international program. The experience of Kherson State University in ICT training for university students in accordance with the international standards is also demonstrated.

Keywords: Education and learning process · Information and communication pedagogical environment · Information competence · Information technology

1 Introduction

The globalization of the world economy and business processes is a push for the globalization of higher education. Among universities of the world there is currently a fighting for influence on the educational markets of other countries. The leaders of the "global" education are the United States and the European Union. The globalization of education became possible due to the development of innovative educational technology, and thus IT becomes a key strategic resource of university. The educational establishment which can create the best conditions and resources for teaching using IT comes to a new level of a modern world university. Thus, the requirements for education in Ukraine need to focus on the balance on the one hand - the possibility of the educational establishment to have appropriate strategy resources, on the other hand - the students' readiness to use them. Rapid IT branch development characterizes an interaction system of university strategic resources and students' readiness to use these resources as a dynamic system. ICT development and globalization of education create new educational competitive environment which features are ICT and students' mobility. To ensure highly competitive of universities on the educational services

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 194–216, 2014.
DOI: 10.1007/978-3-319-13206-8_10

market the universities should have better innovated educational technology than their competitors have.

One of the most important criteria for professional competitiveness in the job market today is its ability to learn, acquire knowledge independently, use the acquired knowledge in the new environment and professional situations, think creatively and make unconventional decisions. Accordingly, there is a necessity of implementation of a new educational paradigm.

The implementation of ICT in the educational process and the problems associated with the creation of information and communication teaching environments and the adaption of the education system to the new social, informational and technological needs have been the focus of scientists since the 1960 s. A significant contribution to solving the problems of implementing innovative technologies in education were made by both national and foreign scientists: O. Spivakovskiy, V. Shirokov, M. Lvov, L. Petukhova, G. Kravtsov, A. Gurzhiy, S. Rakov, M. Zhaldak, V. Glushkov, H. Reinhold, E. Wenger, K. Swan, P. Shea, V. Sholohovich, S. Papert, B. Hunter, A. Kryvosheyev and many others.

Competences associated with the growth of the information society, mastering of this technology and understanding of weak and strong sides of their application, critical judgments in relation to the information which is spreading by means of mass media and advertising is one of the five core competences adopted by the European Council [1].

Among the key competences identified by Russian scientists (A. Hutorskiy, I. Zymnya, A. Markova, S. Shyshov) are the competences in the sphere of information technology: receiving, processing, information presentation and its transforming (reading, precis-writing), mass media und multimedia technology, computer literacy; use of Internet technology (by I. Zymnya).

In the list of key competences determined by Ukrainian educators, there are also competences in information and communication technology which include student's ability to navigate the information space, own and use the information according to the labor market needs. They are related to technical and technological qualities of an educated person, prepared for life and active work under the conditions of modern high-technology information society, covering the main components of an information culture of the students, based on rational coexistence with technosphere, according to their professional self-determination and taking into account the individual possibilities (based on discussion materials organized within United Nations Development Programs project "Education Policy and Education" "peer-to-peer", 2004). David McCann, Jenny Christmass, Nicholson, Peter and Jeremy Stuparich determine that ICT can be used to meet the changing needs within the educational sector: for more flexible studying; extension of university services to national and international markets; stronger economic effect of higher education development in extremely competitive environment [2].

Joseph Lee, Ng Lai Hong and Ng Lai Ling in their turn said that the success of any virtual environment for studying depends on the skills of students and their attitudes toward the technology used [3].

Ukrainian scientists Alexander Spivakovskiy and Lubov Petukhova emphasize the need to establish the information and communication teaching environment (three-subject didactics) which in their opinion, helps to teach and learn, make education more accessible, especially for those who lack these educational materials, develop learning culture, creating, sharing and cooperation in the rapidly changing knowledge-based society, thus forming a positive attitude to studying, desire to study, acquire knowledge, and as a result provides positive motivation for studying in a new information educational space [4, 18].

Psychological and educational literature of the latest decade is dedicated to the informational support of education includes such a term as "information and educational environment", indicating new integrating essence of educational and information environment.

Target of research: use of information and communication technology in the preparation of university students in Ukraine.

Statement of purpose: formation of information competence of university students in Ukraine.

The hypothesis of this study is based on the assumption that the process of information competences formation in Ukrainian university students will be effective under the following conditions:

1. The availability of ICT infrastructure at the institution (establishment of information and communication pedagogical environment)
2. The pedagogical, didactic and psychological readiness of lecturers to incorporate ICT into their professional activities
3. Use of the ECDL international program to create a basic level of information competency in first-year students
4. The systematic use of innovative teaching and ICT in the study of academic disciplines
5. The formation of positive motivation in students to use ICT in their professional activities.

2 Previous Related Research

This research is the continuation of the research work carried out by Kherson State University from 2010–2012 under contract number IT/583-2009, dated 10.23.2009, within the state program of informational support for Ukraine, as well as our research on the current condition of distance education in Ukraine, the influence of the quality of electronic educational resources on the quality of educational services using distance learning technology [5, 34], and an building ICT infrastructure at the university. The research also uses the results of the EU project TEMPUS TACIS "ECDL for Ukrainian Administrators". The institutions cooperating in the project are the University of Nice Sophia-Antipolis (France), Kherson National Technical University (Ukraine), the University of Klagenfurt (Austria), Kherson State University (Ukraine), and the Centre of Retraining of Civil Servants of Kherson Province (Ukraine). The results of the Ministry of Education and Science of Ukraine's project "Creating the Internet Portal for Distance Learning ECDL for Higher Educational Establishments" were also used in research. Some of the data obtained in the framework of the project of Ministry of Education and Science of Ukraine "Creating Electronic Data Bank on Distance Learning for Higher Pedagogical Education" are also presented in the article. The authors of the paper participated in all these projects.

3 Methodology

In this section we describe methods of research, information-technology competency model and the pedagogical conditions to form students' IT competency.

3.1 Research Methods

To solve the problems, a range of methods are used:

Theoretical – the analysis, synthesis, comparison, generalization and systematization of psychological and educational literature, legal documents, standards and informational resources are consulted to define the essence of university-students information competency and assess the theoretical and methodological bases of information competency formation in university students; pedagogical modeling - to build the model of forming these competences; generalization of scientific, theoretical, and practical data to determine the pedagogical conditions of information competences formation of university students during the study of specialized subjects;

Empirical - questionnaires, surveys, testing, and self-esteem; pedagogical experiments to test the study hypotheses;

Methods of mathematical statistics are used to determine the reliability of the results on the basis of quantitative and qualitative analysis of the empirical data.

3.2 Information Technology Competency Model

We analyzed IT competency [11, 12, 17, 18, 35], and defined the set of IT competency necessary for all students need to succeed in practice and education in today's digital era. This body of competency was evaluated and condensed to create the three parts of the Student's IT Competency Model:

1. Basic Computer Competency
2. IT Literacy
3. IT Management

Once the competency categories were established, each was aligned with an existing set of competences maintained by standard development organizations or de facto standards. For example, excellent alignment was found with the existing standards of the European Computer Driving License (ECDL) Foundation for basic computer competency [12, 25–31].

The certification is carried out according to worldwide standards. The European Computer Driving License, European Computer Law Certificate is an accepted standard not only in Europe but also throughout the world, one that confirms that a holder has the basic concepts of information technology and is able to use a personal computer and perform basic applications.

ECDL certification has certain advantages for the university and it raises graduates' competitiveness, simplifies the process of obtaining of international grants to continue education, establishes a transnational system of knowledge measurement, verifies educational quality, enhances the prestige of Ukrainian education, raises the recognition of the competence of Ukrainian students abroad, increases students' mobility, and allows students to participate in exchange programs.

Since 2003, Ukraine has been a participant of the ECDL program. ECDL as a standard of basic computer competences is recognized by the European Commission, UNESCO, the Council of European Professional Information Technology Societies, the European IT Community, and the Ministries of Education indifferent countries.

IT competence at a higher level is determined by educational qualification characteristics (EQC), i.e. qualification requirements for social and productive activities of the graduates of a certain specialty, educational, and qualification level. EQC is a component of the industry components of state educational standards of Ukraine [35].

Table 1 illustrates the relationship between the competency category and the standard development organization.

Table 1. Student's Information Competences Model

Component of the model	Standard	Source
Basic computer competences	European Computer Driving License	European Computer
		Driving License
		Foundation
		www.ecdl.org
Information literacy	Industry standard of higher education	Ministry of Education and Science of Ukraine
	Educational and qualification characteristics in specialty	mon.gov.ua
Information management	Industry standard of higher education	Ministry of Education and Science of Ukraine
	Educational and qualification characteristics in specialty	mon.gov.ua

3.3 The Pedagogical Conditions to form Students' IT Competency

In this section we determinate the pedagogical conditions to form IT competency of university students.

An analysis of the educational literature and results of previous projects have allowed us to define the pedagogical conditions for the effective formation of IT competency in university students:

1. The availability of ICT infrastructure at the institution (the establishment of an information and communication pedagogical environment)
2. Pedagogical, didactic and psychological readiness of lecturers to use ICT in their professional activities

3. The use ECDL international program to generate the basic level of IT competency of first-year students
4. The systematic use of innovative teaching and ICT in the study of academic disciplines
5. Positive motivation for students to use ICT in their professional activities.

4 Research Results

In this section we report on the implementation of pedagogical conditions for forming the information competences of university students.

4.1 Availability of ICT Infrastructure at the Institution

Under the research programme funded by the Ministry of Education and Science of Ukraine entitled "Creating Electronic Data Bank on Distance Learning for Higher Pedagogical Education" the research was conducted at 73 educational institutions.

To assess the presence and status of ICT infrastructure, a questionnaire (with an accompanying letter from Ministry of Education and Science of Ukraine) was developed, and surveys of higher educational institutions were conducted.

In the last three years the number of laptop computers used by the 1st year students has increased by 43 % and 5th year students by 36 % (Fig. 1). Moreover, all respondents having appropriate technical means use laptops for studying. Thus, we can assert the availability of laptops among students of IT-related specialties is determined by their professional orientation, as it is seen from the questionnaire analysis carried out among the students of these specialties, as well as by the rapid information and technological development of the society, as it is indicated in the responses of a survey carried out among the students of other specialties.

One of the most important aspects in this educational environment is the free access to information, which is ensured through the existence of LAN at the university and access to the Internet. The main aim of the educational establishment is to provide students access to the necessary resources at any time and from any location on campus. So, today we speak about 100 % of availability of local networks, their direct access to the Internet in all universities of Ukraine and established on their base regional (5 % of universities) and corporate (26 %) networks. However, we should note that the provision of access to educational resources in hostels paying attention about more than 40 % of the universities.

Access dynamics evaluation to the Internet of the 1st year students during the last three years makes it possible to draw the following conclusion (Fig. 2):

1. Students access level to the Internet at home has increased from 45 to 90 % during three years
2. The number of students who use the Internet access at higher education establishments increased by 40 % and is 70 % in 2013
3. Indicators of students, who do not have access to the Internet, have reduced from 30 % to 2 %.

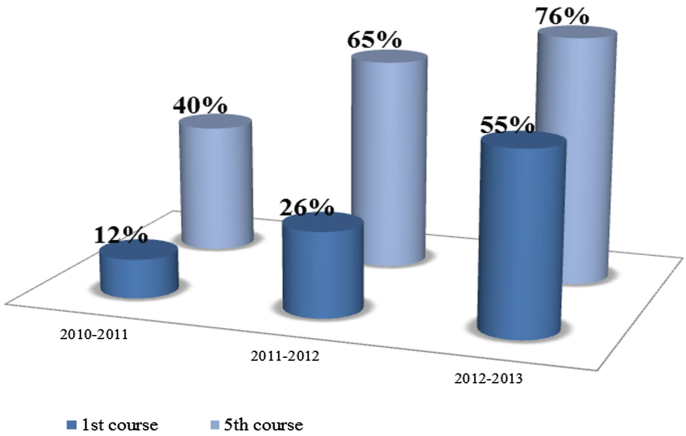


Fig. 1. Number of computer equipment in 2010–2013 academic years used by the 1st- and 5th-year students

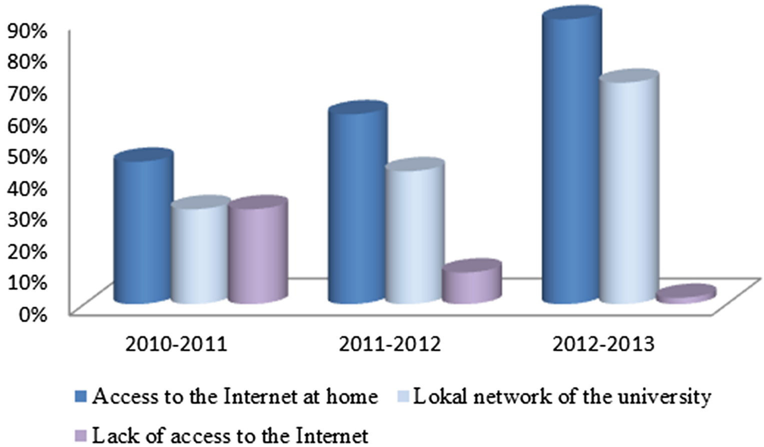


Fig. 2. Access to the Internet by the 1st year students as of 2010–2013

Wireless technology (Wi-Fi) is able to provide an increase of client accessibility and free access to necessary data without being tied to a fixed workplace. Nowadays, 60 % of universities in Ukraine use Wi-Fi technology, besides the third part of educational establishments have appropriate resources not only in academic buildings, but also in students' dormitories and other territorial units of the university.

Use of Wi-Fi technology by the university allows you to create an additional "client" place that is important today, because most users prefer to use their own computer equipment. In addition, the use of this resource provides:

- Access to e-learning servers
- Access to the Internet

- Work with personal folder on a file server of computer classrooms
- Work with corporate e-mail of computer classrooms
- Work in a local network with access to all resources
- Work in research networks of the world, etc.

which has a direct impact on the formation of a high level of information competence of university students.

A central player in the formation of material and technical resources of ICPE are distance learning systems (DLS) and electronic means for educational purposes (EMEP). Thus, teachers having access to resources and can create their own e-courses, fill them with the necessary methodological support, and use the necessary EMEP to improve the demonstrativeness of the proposed material. Students have access to the resources offered by a teacher, get more opportunities for independent materials processing, and are able to conduct online discussions.

According to the results of questions, the official site has 73 % of all pedagogical educational institutions, e-mail address indicate 70 % of all pedagogical universities. Information on developed distance-learning courses and distance-learning centers were included on only 30 % of educational institutions' websites. According to the results, the ensuring of educational process with electronic textbooks (textbooks, lectures), guidelines and electronic tests consider to be satisfactory - 70 % of universities, unsatisfactory - 30 % of universities. Almost 20 % of universities have not implemented distance learning. In most universities, the distance learning is still in the development and testing stages.

Using the appropriate means both students and teachers are able to create informative and educational pages and groups, provide for the general access needed educational resources and files etc. Popularity of respective networks causes a higher percentage of students' attendance of the proposed resources by lecturer, possibility of real-time communication, creation of account-oriented prototype of educational services.

Nowadays, almost 100 % of the students have their own accounts on such social networks like VK, Odnoklassniki, Facebook and others. The most popular social network among the students is "Vkontakte" (64 %), according to the data provided by the Foundation of Public Opinion (Russia) [22]. A large number of students use the corporate mail Google + (26 %) and the remaining 10 % of respondents prefer such networks as "Odnoklassniki", "Facebook", "Twitter," and others (Fig. 3). The leader in students' computer communication tools is "Skype", used by 80 % of surveyed respondents.

Computer communication means providing the ability to create new educational projects, to discuss work specifics between the participants, to share common objects, and to get access from anywhere (Odnoklassniki, Vkontakte, Facebook, Twitter, Skype, ICQ, Google + , e-mail). Practically, all data is publicly available for collaboration with students, teachers, and scientists, through creating workgroups.

Kherson State University has developed the ICT infrastructure comprising:

- Computer training labs, university departments to provide ICT infrastructure
- Internet, LAN, WiFi in academic buildings and dormitories
- Official website (www.kspu.edu), ICT resource sites

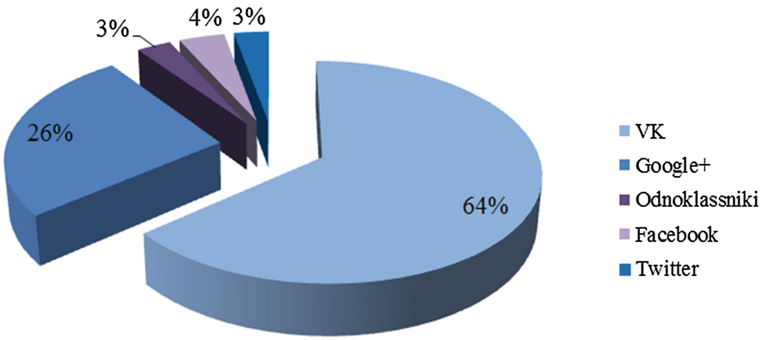


Fig. 3. Students' use of social networks and communication means

- Distance learning websites, Kherson Virtual University and KSU Online
- Providing studying using electronic textbooks (means, lectures), guidelines and electronic tests
- An ECDL Testing Center was established and provides on-line testing.

4.2 Teachers ICT Readiness

The report of UNESCO on information technology in education shows that in Ukraine there is a "rapid advancement of information and communication technology into the education sphere, which needs continuous improvement in the efficiency of use of new ICT in the educational process, timely updates of educational content and an increase in the quality of training in ICT". However, there are some problems which are primarily associated with a low readiness of teachers (psychological, methodological and pedagogical) to rapid changes in information technology.

The formation process of modern specialists is provided by a qualitative increase in efficiency and productivity of subject-subject relations due to pedagogical communication in the modern university. This is facilitated by purposeful deliberation by teacher organization designed to communicate with students on a personal level. The teacher acts as an organizer and initiator of communicative-competence communication on a practical and equal basis. So the modern teacher should have a set of professional and personal qualities.

The optimal aspects in the system of teacher-student are relationships of cooperation, partnership, mediated interpersonal relationships based on mutual respect and deference to the individual, rather than formal relations with considerable social distance. An important role in modern education is the educational environment, actually the role of a separate subject [10]. Therefore, the modern teacher should be prepared to organize the learning process in terms of the ICT environment, using the latest pedagogical approaches and innovative techniques.

For more effective use of ICT in the educational process, the teacher should learn methods and strategies which promote critical thinking and problem-solving skills (particularly relating to future professional activity). Teachers should inspire students with the desire and motivation for constant self-improvement and life-long learning.

ICT and services associated with them quickly develop and spread (Internet, social networks, blogs, virtual libraries, e-books, digital audio, video, photo, cell phones, instant messaging, Internet telephony, handheld computer, etc.) can make education more interesting and a teacher's job more efficient and diverse. Therefore, it is necessary to train teachers to incorporate technology and to create and modify software and educational products, educational presentations, web resources, etc., to continuously improve their own qualifications.

ICT use by teachers in their professional activity involves the possession of innovative technology, elements of which are:

1. Content structuring on a modular system
2. Problematic education
3. Use of self-evaluation techniques of educational-cooperative effectiveness
4. The organization of work in small groups

For the effective use of ICT in a professional activity, the teacher should create individual work, group work, project, research and any other work in an IC educational environment.

The arsenal of a teacher, who is planning to use ICT, should include a set of both traditional and innovative research methods. For example, projects' methods, modeling within educational process of professional activity specialists' conditions, didactic games, etc. All of these techniques contribute primarily to reveal the creative potential of students.

A teacher's readiness to use ICT in learning activities involves skills and the appropriate use of various educational means: teaching materials, e-learning systems, web 2.0, and a cloud technology.

To form a teacher's readiness to use ICT in his/her professional activities at Kherson State University, we:

- Implemented a joint program of the Ministry of Education and Science and Microsoft Corporation, "Partnership in Education," and taught the teachers information technology basics (organized courses and master classes for teachers)
- Trained teachers on the Intel program "Education for the Future"
- Organized teachers' training in universities of Ukraine and abroad, in particular within the framework of European projects
- Had University professors receive Intel certificates for "Education for the Future"
- Ensured the university teachers were trained at Microsoft Academy and certified as Microsoft Specialist
- Made sure teachers passed independent testing on ECDL program and got certificates
- Had 2 teachers of Kherson State University train to be authorized ECDL test center trainers
- Held regular seminars and workshops for teachers with a view to explore the possibilities of distance learning systems at Kherson Virtual University and KSU Online.

4.3 Students' Basic IT Competency and ECDL Program

In this section the use of ECDL international program to create the basic level of students' IT competency in the first year of studying will be presented.

One way to resolve the contradiction that exists between the need of information and communication technology use in the educational process of the university and the lack of initial level of students' information competence is to implement the international ECDL program for the first year students (1 semester).

The ECDL course program has 7 modules:

- Module 1 – Basic knowledge of information technology
- Module 2 – Using computer and working with operating systems
- Module 3 – Text processing
- Module 4 – Spreadsheets
- Module 5 – Databases
- Module 6 – Presentation
- Module 7 – Information and Communication

We should note that students somewhat overestimate their level of ICT skills. So first-year students were given a questionnaire for the purpose of self-assessment of their ICT skills level. For the same group of students there was held ECDL incoming testing. Analysis of the results shows the following: a high level of information competence-4 % of students (37 % for self-assessment); medium - 82 % of students (60 % for self-assessment); low - 14 % (3 % for self-assessment).

Kherson State University where an authorized ECDL test center has been created has an extensive experience in the analysis of ICT competences and learning computer technology.

Within the project of the European Union TEMPUS TACIS "ECDL for Ukrainian Administrators" and Ministry of Education and Science of Ukraine "Creating the Internet Portal for Distance Learning ECDL for higher educational establishments" at KSU the software and methodological support designed for use in lecture and class and distance learning forms for the study of ECDL courses.

As a result of the projects, the integrated environment has been implemented that includes:

1. Electronic materials for teachers, including:
   - The course program "Basics of ICT for workers of educational, professional and technical and higher educational establishments designed in compliance with the international standards of ECDL"
   - Presentation of the lecture material
   - Objectives for practical work
2. Electronic materials for students, including:
   - Summaries of lectures
   - Problems for practical and independent works
   - Tests for self-evaluation

3. Knowledge evaluation system in the field of ICT, including:
   - Intranet test system of knowledge evaluation and skills of students in the field of ICT use
   - Automatic testing system (software module)
4. For publishing the abovementioned materials (1-3) online and also for the management of the training processes the integrated Web-based application.

Thus, the developed educational and methodical complex contains input questionnaire, final questionnaire, seven modules (according to ECDL program), each of which has an identical structure: module's program (teaching and thematic plan), requirements for knowledge and skills of students and audience, module files (archive which contains methodic recommendation for practical works, files needed for the implementation of each practical work and demonstration materials), lecture materials, module tests, journal.

Surveying in the beginning of the course enables students to evaluate ICT user level and promotes reflection and reflective attitude towards studying.

The teacher may have in the beginning of the educational process to determine the general group level in a certain module, identify the opportunities list for software applications which most of the students do not know how to use and learning on how to use it can cause casualties, and therefore require more time.

As a result of the course studying the applicant to pass ECDL tests has to have a basic understanding of the basic concepts of information technology, understand such concepts as aspects of safety and factors of possible computer harmful effect on the health condition and others (Module 1); to know basic functional possibilities of computer and its operating system and be able to apply this knowledge in practice (Module 2); to possess the skills to work with a text editor (Module 3); to handle numeric and text data using electronic tables (Module 4); have basic concepts of databases and ability to use databases (Module 5); to be able to create, format, modify changes in presentation for displaying or printing (Module 6); to know the basics and terms relating to the Internet, understand the use of e-mail tools as well as the basics of security; to be able to search for information online using search engines and browser and work with messages (Module 7).

Teaching materials should be constructed in such way that there is a choice of modules that need to be studied as well as the procedure for passing such modules. The first module is a theoretical one, others - practical. ECDL tests are designed to show a basic knowledge of computers and applications regardless the scope of their application. It opens wide prospects of the course in preparing future specialists of all specialties and can be used as a spline-course for the first year students. Some modules can be partially or fully used in the study of certain disciplines.

We should note that for success of the real ECDL test the student should have a sufficient level of correct answers (above 85,7–88,9 %). Using this resource allows to influence directly such factors as: the quality of knowledge, motivation and information competence of the student.

The developed software and methodological materials for the ECDL program are used to form the basic IT competency of university students in the course "Computer Information Technology," held during their first year of studying.

4.4 Innovative Pedagogical Methods and ICT in Studying Academic Disciplines

As shows the experience of higher educational establishments the development in the world, effectiveness of preparation of future professionals is in the dialectical unity of educational and studying process, ensuring close relationship of professional training and practice. From this perspective a special role is the problem of theoretical grounding and experimental testing of appropriate pedagogical techniques for the organization of the educational process [8].

Today, students are accustomed to the selective and free perception of information flows. The desire to learn sequentially from textbooks has been replaced by a desire to learn through participation, conducting experiments, and through social practice. Consequently, there is a need for major changes in models and types of the classroom and independent work of students.

Organization of the educational process should be done to ensure that in the process of studying, the students solve not only job situations but also real, practical problems, ones that are learned not only from the teacher but also from each other. This process should involve working with databases and using a lot of information and communication technologies, leading students to become integral participants in the information and communication pedagogical environments (ICPE) of higher education institutions (HEI).

The description of the unique possibilities of new means of information technology deserves special focus, the implementation of which creates conditions for unprecedented intensification of the educational process and the creation of a new methodology focusing on the development of the individual student [23]. These features include the presence of feedback, educational information visualization, easy access to textbook archives, methodological support from scientific periodicals, the automation of information and search activity processes, and the management organization of educational activity and knowledge quality control.

Let's return to the definition of ICPE: Information and communication pedagogical environments are capable of self-development by systematically organized aggregate data transmission facilities, information resources, interaction protocols, hardware and software, and organizational and methods of support oriented to meet the educational needs of the users, which includes teaching, methodical materials, creative tasks for students, automated elements of educational process management and mandatory provision of opportunities to work with the Internet resources [18].

The use of EMEP allows students to focus their attention on discussing issues at the moment, to determine their level of informative competence, to acquaint themselves with modern teaching aids, to coordinate the group work. Electronic libraries enable rapid access to the relevant literature. It can be said that students' access to Internet resources during seminars and workshops allows them to find information instantaneously, including answers to the topic of discussion. In turn, it motivates the students in a certain way to work actively and participate in discussions.

Thus, the interaction between students and teachers through EMEP increases the degree of processed material assimilation by simultaneous conjunction of different work types and information perception methods.

Within the research we questioned 200 students of Kherson State University in the specialty of "Informatics", "Software Engineering", "Primary Education", "Physical Education", "Human Health" and "Chemistry" in order to identify their level of information competence at the beginning of the 1st semester.

We should note that modern universities of Ukraine already implement the information technology into the educational activity from the first semester. For example, the necessity to use Microsoft Office by the students of almost of specialties emerges from the first class, while the level of the given technology and training courses using ICT does not meet the necessary requirements. The research results show that only 11 % of the1st course students of the specialties "Human Health", "Physical Education" and "Chemistry" have sufficient knowledge on how to use programs for general purposes and ≈ 2 % of programs for educational purposes, while having the need to use them in the educational process in a percentage of 100 % and 26 % of students respectively, and teaching a course on using information technology started most of them only from 3rd–6th semester.

As shown in the Table 2, the information technology has become an integral part of studying, social and private people's life. The main and most important sharing information means is a global Internet network now which is constantly being updated, expanding its space, offering many new services and ways of getting information. These services are social networks and different computer communicative means which in our time have a significant impact on the students.

Social networks include websites that allow you to find business contacts, friends and partners. The dominant features of social networks are: connection without spending money; large number of users; contingent of like-minded users (in a group); set of supporting services, which help each user to create his/her own information space.

Table 2. Quantitative indicators of ICT use by students in the educational process

Specialties	Programs for general purposes		Programs for educational purposes		Computer communi-cative means		Cloud technology means	
	Own (%)	Use (%)	Own (%)	Use (%)	Own (%)	Use (%)	Own (%)	Use (%)
Software engineering	60	100	4	95	96	60	2	60
Informatics	65	100	5	87	86	60	2	60
Physics	30	100	3	65	77	40	1	10
Mathematics	27	100	4	62	76	40	1	10
Primary education	16	100	2	72	67	10	0	5
Human health	10	100	2	27	65	5	0	0
Physical education	10	100	1	25	64	5	0	0
Chemistry	12	100	2	27	66	5	0	0

Success in assimilation of the given and processed information by the students depends not only on the method of its presentation but also on the quality of the proposed materials, the available clarifications and guidance for solving certain tasks, the specifics of the proposed tasks according to the specialty, and the initial level of students knowledge on the discipline.

Let's take pedagogical specialties as an example. Due to the rapid process of education informatization, there exists the increased need for informatization and pedagogical staff who provide not only by the introduction of information technology into the educational process and their use at the class and independent students' works but also the introduction of new courses that would allow increasing informative and scientific and research competence of future teachers.

While introducing new disciplines and expanding the range of disciplines to choose for specialties, it is common that little attention is paid to the overriding of tasks for independent and classroom students' work, making their performance difficult and uninteresting and, as a result, does not motivate; conversely, it hinders the educational process.

For the development of practical skills and assimilation of theoretical ones during seminars, it is appropriate to use a combination of different methods of organization for students' teaching and learning activities, incorporating stimulating methods for the motivation of their education, taking into consideration the personal characteristics and educational and qualification profiles of participants [21, 22].

4.5 Forming Motivation for Students to Use ICT in their Professional Activities

It is important to assess the role of ICPE not only on teachers but also on students. Feedback is important for relevant studies, allowing evaluation of not only students' independent and classroom work, but also the work of teachers, departments and even the university as a whole. For example, the existence at KSU of "KSU Feedback" enables the evaluation of educational activities with such parameters as teacher's punctuality, student evaluating objectivity, student interest, student motivation, student self-assessment of his/her residual knowledge, course material compliance of the proposed tasks, material complexity ratio (for classroom and independent work), comprehensiveness of the educational material, audience interest, the use of modern information and communication technology by a teacher, and the clarity of the presented material.

The key indicators that allow the evaluation of the quality and necessity of the use of information technology in organization of individual and classroom works may be:

- Use of a modern information and communication technology by a teacher (Q1)
- Evaluation of residual knowledge by a student (Q2)
- Clarity of the represented material (Q3)

For example, comparison of Q1 with Q2 and Q3 enables us to observe the qualitative and quantitative changes in the implementation of new technology and methods into the modern education system.

Getting the parameters is a possible way of monitoring the annual students' surveys based on specific criteria:

1. Determination of the time interval (for example, for the given service it is possible to monitor the conducted surveys from the 2010 thru 2013 academic years)
2. Concerning changes in the curriculum and the introduction of new author's programs and courses, it is advisable to choose disciplines that were held in a specified time period for the same age group and at the same work program

As the relevant part of disciplines is not taught during the whole course of study by students at the university, we must consider the obtained data for the disciplines of the specialties "Informational Science" and "Software Engineering" for each course, which work programs have not undergone major changes during the period from the 2010 thru 2013 school years. Also, it was taken into account the results of the evaluation of the same student group in different courses.

As a result the following parameters were obtained:

- Evaluation of Q1 (use of modern information and communication technology by a teacher) in 2010–2011 academic year was 7.21 points (out of 10 possible)
- The lowest mark came from the index Q3 in the same year (clarity of the represented material), whereas in the 2012-2013 academic year, these figures increased to 24.3 % and 34.5 %, respectively (Fig. 4)

Interestingly, the fact of low assessment levels of residual knowledge by students (Q2) and the clarity of the presented material (Q3) in 2010–2011 academic year, while

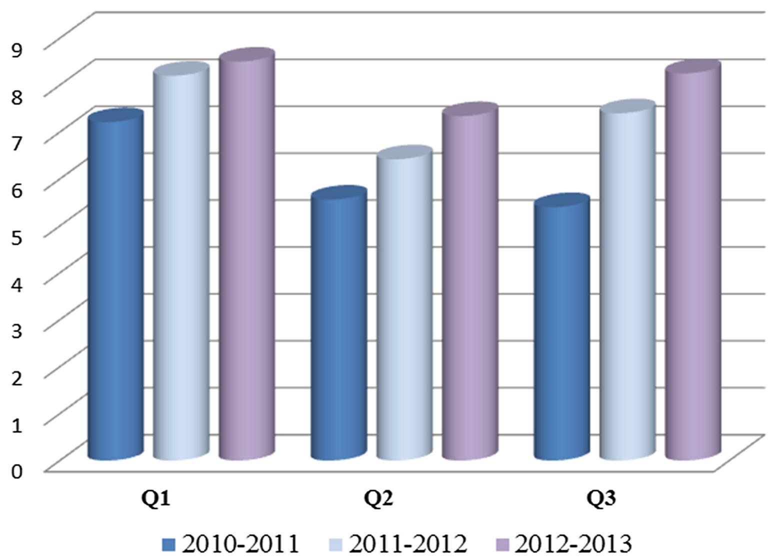


Fig. 4. Changing dynamics of the studied parameters as for 2010–2013 academic years

the rate of use of modern ICT by teachers (Q1) got fairly high marks. According to respondents, the main reasons for this were:

- Insufficient readiness of the teaching staff to use modern information technology
- Inconsistency of the curriculum with the new educational standards
- Insufficient basic level of information competence by students
- Low level of hardware and software

5 Discussion

In this section we represent The Concept of Competence, Informative Competence Components, Structure of teachers ICT competence and ECDL Certification.

5.1 The Concept of Competence

Over recent years, the concept of competence has been widely discussed among member states of the Organization for Economic Cooperation and Development (OECD), which conducted in-depth research on this issue. The International Board of Standards for Training, Performance and Instruction defines *competence* as the ability to carry out qualified activities and perform subsequent tasks/work (International Board of Standards for Training, Performance and Instruction (IBSTPI)). In order to facilitate the competences evaluation process, the Board proposes emphasizing the following indicators: the acquisition of knowledge and skills, the attainment of satisfactory levels, and educational achievement [16].

According to J. Raven, competence should also refer to the special abilities of a person, those required to perform a particular action in a particular subject area, which includes highly specialized knowledge, skills, ways of thinking and a willingness to answer for their actions. In the works of A.G. Bermus, an emphasis is placed on a system unity that integrates personal, substantive and instrumental peculiarities and components.

The competence approach shifts the focus to the process of building specified knowledge and skills. Additionally, emphasis is placed on achievement in the sphere of students abilities to form and develop practicality and creativity, using newly-acquired knowledge and experience in different situations. This requires the teacher to shift the emphasis in their teaching activity from the informational to the organizational and management plane. In the former case, the teacher plays the role of "knowledge repeater", while in the latter - the organizer of educational activity. The model of student's behavior also changes - from passive acquisition of knowledge to a research-active, independent and self-educative activity. The learning process is filled with a developing function that becomes an integrated learning feature. This characterization must be formed during the learning process and includes the knowledge, skills, attainment attitude, experience and behavioral patterns of the individual.

The competence approach is an approach that focuses on educational outcomes, and, as an outcome, it should be judged by the amount of learned information, facts, data, and the person's ability to act in different problematic situations. Among the key competences, which are defined as a guideline for determining the effectiveness of the educational process in Ukraine, the academic, social, competency in information and communication technology, economic, general cultural, social and health should be highlighted.

Thus, information competence is considered as a constituent in the structure of professional competence.

Accordingly, the functions of information competence are:

- Cognitive (knowledge-informative)
- Pragmatic (technologic)
- Practical

According to [18], each of the components holds a set of values which are filled with specific professional (teaching-representative), educational, communicative, and value content (Fig. 5).

Cognitive (knowledge-informative) provides definitions and facts of topical areas, provides scientific characters and innovation based on provisions proven by science and through practice; it takes into account the content variability of educational needs, a people's motives at different stages of their professional development. Pragmatist (technological) provides the formation of the skills of professional and household activities in the field of information technology. It provides mastery of the pragmatic components necessary for the ability to solve standard and non-standard problems in industrial processes using information technology. Practical is aimed at mastering how to use information technology - the ability to use information technology to solve problems that are relevant and of practical significance to the professional activity of individuals and society.

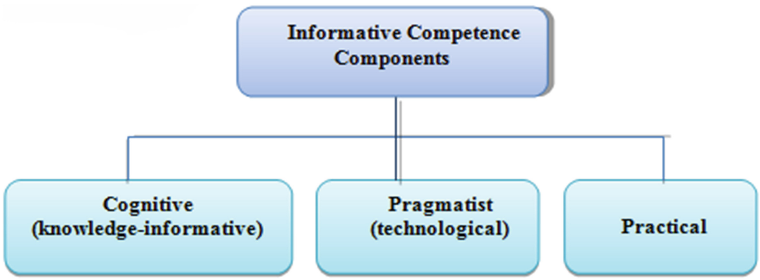


Fig. 5. Informative Competence Components

In the context of the widespread use of modern computer technology in the educational process, the requirements for teacher training significantly increase. Besides the basic knowledge necessary for the implementation of educational activity, the teacher needs to learn the basics of working with ICT, to be able to use information and communication technology to achieve certain educational goals, and to develop new organizational forms of educational activity [19].

The fruitful cooperation between UNESCO, CISCO, INTEL, ISTE and Microsoft in 2011 resulted in the document, "Structure of ICT teachers' competence. UNESCO recommendations" (Fig. 6) [17].

UNESCO recommendations emphasize that it is insufficient for the modern teacher to be technologically literate and be able to form appropriate technological skills by the students. Modern teachers should also be able to help students to use ICT in order to collaborate successfully, to solve problems that arise, learn the skills of teaching, and eventually become full citizens and employees. Thus, the recommendations affect all aspects of teachers' work. Among them: an understanding of the role of ICT in education; curriculum and assessment; pedagogical practices; ICT hardware and software; organization and management of the educational process; and professional development.

The recommendations take into consideration three approaches to the informatization of schools that are associated with the respective stages of teachers' professional development and mastering ICT in the rich educational environment. The first approach – "ICT Implementation" - requires the ability of teachers to help students use ICT to improve the efficiency of educational work. The second – "Knowledge Absorption" - requires the ability of teachers to help students in deep learning of the content of academic disciplines and to apply the acquired knowledge to solve complex problems that occur in the real world. The third – "Production of Knowledge" - requires the ability of teachers to help students, future citizens, and employees to generate new knowledge, which is necessary for the harmonious development and prosperity of society.

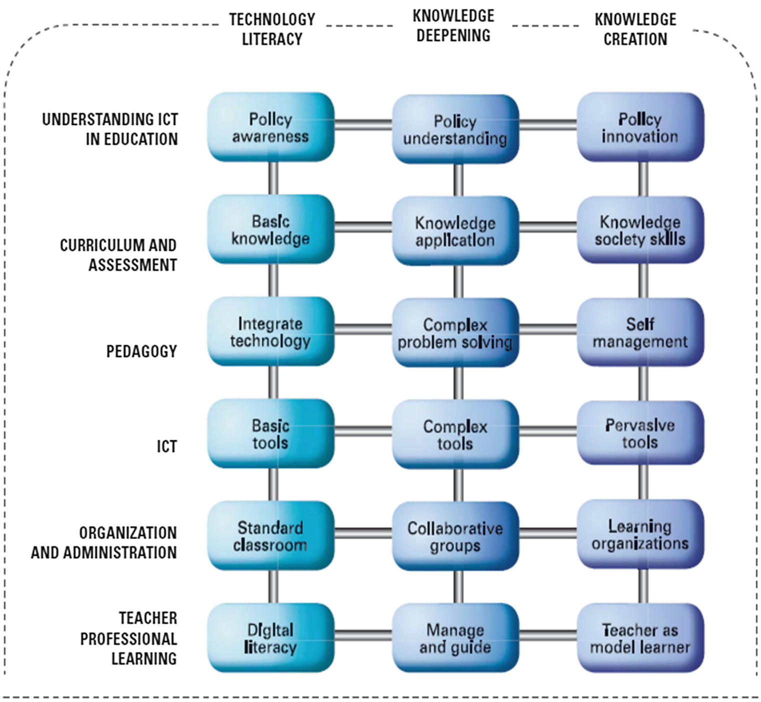


Fig. 6. Structure of ICT teachers competence

The recommendation states that teachers should adopt the methods and organizational forms of academic work, which meet the knowledge requirements of a developing society. Students should be able not only to master the content of the proposed academic disciplines but also to understand how they can produce new knowledge using the potential of modern ICT tools [17].

5.2 ECDL Certification

ECDL project harmoniously fits into the Bologna process allowing establishing a transnational system of knowledge evaluation. Participation in international certification program enhances the prestige of the Ukrainian education and recognition of IT competences of Ukrainian students abroad. The presence of the international certificate by the students increases their mobility and enables them to participate in student exchange programs.

ECDL certification is fully consistent with the “Action program”, which opens approaches to the implementation of the Bologna Declaration in the system of higher education and science in Ukraine, which refers to the need for a reasonable system development for determining the level of competence of higher-education-institution graduates and a method for the development of the objective assessment of specialists’ competence in certain educational and qualification levels in Ukraine.

International ECDL certification implementation is a confirmation of the educational quality. Working with the ECDL program, the university gets opportunity to prove the quality of its courses in information technology with international standards.

Thus the main objective of the university, which is aiming to implement and use ECDL course, should be:

1. The formation of the requirements for teachers’ and students’ training to use a course, and namely the definition of the required quantity and quality indicators of technical, scientific, and teaching methods, taking into account the initial level of competence, industry orientation, etc.
2. Academic staff training to use appropriate resources in the educational process (teachers’ retraining, conducting research and teaching seminars and workshops using ECDL program)
3. ECDL program passing by students (processing of theoretical and practical modules, testing)
4. Providing opportunities for online communication such groups as “student –student” and “student – teacher” on the ECDL Portal of universities

So, since 1999 ECDL is an official standard for evaluating computer skills in the educational system in Italy. During this time, more than 640,000 students in 2000 schools and 200,000 students from 200 universities have been trained and certified at ECDL.

6 Conclusions

Based on the analysis of scientific sources, we have defined the essence of the basic concept of the research. The definition of information competence is treated as competences associated with the growth of the information society; mastering of this technology and understanding of their application, weak and strong sides, critical judgments in relation to the information which is spreading by means of mass media and advertising.

As a part of the study, we have created the IT Competency Model. The relationship between the competency category and existing standards has been presented.

To form the basic IT competency, it is recommended to use the European Computer Driving License. This is documented by:

- International user experience of the ECDL standard
- The findings of the EU project TEMPUS TACIS "ECDL for Ukrainian Administrators"
- The findings of the project Ministry of Education and Science of Ukraine's "Creating the Internet Portal for Distance Learning ECDL for Higher Educational Establishments"
- The results of our research

Effective development of IT competencies among university students presupposes the following conditions:

1. The establishment of an ICT infrastructure at the institution (technical: computers and networks, services, software, support, information, and communication pedagogical environment)
2. Pedagogical, didactic, and psychological readiness of teachers to use ICT in their professional activities (the teacher should possess forms of educational activity organization in IC educational environment and apply innovative methods for facilitating the creative development of students, ideally using IC learning tools)
3. Use the ECDL international program to create the basic level of students' IT competency in the first year of study (Competence increasing project in the field of information and communication technology ECDL is the leading global certification of computer users which has been widely spread in Ukraine due to the support of the European Commission. Issued international certificates are recognized in over 150 countries. As of 2013 the network has more than 24,000 test centers which have certified more than 13 million people. Basic IT competency is the requirement for further effective teaching in special disciplines).
4. The systematic use of innovative teaching and ICT in the study of academic disciplines (student and teacher are equal subjects in the IC learning environment)
5. The formation of positive motivation in students to use ICT in their professional activities (the formation of motivational methods in professionally-oriented tasks, success, recognition of success, importance of a collective activity, feedback with teacher, self-assessment of activity results, mutual evaluation of student groups, use of innovative methods etc.)

The main aspects to be studied in further research are:

- The use of innovative methods for the formation of IT competency and the development of the creative abilities of students
- Organizational and methodical support for the certifying the participants of the educational process.

References

1. Savchenko, O.Y.: The Learning Abilities as a Key Competence of Secondary Education Competence Approach in a Modern Education: World Experience and Ukrainian Prospects: Library of Educational Policy. K.I.S., Kiev (2004)
2. Educational Technology in Higher Education. http://www.voced.edu.au/content/ngv22663
3. Joseph, L., Ng Lai, H., Ng Lai, L.: An analysis of students' preparation for the virtual learning environment. Internet High. Educ. **4**, 231–242 (2001)
4. Petukhova, L.E., Spivakovskiy, O.V., Voropay, N.A.: By evaluation of the interaction in the model "teacher-student environment". Sci. Educ. **4**, 401–405 (2011)
5. Kravtsov, H.M., Vinnik, M.O., Tarasich, Y.H.: Research of influence of quality of electronic educational resources on quality of training with use of distance technologies. Inf. Technol. Educ. **16**, 83–94 (2013)
6. Engage: Social Media Michaelmas. http://www.ucisa.ac.uk/~/media/Files/members/awards/amma/2012_AMMAentries/OxUni-ambermiro-award-engage
7. Guidelines for Using Social Media. http://www.provost.harvard.edu/policies_guidelines/Social_Media_Guidelines_FINAL_Version_1_0_effective_080112.pdf
8. Mykytyuk, A.M.: Theory and Practice of Scientific Research in Higher Educational Institutions of Ukraine in XIX century. Dis. Dr. Ped. Scii. Institute of Education Pedagogical Sciences of Ukraine (2004)
9. Petukhova, L.E., Osipova, N.V., Kushnir, N.O.: Actual problems of implementing ECDL course in the training of teachers. Inf. Technol. Educ. **8**, 17–22 (2010)
10. Spivakovsky, A., Petukhova, L., Spivakovska, E., Kotkova, V., Kravtsov, H.: Three-subjective didactic model. In: Ermolayev, V., et al. (eds.) ICTERI 2013, CCIS, vol. 412, pp. 252–273. Springer, Berlin-Heidelberg (2013)
11. Zymnia, I.A.: Key competences – a new paradigm of education result. High. Educ. **5**, 34–42 (2003)
12. ECDL Ukraine. http://www.ecdl.com.ua/
13. Italian Ministry of Education endorsing ECDL. http://www.ecdl.org/media/ecdlportal_italian%20schools.pdf
14. Osipova, N.V., Kushnir, N.O.: Peculiarities of creating of internet portal of distance learning ECDL for establishments of higher education. Inf. Technol. Educ. **4**, 157–163 (2009)
15. Kravtsov, H., Sharko, V.: Testing as the basic technology of the knowledge control of listeners of ECDL course. Inf. Technol. Educ. **8**, 39–46 (2010)
16. International Board of Standards for Training, Performance and Instruction. http://ibstpi.org/
17. UNESCO ICT Competency Framework for Teachers. ite.unesco.org/pics/publications/ru/files/3214694(1).pdf
18. Petukhova, L.E.: Theoretic and Methodic Formation Principles of Informative Competences of Primary School Teachers. Dis. Dr. Ped. Sci. Odessa (2009)
19. Imber, V.I.: Pedagogical Conditions of Education Multimedia Use to Prepare Future Primary School Teacher. Dis. Cand. Ped. Sci. Kotsiubynskyi Vinnytsia State Pedagogical University (2008)

20. The Use of Social Networks for Educational Purposes, http://www.osvita.org.ua/articles/1762.html
21. Kukharenko, V.: Methods of contents curator. Inf. Technol. Educ. **16**, 100–107 (2013)
22. Social Networks in Russia, Summer 2013: Figures, Trends, Forecasts. http://habrahabr.ru/company/palitrumlab/blog/186422/
23. Kravtsov, H.M.: Quality monitoring system of university electronic information resources. Inf. Technol. Educ. **2**, 42–46 (2008)
24. Kovalenko, I.: Edgar Dale's Training Cone. http://igorkovalenko.com/2012/05/24/konus-obucheniya-edgara-dejla/
25. Kolesnikova (Osipova), N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Basics of Information Technology, Kherson (2007)
26. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, OM.: Information and Communication Technology. Work at Computer. Basic Operations of File System Management, Kherson (2007)
27. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Texts Processing, Kherson (2007)
28. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Spreadsheet, Kherson (2007)
29. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Databases, Kherson (2007)
30. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Presentation, Kherson (2007)
31. Kolesnikova, N.V., Kozub, N.O., Kushnir, N.O., Melnichenko, O.M.: Information and Communication Technology. Information and Communication, Kherson (2007)
32. Kolesnikova, N., Kushnir, N.: Analysis of Previous Knowledge of Listeners and ECDL Tests Passing. In: The First International Training Conference "ECDL for Ukrainian State Officials", Kherson, pp. 52–53 (2007)
33. Kolesnikova, N., Kushnir, N.: Results Analysis of ECDL Test Passing. In: The Fourth Research and Practical Conference "Education Informatization of Ukraine. ICT at Higher Educational Institutions", Kherson, pp. 24–26 (2008)
34. Fedorova, I.B., Osipova, N.V., Siedov, A.O.: Organizational Maintenance of Distance Learning for the Higher Pedagogical Education. Inf. Technol. Educ. **5**, 65–71 (2010)
35. Official Website of Ministry of Education and Science of Ukraine. http://www.mon.gov.ua/

Future and Experienced Teachers Should Collaborate on ICT Integration

Nataliya Kushnir(✉), Anna Manzhula(✉), and Nataliya Valko

Kherson State University, 27, 40 Rokiv Zhovtnya St., Kherson 73000, Ukraine
{kushnir, valko}@ksu.ks.ua, ilovetrees@mail.ru

Abstract. School teachers of the future are coming from a new generation. This generation is fluent and even unseparated from digital devices. As representatives of a new culture, future school teachers should play a major of implementing ICT in education. Unfortunately, schools are still traditional and during Curricular Practical Training school mentors can't share their ICT in teaching findings, because they have little or none of such experience. Thus, future teachers should be ready to develop new approaches in education and help other teachers of elder generations to take advantages of it. University educators and the authors of ICT policies should provide opportunities for future teachers to communicate with their elder colleagues as peers and form their ability to discuss practical issues about ICT in Education. To meet these needs teachers from Kherson State University have organized series of joint ICT workshops for school teachers and students of pedagogical specialties. There are many benefits from such learning format, both for students and teachers. Teachers can learn more about ICT and share their experience, get inspiration and find affinity group, to become mentors for younger colleagues. In such groups, students can communicate with more experienced colleagues, find out more about teaching methods from experts, develop a better ability to express their vision, defend their ideas, help others to create teaching content with ICT, and be proactive as professionals. The results of such learning format, typical trends and difficulties are outlined and analyzed in the paper.

Keywords: Generation net · ICT training · Teaching with and through ICT · Young profess

1 Introduction

Future school teachers ARE representatives of a new generation. Schools and universities have met new generation of students. ICT is a mere modern context for teachers of previous generations, but it is an essential part of modern student reality. Special characteristics of modern students are formed by heavy use of digital devices. Twenty learner characteristics typical for present generation students have been listed in the work of Ronald A. Berk [1]. They are the following: technology savvy, relaying on search engines for information, interest in multimedia, creation Internet content, operating at twitch speed, learning by inductive discovery, learning by trial and error, multitasking on everything, short attention span, visual communication, craving social

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 217–237, 2014.
DOI: 10.1007/978-3-319-13206-8_11

face-to-face interaction, emotionally open, embracing diversity and multiculturalism, preferring teamwork and collaboration, striving for lifestyle fit, feeling pressure to succeed, constantly seeking feedback, thriving on instant gratification, responding quickly and expecting rapid responses in return, preferring to type. The results of our previous research have established the changes in students' learning style and students' attitude to ICT in recent years. According to new generation learner characteristics, we have developed the strategies for an ICT training course construction and appropriate teaching methods [5]. The results of university students and teachers feedback after the ICT training course have shown the efficiency of the strategies. Future school teachers are representatives of new generation and a new culture. However, despite the fact that they can learn in a new style, they have no good examples how to teach in a new way (Fig. 1).

Fig. 1. Child's world of information

ICT in Education is a part of modern world, not all the rage. ICT integration in education is not a tribute to the "digital" fashion. "A key trend that characterizes the environment beyond the classroom is the exponential growth in information and knowledge. As a consequence, students need to develop information literacy and other related skills to search for information from these seemingly unlimited sources on the internet, to evaluate this information and to select wisely from it" [9: p. 10, 14]. The teachers of previous generations oppose or remain passive to ICT integration in education. However back in early last century, scientists, psychologists and educators dreamt and described technologies that in their point of view could make significant progress in teaching of children and their mental development. "In the mid-1960-s, long before the appearance of personal computers, at the time when a single computer occupied one or several big halls, Papert and Feurzeig 13 were pondering on the idea that computers should be at children's disposal so that they could learn in a different way" [7]. The brightest example is Alan Kay's idea who forestalled the analog of modern iPad. In 1968 Alan Kay wrote and made sketches of a computer for children, "which he described as 'a portable interactive personal computer, as accessible as a book. *The Dynabook* would be linked to a network and offer users a synthesis of text, visuals, animation and audio. Kay drew an initial pen and ink sketch of this device, which is widely considered the prototype for the notebook computer, but in fact is much closer to today's iPad" [7].

These words are very close to a phrase that was reportedly spoken by Thomas Edison in 1913. Edison pioneered the development of machines for displaying motion pictures, and he was confident that these devices would be used extensively to help to teach students. The well-known *Wizard of Menlo Park* was asked to speculate about the future: "Books," declared the inventor with decision, "will soon be obsolete in the public schools. Scholars will be instructed through the eye. It is possible to teach every branch of human knowledge with the motion picture. Our school system will be completely changed inside of ten years". The interview article was part of a series of stories in the newspaper about the "Evolution of the Motion Picture" [15]. As we see, Edison has made a mistake in timing, but the main idea about the changes in Education comes true. *Technology has already dramatically changed the role of a teacher.*

At the modern stage of ICT development the words of Ray Clifford (Associate Dean, College of Humanities, BYU, USA) have taken on a new lease of life: "*Technology will not replace teachers but teachers who use technology will probably replace teachers who do not… We are not going to replace teachers, but we can help teachers – so let's begin now to address the challenges of the future*" (1987) [16].

The level of technology development allows teachers to bring their creative potential to educational process through ICT. Teachers can make notes, social bookmarks and blogs online, and have documents and websites at their disposal. Such services as Wiki, YouTube and SlideShare make possible online collaboration and participation, as well as a presentation of teacher's results and work and a collection of the feedback from all over the world. There are also a lot of resources that involve different types of information (text, video, audio, chart, mind maps, etc.). Smart and efficient use of such services inspires young professionals to implement such innovations in education. It helps teachers to create and organize their assignments quickly, provide feedback efficiently, and communicate with their classes with ease. It also makes them feel them more confident and work up their own teaching style.

The problems of ICT training of future teachers are in the focus now. One of the UNESCO reports presented two components of ICT training for teachers: LEARNING about ICT and TEACHING with and through ICT and stages that schools typically pass through ICT adoption and use: Transforming, Infusing, Applying, Emerging [9: p. 30–34, 57] (Fig. 2).

Learning about ICT has gained widespread popularity. *How to use ICT in subject teaching* is a part of every Teaching Methods course that is taught to students of pedagogical specialties. *Teaching with and through ICT* is a key component for ICT integration is based on *Applying productivity tools*, *Enhancing traditional teaching*, *Facilitating learning using multi-modal instruction* and *Creating and managing*

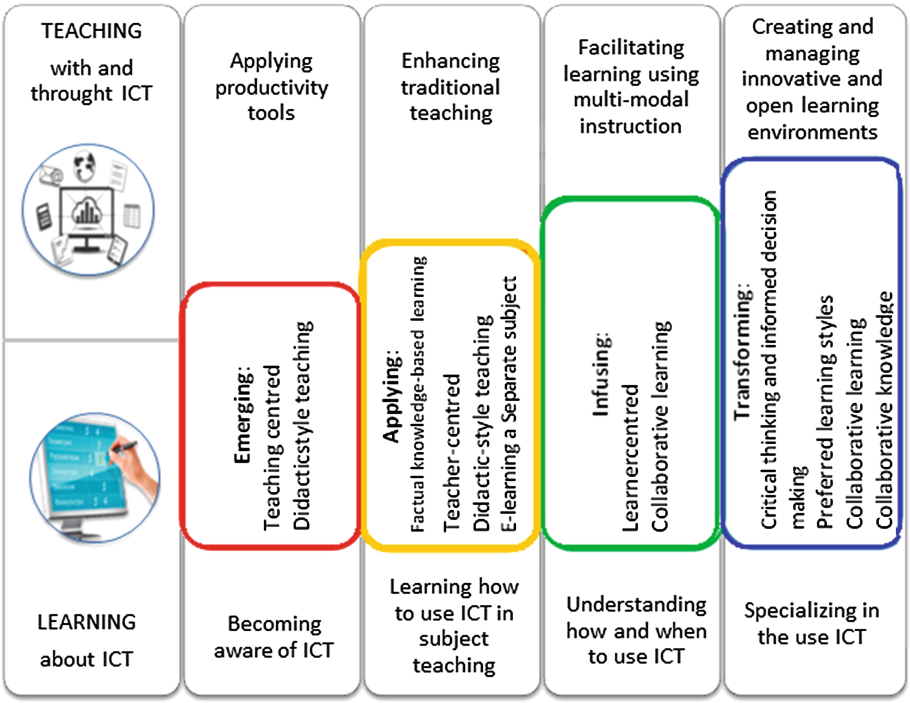


Fig. 2. Stages of ICT integration in Education

innovative and open learning environments. Ukrainian educational system has stuck on the level of *Applying productivity tools* and *Enhancing traditional teaching*, but it needs further development.

The next step *Facilitating learning using multi-modal instruction* assumes various forms of collaboration with visual, audial, tactile digital interfaces. The technologies that support multimodality only started to come into educational environment. As an example we can take a look at the set of services that is developed by Google. There are many Google resources and applications that can be utilized in the classroom. They have developed a great amount of applications and services for recognition and retention of texts, images, and sounds. Now, project Google Classroom is in the planning stage to realize a communication between different services for running applications (downloads), collaboration and research resources. For example, it allows create 3D drawings and models. These drawings (if done to scale and look similar to buildings) can be placed on Google Earth as a 3D image, etc. [17].

To cope with such great amount of information, applications and services and to feel comfortably in such environment can only a digital native, but for a teacher of previous generations it is very hard. Their negative attitude towards technological innovations, force Ukrainian Education not to go further than enhancing traditional schooling. There is no wonder that effective teaching methods usually used are not as good as they were few years ago and are not well aligned with new generation of students.

Innovation is always contrasted with tradition. Our schools remain still traditional. The problem of antagonism between tradition and innovation is classical in philosophy. Now it takes a new significance in education. A well-known piece "Structure of scientific revolutions" (by T. Kuhn) is devoted to the problem of gaining new knowledge and correlation between the tradition and innovation. It is essential from our point of view that Thomas Kuhn has made an emphasis on the determination of a scientist by traditions of scientific community he belongs to [18]. We can also draw a parallel with teaching stuff community.

Educational system is stagnant. There are high expectations from teacher's activities and from quality of knowledge the students obtain. Pedagogical experiments can not be made impulsive and rash, because it will influence the students. This great responsibility assumes implementation of traditional approaches in educational process. New generation of teachers live in the established culture of pedagogical stuff. Teaching stuff communities are conservative in their teaching methods and tools. This results in a great tension. Novices in teaching practice do not have enough time and energy for deep analysis of their own activity and professional development, and implementation of new approaches to their practices.

Young teachers – new culture beams – go to schools where everybody teaches traditionally. Young teachers are tech savvy and share learning style with their younger students, though coming to University as students and then to school as teachers they are surrounded by traditional educational environment. In every subject teaching discipline students learn how to use ICT for pedagogical purposes, but basically ICT are considered only as a tool to enhance a tradition schooling. Unfortunately, university teachers of previous generations are also not ready to transfer educational process into innovative and open environment.

The educational system, from kindergarten to university in its present state, is insufficiently flexible for the modern world requirements. Considering education as developing science in the Kuhn's model we would obtain switching to another "gestalt" and the change of roles of pedagogical subjects in education. Nevertheless, teachers can not switch their style so easily.

It is important to develop new approaches in education that can facilitate transition to information society and reach high level of ICT integration in education. ICT have a great potential in forming new relations between subjects of educational process. But there is the problem, the absence of continuity and balance between innovation and tradition. Young professionals usually have no time to implement and develop new approaches. They spend all their energy for mastering traditional techniques and approaches, thus ICT remain just a support for traditional schooling.

During curricular practical training school mentors can't share their ICT in teaching experience, because they have little or none of such experience. Desire to communicate with people of the same profession and to share common professional interests is natural for everybody who desires to become a whiz-bang. At universities, future teachers are involved in learning process and to act accordingly to their roles of being students. They communicate actively and share their experience about exams, grades, and homework. They do not consider themselves as future professionals. They pay no proper attention to their progress in getting new teaching experience and increasing their professional competency. University teachers and authorities need to regard needs of students in accordance with their future professional background.

A bottom line in forming students' practical skills during a university study is their quasi professional activity that is developed through Curricular Practical Training. The University authorities organize and hold Curricular Practical Training off-campus. It is hosted by the best municipal schools. The curricula assume the following kinds of practical training for all students of pedagogical specialties:

- 2–3 year students – Curricular Practical Training (passive). Students are ordered to schools where they should oversee experienced teacher's work. While they are at this kind of practical training they must organize out-of-class cultural and education activities as playing with children during the breaks, help teacher to manage children in the canteen and during excursion, write psychological characteristic for pupils, short-hand notes and analysis of the lessons, organize and perform one cultural event for children and their parents. During this training students adopt teacher's methods, get acquainted with best practices of managing children group, a structure of lesson plan and planning other school activities. The duration of this practical training is 2 weeks.
- 4–5 year students – Curricular Practical Training (active). Students become teacher's assistants and actively get involved into educational process. The core activities of this training are conducting lessons under the teacher's control, organization of cultural events, working with recordkeeping (making records in a class-book, correcting exercise-books, analysis of curricula and lesson plan). The duration of this training is 6 weeks for 4 year students and 9 weeks for 5 year students.

Thus, Curriculum of pedagogical specialties include more than 4 months of practical training at school (at schools and at universities - for Master's program), but despite of variations of pedagogical trainings at school (passive and active practical trainings, training "The first days at school") and a lot of positive examples of teachers and students collaboration during training there is no special opportunity for students and teachers to discuss ICT in education. Usually teachers are bound up in solving current problems and fulfilling their day-by-day duties. They try to teach interns how to manage ordinary school challenges. There is no opportunity for students to get an experience of teaching with and through ICT.

Young teachers should be ready to develop new approaches in Education at schools and help other teachers to take advantages of it. Even young professionals - yesterday graduates of pedagogical institutions - have difficulties with implementation of innovative approaches. In their attempt to fit the norms and traditions of school staff, sometimes they can't stand up for their vision and sacrifice it for the sake of social mimicry and, as a consequence, adopt the dominant teaching style (Fig. 3).

We share a view that future teachers – today's students of pedagogical specialties – and experienced teachers should be ready to collaborate in implementation of ICT in teaching practice. During the transition period to the new educational paradigm it is necessary not only to develop new approaches, but also to motivate future teachers to bring a new teaching style to schools. Schools and universities make attempts to implement new approaches, but different generations of teachers react to innovation

Fig. 3. The word cloud created by future and practicing teachers "What kind of person is a modern successful teacher"

differently. The aim of our research is to find new types of interaction and organization of collaboration between future and experienced teachers. In our opinion, it is especially actual when one considers ICT in education from different generational perspectives. It makes possible to save pedagogical tradition and to enrich it by innovative approaches. Every generation can contribute. For example, teachers can share their knowledge in teaching methods drawing on their expertise and experience, enclose methods of teaching with ICT to real school's conditions. Students can contribute new ideas and "fresh" ways of solution to "old" problems, create electronic didactical materials quickly and easily, keep up the atmosphere of openness and collaboration, delegate more teachers' functions to ICT. The role of the university teacher is to organize such forms of communication that assume efficient share and communication between teachers of different generations.

2 Related Work

We found support and inspiration in the ideas of Ukrainian scientists Lvov and Spivakovsky who study the problems in ICT training for future teachers [8] as well as from publications of the foreign researchers. Thus, the authors of the article "Oceans of innovation. The Atlantic, the Pacific, global leadership and the future of education" [10] also emphasize the importance of innovation. They highlight the pressing need of new culture of communication in overall life: "Accelerating the pace of innovation to meet this challenge will require great diverse cities, great universities, great new and established businesses and extensive interactions between them. It will depend on individuals who are open to ideas and argument and who are part of teams in which vigorous debate, dissent and discomfort exist. It will require a culture of openness – to argument and ideas, experts and outsiders, the young and the new". This idea is also applicable in education, both experienced teachers and young professionals or students should be able to communicate and collaborate as peers, to feel themselves confident, to express ideas freely and think critically in the cause of science.

As we know, from the clash of opinions emerges the truth. Johnson carries his point about efficiency of team work and demonstrates that the "…most important ideas emerged during regular lab meetings, where a dozen or so researchers would gather and informally present and discuss their latest work. If you looked at the map of idea formation that Dunbar created, the ground zero of innovation was not the microscope. It was the conference table." [11].

Barber, Donnell and Rizvi briefly describe the style of communication in creative working group that inherited new culture that is sufficiently close to that model of interaction we have attempted to recreate at workshops: "The teams that work well get along but are not too comfortable with each other. They challenge and question each other. Dissent is welcomed. Moreover, they are not all experts. Some are outsiders who bring a different, fresh perspective to bear and force the experts in the group to come up and out of the professional rabbit holes they are so fond of diving down. There are many ways to build diversity into a team, but it doesn't happen by accident. An opportunity to bring diversity to bear that is often missed is to bring young people into positions of leadership". It also makes a great significance to invite experienced professional, a mentor in a creative working group. In a report examining the key success factors of internet start-ups we have found that: "Startups that have helpful mentors, track performance metrics effectively, and learn from startup thought leaders … have 3.5x better user growth" [12].

Teacher in transformation of Education plays a critical part. We are of the same point of view with K. Robinson's that the question about whether innovative creative impulse will be delivered to children directly depends on inner organization of school system [20]. One more example [13], recent studies in Peru and Thailand have shown the failure of 'one child, one laptop' policies to make any impact on learning. It is critical that technology incorporates best practice pedagogy and is integrated systematically into the learning process. Michael Fullan, in [14], published "…developing innovative responses to the current challenges … consists of integrating three components: deep learning goals, new pedagogies, and technology… But the key is that the teacher takes a highly proactive role in driving the learning process forward, using whichever strategy works for a specific student or task (and analyzing which strategy works best). In the new pedagogies, this means interacting with students to make the students' thinking and questions about learning more visible."

Hence, teacher should lead ICT implementation at classroom, discuss problems he/she faces with the colleagues as well as share the findings. "There is growing evidence that technology can and does influence outcomes for the better, not when it replaces the teacher, but when – alongside an active, motivating teacher ('teacher as activator' as John Hattie describes it 109) – it reinforces, extends and deepens students' learning opportunities. As this unfolds not only will the content of the curriculum have to change, but also the organization of the learning day and, above all, the way in which students are assessed".

Barber, Donnell and Rizvi also admit a gap between conceptual frames, strategies and decisions in ICT and teachers' current challenges that makes a difficulty in education transformation: "…while education reformers are seeking to design a system for 20 years ahead, teachers struggle with the present and parents remember the system of 20 years ago: the conceptual gap is therefore 40 years – a major communications challenge which governments and educators often underestimate. You could argue that the gap is even bigger than this, given that school students of today will still be part of the global workforce 50 years from now". As a solution, Barber, Donnell and Rizvi have worked up several key conditions for innovation that all researches in ICT in education field should consider: "Innovation model cuts across four levels of civil society: people, teams, organizations and culture. Innovation requires, first of all, people with the right skills and attributes. First, innovation requires, first of all, people with the right skills and attributes. Second, these individuals must be selected and combined into effective teams. Third, organizations should be structured to be cross-functional and have fluid organizational roles. Finally, society must furnish a culture that is progressive and open to the transmission of new ideas, welcoming of diversity and rules-based".

Innovative approaches that appear recently are in sharp contrast with the previous ones, so there is a strong need to develop a number of strategies that can help to bring new vision of young professionals to teaching practice and make them adequate to the present. There is also a strong need to support young professionals in implementation of new approaches to practice and connect educators of different levels: students, young professionals, experienced teachers and teachers of pedagogical universities.

3 The Analysis of Young Professionals Polling

We research the problem of digital competence forming [4] in order to estimate the experience in ICT training of the future teachers. Investigations of different aspects of this problem have assured us to redesign our ICT course and to change all its components of educational process (content, organizational forms, teaching methods, etc.) [5]. We researched specialties of modern students generation, compared the data gathered over the years and results of ICT training for students of different specialties. We fully realized efficiency of our improvements and new strategies that were developed earlier.

Monitoring ICT integration at schools we have found out that students that deal with ease and appreciate innovative teaching methods are not able to implement them in their professional practice. The results of diagnostic polls of young teachers and analysis of related works led us to highlight difficulties of a young teacher and stimulated us to research the problem of interactions between young and experienced teachers in details. For this purpose, the system of curricula profession training was analyzed and series of interviews with young teachers was conducted. Then we determined the level of interest among school principals and teachers regarding cooperation in ICT field. During the presentation in front of school teachers we have underlined specialties of modern students -representatives of new generations and tendentious in ICT development. On this stage we have found several facts, e.g.

administration of some schools expressed interest in training school students in their preparation toward competitions and mentoring students' research projects. However they objected the necessity of additional training for school teachers that is beyond governmental program. Another interesting fact is that both teachers and students consider themselves as experienced ICT users. That is the reason why stuff of some schools is skeptical of the idea of joint ICT training. Nevertheless, it's important to note that:

- Teachers have some skills in ICT (e.g., they can surf the net and search for information, create presentation in MS PowerPoint and work with interaction board), consider themselves as experienced users of ICT, fail to realize that many applications and technologies get older earlier than teachers begin to use widely them.
- Students have access to unlimited resources and use them for personal needs as making a photo of their meal or themselves… or to buy something online. Only a small number of students use ICT for learning and personal development beyond home tasks.

We fully agree with opinion of Bennett, S., Maton, K., and Kervin, L. who "research into how to design games that foster deep learning is inadequate": education requires effort. Nevertheless, it is also true that modern multimedia world should offer students more than just a paper book to get knowledge out. "This evidence points to differences in the ways young people use technology inside and outside of school, and suggests that school use of the Internet can be frustrating, but there is little basis to conclude that these differences are causing widespread and profound disengagement in learning. Rather, they tell us that technology plays a different role in students' home and school lives." [21].

The joint ICT training was developed and held at municipal schools where administration was interested in such cooperation. After trainings online polls were conducted among participants of seminars (teachers and university students).

3.1 The Results of Young Teachers Polling

Teachers of municipal and rural schools in Kherson region were surveyed. The questionnaire contained three types of questions that assume one-choice answers, assessment on a scale from 1 to 5 and free-answers. They were asked to rate their knowledge and skills on a five-point scale they have acquired during pedagogical internships and university study. Among the characteristics were self-discipline, recordkeeping, the actual knowledge of the subject, teaching methods, communication skills with children, communication with school staff, team-building and children's collective management, the organization of the educational process. The lowest ratings for all characteristics were got from teachers with 2–3 years working experience. Teachers that work the first year gave average scores (Fig. 4).

Teachers with 5 and more years of working experience gave sufficiently high scores in all categories. It can be explained by the fact that these teachers have sufficient teaching experience. It is easy to notice that evaluation made by teachers of different

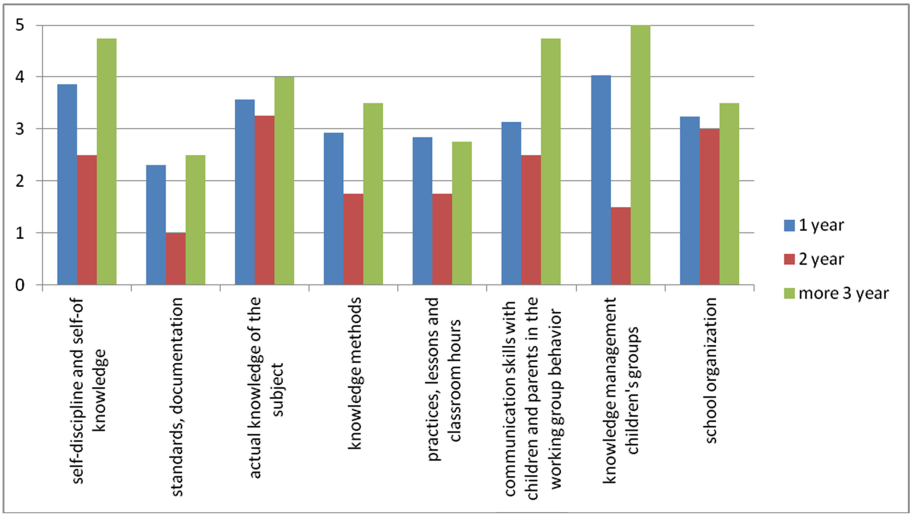


Fig. 4. The level of knowledge and skills development at university graduation time (results of teachers' self-assessment)

experience is arranged synchronously. However, there is a large discrepancy in the results of their behavior in school community. One has to pay a special attention to changes in teachers' self-esteem as the years go by. They are depicted on the chart with a large "spread" (Fig. 4). That means that teachers that have been working for 2 years have difficulties in communication with school stuff, their ideas don't get enough respect and attention. That showed that young teachers feel uncomfortable and unconfident. Under such conditions implementation and realization of creative approaches in education, including new approaches based on ICT use, have low probability. Therefore university teachers should prepare future graduates to be ready for implementation their vision and ideas.

The answers to the question about the problems that arise in young teachers' work in the early years of work can be divided into three categories: methodological (lack of teaching skills, frequent changes in the documentation standards), educational (disciplinary matters, relations with parents and children), technical (insufficient computer equipment).

As you see, near 30 % of respondents noted that they need more special equipment for teaching as computers, screens, interactive boards, etc. So we can conclude that they believe that ICT in teaching is something vital.

Also the teachers were offered to choose the solution of the problems that appear in their professional activities. Mostly respondents noted their pro-active attitude to make changes in Education (their answers assume active actions) (Figs. 5 and 6).

In our recent research we have studied the motivation of students' study at a university and have obtained that 56 % of students are oriented to reach academic achievements, rest 44 % defined motives that are related with the status of teacher and manager of children's group. Though, it was important to research changes in attitudes

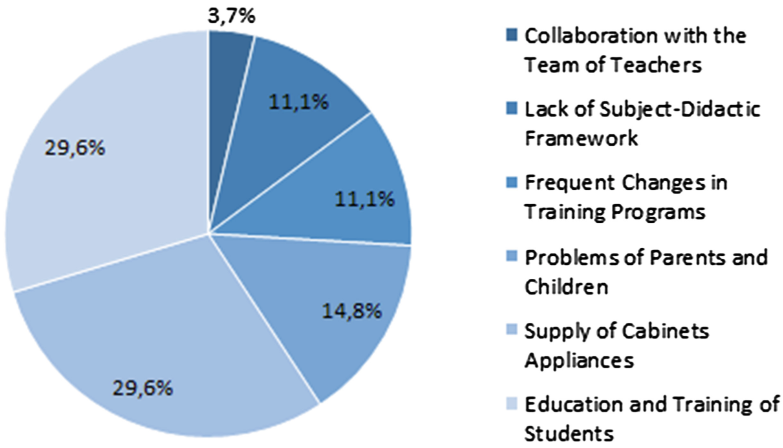


Fig. 5. What problems often encountered during the first years of work?

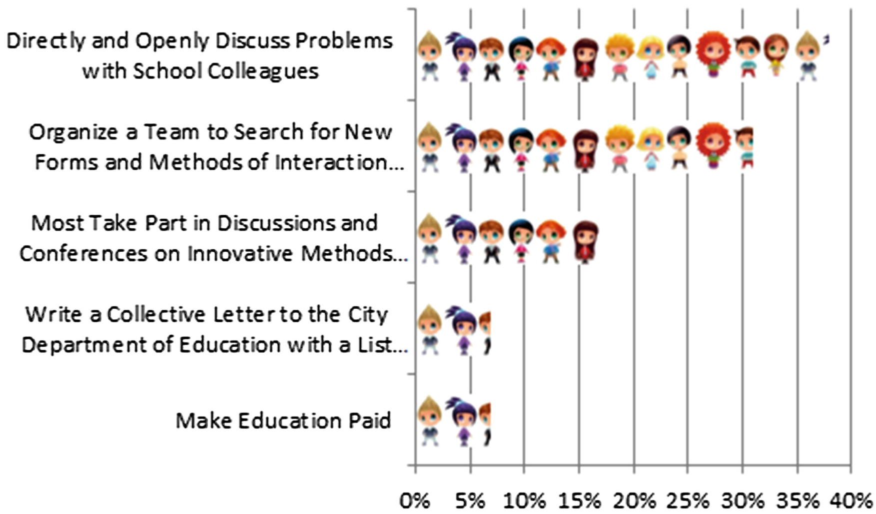


Fig. 6. What activities will help improve the learning process?

to pedagogical work when students become teachers. The respondents were offered to choose one pair of “smiles” (Fig. 7).

As depicted above, 69 % of teachers have satisfied their expectations and considerable part of respondents (20 %) changed their attitude to the work at school for better and only 11 % of professionals admitted crease in their attitude.

The surveyed teachers were also offered to write short recommendations for future teachers. Most of the recommendations are related to communication and collaboration with their elder colleagues. There are following recommendation were suggested and mainly include learning how to communicate, how to take an experienced colleagues’

Fig. 7. The fragment of the questionnaire and the dynamics of shifting the attitudes to the teaching profession

advice, never stop to learn new things, getting a joy and inspiration from your professional activity and interaction with children, making students interested in learning, attending other teachers' lessons, working on own teaching style", etc.

Thus, the analysis of young teachers' polling shows that during their University study they get appropriate theoretical preparation at sufficient level and seek for practical experience and communication with school staff.

4 Transition to Joint ICT Training Workshops

The sufficient hours of Curricular Practical Training do not provide adequate practical preparation in ICT in education and give no bright examples of teaching with ICT at schools. We tried to find out how to create such educational environment that would provide a better understanding of future professional challenges, but would also be sensible for students' influence and modeling. We also find out that it is important to form students' readiness to realize their vision about education, to share it with their elder colleagues, to discuss and work out a joint stance with them. As a solution joint ICT training workshops for school teachers and students of pedagogical specialties were proposed.

4.1 The Structure of the Workshop

Joint workshop includes *learning about ICT* and *teaching with and through ICT* components, assumes execution of tasks in mixed (students and teachers) and unmixed groups, discussion the classroom situations where ICT can be used efficiently and working Teaching methods with ICT out. The most effective work can be organized in a learning group of 14–20 participants (the equal quantity of teachers and students) with support of following equipment: 10 computers or similar devices connected to the Internet, printer and projector.

In the design of workshop we included efficient organizational forms, tasks, and teaching methods that satisfy educational needs of students based on new generation characteristics as technology savvy, interested in multimedia, creation Internet content, learning by inductive discovery, craving social face-to-face interaction, emotionally open, embracing diversity and multiculturalism, preferring teamwork and collaboration, constantly seeking feedback, responding quickly and expecting rapid responses in return, etc.

We paid special attention to group forms of organization of learning (discussions, research projects, round table discussions and etc.), because this generation learns by sharing. Students crave for real interaction and are ready to make their contribution. They express emotions, demonstrate their attitudes and create new things together.

We also used special tasks to realize learning strategies that are suitable for this generation, for example:

- **Interested in multimedia, "visual" communication.** The tasks are designed considering students' interest in "visual communication" and multimedia. Video clips and presentations are added. The tasks and manuals contain minimum of text information and maximum of graphics and illustrations.
- **Multitasks on everything.** A teacher encourages students to use search engines during the course and work with Google services as Google.Drive, Google.sites, etc. This satisfies the interest in creating online content and allows simultaneously to work with several documents. It's also an efficient tool for organization of group activities.
- **Retention in the profession.** Organization of students' activity is a process of constant modeling of professional activity of specialists under learning conditions. The students of teaching specialties create educational games, tests, documents, tables that are useful in their professional practice. The results of the work are evaluated not only by technical element, but also by educational one.
- **Teamwork and cooperation.** Some tasks should be complex and presuppose collaboration, while their execution reinforces the cognitive interpersonal communication and interaction of all participants. An important stage is to prepare a learning group to work together. To ensure about students' readiness for cooperation a teacher should arrange Ice braking, team-building and communicative exercise (up to 5 min) at the beginning and at the end of lesson.
- **Emotionally open.** Tasks should be person-centered and presuppose creation unique results that will describe student's own lifestyle, attitude, etc. This approach helps to decrease the quantity of plagiarisms and contributes to the development of their creative abilities.

The organization of joint activity of students and teachers assumes several stages. The first one is aimed to *establish a contact and create positive atmosphere*. Before announcing the plan of workshop it's important to establish contact between students and teachers. Usually we use a pedagogical technology "Icebreaker". Both students and teachers are sufficiently constrained in the beginning at workshop and "Icebreaker" allows relieving the tension and improving communication. *Motivation and problem statement* follows next as traditional elements of ICT training workshop (Fig. 8).

Organization of joint actions aimed to execution tasks by ICT tools. At this stage, it's important to organize work of teachers and students in tandem. Teachers' expertise and

Fig. 8. The stages of workshop

experience help formulate and justify the use of ICT tools to improve classroom teaching. Students as Net Generation representatives supply technical support and tasks that assume intensive use of ICT. The results of joint activities should be *presented and published* (Table 1).

Table 1. The stages of the workshop

Stage	Description	Technology
Connect people	The format of joint workshop assumes trust, open attitude, freedom to express participant's ideas and feelings	Icebreaking, using of positive symbols, working in the round, small groups
	This stage is the key for further concerted efforts, because teachers and students are unfamiliar, the coordinator – University teacher – should to create a positive atmosphere and convert classroom in a friendly zone	
Topic, Goals, Format, Plan	Preparation to acquire new information, "exposure" a workshop plan to coordinate participants' activities and make participants ready for rapid switching from one activity to another, tracing their own progress	Problem solving tasks; contribution of every participant

(*Continued*)

Table 1. *(Continued)*

Stage	Description	Technology
Motivation	Going "deeper" in ICT and Teaching Methods problems for different school subjects. Teacher has more complete idea about conditions of possible ICT use in educational process. Teacher makes a pedagogical situation more realistic, elaborating details (school subject, topic and learning activity). Hence, students can get closer to Teaching Methods in modeled pedagogical situation at workshop	Brainstorming (work in groups), discussion
ICT learning and teaching with ICT	Instruction before work in pair	Demonstration how the service or technology works by university teacher
	Execution of task by mixed pair of participants (teacher - student) that assumes creation of pedagogical content and a fragment of lesson demonstration (2–3 min), where this content is used	Splitting the group into pairs "teacher-student", managing the work in pairs, organizing presentation of results
Reflection	The analysis of obtained results by participants, getting feedback for university teachers	Polling

4.2 How Can Such Workshops be Organized?

According the government standard of elementary general education, letters and orders informatics as obligatory discipline is added to a curriculum from the second grade of elementary school. Respectively governmental frames and decisions more ICT disciplines were included to curricula of future elementary school teachers. The practicing teachers completed special ICT training on a mandatory basis to be ready to teach school subject "Stairs to Informatics" in Municipal Institution of Higher Education Kherson Academy of Lifelong Learning attached to Kherson Council 36 h - short training and 24 h - distant variant of the course. This way both, students and teachers, get additional training in ICT in education. The program of the ICT training partly coincides with the program of ICT discipline for future teachers that gave us opportunity to organize joint ICT training for future and practicing teachers in a form of workshop.

ICT laboratory chief of MIHE Kherson Academy of Lifelong Learning also insists on the cooperation of students and practicing teachers. He describes following problems in ICT training for school teachers: "Many teachers that are near the retirement age and do not consider using ICT during theirs lesson at schools as a helpful tool, because they are used to teach traditionally. Teachers that are used to be competent, to know more than students feel constrained and unconfident dealing with computers. There is no stimulus for active self-development. Many teachers explain this by absence of additional funding, but self-development can not be bought. Also teachers are not used to work in groups. It is difficult for them to share their knowledge and materials and also to perceive new ideas". In joint ICT training, teachers can learn things together with students, to consult with students, to take more open position and to start communication and exchange. "Students contribute special atmosphere to the work - they are opened, express their emotions easily, inherent with positive thinking and attitude".

For a university teacher such joint ICT training is an opportunity to reinforce students' motivation. During the workshop students study together with practicing teachers, who will use all technologies "tomorrow" and their interest in ICT learning is the best argument of practical necessity of ICT use at school. Also it is possible to organize communication between students and teachers for better understanding the real situation at school. Experienced teachers can make their contribution to development of the teaching with ICT, describe real pedagogical situations of their use, to diversify tasks and types of didactical materials, consult and discuss ICT in education topics.

Experience of joint ICT training was successful. Under agreement about Kherson State University and MIHE Kherson Academy of Lifelong Learning the first series of joint ICT future and practicing teachers was held on the topics: "Using online services to support the educational process in primary school", "Interactive presentation in elementary school", "Printer serial in MS Word", "In-class interaction by means of online services", etc. in 2013. These topics were chosen in the borderland between students' and teachers' training programs. In 2014 we have organized such workshops that were hosted by municipal schools.

Students and teachers gave a positive feedback about joint ICT training. Its results after such workshops showed that both teachers and students highly appreciated the format of the event and work in mixed pairs. The level of the psychological comfort was evaluated by participants as Good – 12 % and Excellent – 88 %. The characteristic of the psychological state of students and teachers also is presented in the form of clouds of words (Fig. 9).

Teachers spoke positively about "learning new interesting and useful services, communication with students, a good visualization, creative approach, opportunity to share experience, the spirit of goodwill and cooperation, positive atmosphere, high competence of university teachers". Teachers noted the following problems during the workshop: some technical problems and lack of computer user skills. Teachers also made a proposal "to organize such meetings more often".

Students liked the following: it was "easy to work and get a good result", they also appreciated knowledge sharing, collaboration and communication with teachers, opportunity to work with a teacher of other specialty, opportunity to get acquainted with professional distinctiveness of a teacher, excellent mood and atmosphere, the tea party and the people. Among the problems at workshop students identified technical problems with the computer as the low speed of the Internet. The students that took part in the workshop expressed a wish to attend such events often, though the workshops

Fig. 9. Word cloud associations about the workshop (teachers' one is in the left and students' one is in the right)

were organized in extracurricular time and didn't promise obvious bonus at university like a mark.

In organizing such a workshop university teachers met with some difficulties. Still there is no adequate and simple mechanism for workshops organization and future cooperation between students of pedagogical specialties and practicing teachers. This event is a personal initiative of university staff and students, as well as school stuff of MIHE Kherson Academy of Lifelong Learning and municipal schools' stuff.

5 Findings

Future school teachers are representatives of a new generation. They inherit characteristics that were formed under conditions of modern world that is full with ICT and digital devices. These characteristics assume special style of learning and format of its organization. The same format fits for younger students – school pupils.

Future teachers should become initiators of ICT implementation in learning environment. Unfortunately, schools are still traditional and during Curricular Practical Training school mentors can not share their ICT in Teaching findings, because they have little or none of such experience.

We found out that young teachers have difficulties in their professional adaptation at schools, especially dealing with teaching stuff. There is a wide range of reasons, overloading with paperwork, social authority of experienced teachers, constraints to emulate traditional teaching style.

Thus, future teachers should be ready to develop new approaches in education and help out other teachers of elder generations to take advantages of it. University educators and the authors of ICT policies should provide opportunities for future teachers to communicate with their elder colleagues as peers and form their ability to discuss practical issues about ICT in education. Naturally as "digital natives" future teachers can lead or at least make their contribution solving problems of ICT implementation. Nevertheless time for Education transformation has come, there is no reason to object all teaching experienced that have been gained during many years.

All mentioned above became a ground for organization of joint ICT training for school teachers and students of pedagogical specialties. There are many benefits from such learning format both for students and teachers. Teachers can learn more about ICT and share their experience, get inspiration and find affinity group, to become mentors for younger colleagues. Students can communicate with initiative and experienced people of the same profession, to find out more about Teaching Methods from experts, to form an ability to express their vision and to stand out for their ideas, to help other people to create teaching content with ICT, to be proactive professionals.

An efficient structure of such joint workshop assumes several stages: to establish a contact among participants, to inform topic, goals, format and plan of workshop, to state a problem, to organize brainstorming and discussion under school teacher expertise, to demonstrate how technology or service works, to split group is mixed pairs and manage their tasks execution, to help groups to present their results and fragment of the lesson, to sum workshop up.

We foresee the prospects of implementation this form of work at Kherson State University in the future. In our view, practical classes in pedagogical institution have to change under instrumental and constructivist approaches in education. Methods of work in pedagogical institution should be revised and supplemented by the work of the invited expert. Is also necessary to hold classes at schools, provide workshops, etc. Teacher behavior should be changed from relying on information and communications environment as an active subject of learning. There should be a way to allow students to share their own discoveries, thoughts, and to manifest themselves actively. Otherwise immersion in a professional environment for the student is not possible. Only the active forms of work can "awaken" in the brightest students' masters. Many students chose the teaching profession after a successful experience of dealing with children. We should provide students with the opportunity to communicate with children and to attach them to the professional aspects of school life while studying at university. This approach involves the discussion. We expect constructive comments and suggestions from colleagues.

6 Prospects

In general, we plan to popularize our idea about such format of ICT training for experienced and future teachers. We plan more university teachers and other educators as well as school principals to get involved in workshops and their organization to help obtain findings for teaching with and through ICT that were worked out along the workshops; to prepare students for creating and managing innovative and open learning environments, to make them ready to become initiators of technology, and to express their ideas and stand for their vision.

In September 2014 we will work with several municipal schools where the administration expressed interest in cooperation. We will form experimental students-teachers teams for long-term collaboration and research and organize ICT training events and investigation of their efficiency in forming new culture of interaction and communication between young and experienced teachers.

References

1. Berk, R.A.: Teaching strategies for the net generation. Trans. Dialogues: Teach. Learn. J. **3** (2), 1–23 (2009)
2. Geer, R., Sweeney, T.A.: Students' voices about learning with technology. J. Soc. Sci. **8**, 294–303 (2012)
3. College Student's Perceptions of Libraries and Information Resources: A report to the OCLC Membership. Online Computer Library Center (OCLC), Dublin, OH (2006)
4. Kushnir, N., Manzhula, A.: Formation of digital competence of future teachers of elementary school. In: Ermolayev, V., et al. (eds.) ICT in Education, Research, and Industrial Applications. CCIS, vol. 347, pp. 230–243. Springer, Heidelberg (2013)
5. Kushnir, N., Manzhula, A., Valko, N.: Bridging the Generation Gap in ICT Education. In: Ermolayev, V., et al. (eds.) ICTERI 2013, CCIS, vol. 412, pp. 229–251. Springer, Heidelberg (2013)
6. Poll results of Gorshenin Institute (Ukraine, 2011). http://institute.gorshenin.ua/researches/den_uchitelya_v_ukraine.ht
7. Recognizing the Potential of ICT in Early Childhood Education. Analytical Survey. UNESCO Institute for Information Technologies in Education, IITE/INS/AS/2010/01 (2010)
8. Lvov, M.S., Spivakovsky, A.: The concept of teaching science subjects at school and university teacher. Comput. School Family **3**, 21–25 (2003). (in Ukrainian)
9. ICT TRANSFORMING EDUCATION. A Regional Guide. Bangkok, UNESCO (2010). http://unesdoc.unesco.org/images/0018/001892/189216e.pdf
10. Barber, M., Donnelly, K., Rizvi, S.: Oceans of Innovation. The Atlantic, the Pacific, Global Leadership and the Future of Education. Institute for Public Policy Research (2012). http://www.bath.ac.uk/education/events/documents/Barber_et_al_x2012x_oceans-of-innovation.pdf
11. Johnson, S.: Where good ideas come from: The natural history of innovation. Riverhead (2010). http://zgm.se/files/Books/Where_goog_ideas_come_from.pdf
12. Marmer, M., Herrmann, B.L., Dogrultan, E., Berman, R. Startup genome report: A new framework for understanding why startups succeed. Startup Genome (2011). http://gallery.mailchimp.com/8c534f3b5ad611c0ff8aeccd5/files/Startup_Genome_Report_Extra_Premature_Scaling_1.56.pdf
13. Let them Eat Tablets: Trying to Stop the Rot in Thailand's Schools by Giving out Tablet Computers. Economist (2012). http://www.economist.com/node/21556940
14. Fullan, M.: Commentary – the New Pedagogy: Students and Teachers as Learning Partners. Learn. Landscapes **6**(2), 23–29 (2013)
15. Smith, F.: The evolution of the motion picture: VI – 'Looking into the Future with Thomas A. Edison'. N. Y. Dramatic Mirror **9**, 24–24 (1913)
16. Clifford, R.: The status of computer-assisted language instruction. CALICO J. **4**(4), 13–16 (1987)

17. Google Classroom. https://sites.google.com/site/googclass/
18. Kuhn, T.S.: The Structure of Scientific Revolutions. University of Chicago Press, Chicago (1962)
19. Behrstok, E., Clifford, M.: Leading Gen Y Teachers: Emerging Strategies for School Leaders. TQ Research&Policy BRIEF, Washington, DC (2009)
20. Robinson, K.: How Schools Stifle Creativity. CNN Opinion. http://edition.cnn.com/2009/OPINION/11/03/robinson.schools.stifle.creativity/index.html
21. Bennett, S., Maton, K., Kervin, L.: The digital natives' debate: A critical view of the evidence. Br. J. Technol. Educ. **39**(5), 775–786 (2008)

Efficiency Assessment of Computer-Oriented Learning Environment of an Institute of Postgraduate Pedagogical Education: Factors, Criteria, Characteristics

Kateryna R. Kolos(✉)

Institute of Information Technologies and Learning Tools,
9, M. Berlynskogo St., Kiev 04060, Ukraine
porcelyana5@gmail.com

Abstract. This article provides the rationalization for a comprehensive assessment of computer-oriented learning environment efficiency of an institute of postgraduate pedagogical education and provides the definitions of the 'efficiency of computer-oriented learning environment of an institute of postgraduate pedagogical education', and the 'assessment criteria for computer-oriented learning environment efficiency of an institute of postgraduate pedagogical education'. In order to measure the efficiency of computer-oriented learning environment of an institute of postgraduate pedagogical education, a multi-criteria factor evaluation model was created. It includes, in the first place, the factors defining computer-oriented learning environment of an institute of postgraduate pedagogical education. The article provides the rationalization for and describes the developed criteria of efficiency assessment in such an environment and divides them into corresponding criterion characteristics as well as establishes the numeric values of coefficients of importance of all characteristics, criteria and efficiency factors of computer-oriented learning environment of an institute of postgraduate pedagogical education.

Keywords: Efficiency · Assessment · Factor · Criterion · Characteristic · Computer-oriented learning environment of an institute of postgraduate pedagogical education · In-service teacher training · Information and communication technology

1 Introduction

The need for drastic changes required to improve the quality and competitive ability of education in the new economic and socio-cultural environment, the acceleration of Ukraine's integration into the international education environment demands the introduction of modern information and communications technologies (ICT) that improve learning process (LP), the accessibility and efficiency of education [25].

The task of efficiency assessment of learning process, management decisions, teaching techniques, etc., the functioning of an educational institute in general, in terms of the use of information and communications technologies is partly accomplished during the procedure of licensing and accreditation. These procedures at higher

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 238–260, 2014.
DOI: 10.1007/978-3-319-13206-8_12

educational establishments (HEE) are regulated by the fundamental documents on education policy in Ukraine, for example [4, 5, 19–21], etc. These documents require the conformity of performance characteristics of educational establishments to accreditation criteria.

Computer-oriented learning environment of an institute of postgraduate pedagogical education is a controlled and deliberately created artificial space intended for unfolding the learning process by the means of ICT that features all sufficient conditions required for the purpose of effective professional development of pedagogical staff [14].

Within our research we consider computer-oriented learning environment (COLE) as a computer-oriented learning environment of institute of postgraduate pedagogical education.

The research goal is to determine the principal factors, criteria, and characteristics, as well as to explain the efficiency evaluation system of COLE.

2 Research Methods

In order to achieve the abovementioned goal, the following research methods have been used: theoretical (analysis of philosophical, psychometric, methodological, specialized literature on the issue under investigation as well as regulatory documentation on the teaching process organization at higher educational establishments and the development of education in Ukraine; identification, analysis, and systematization of the criteria and characteristics of efficiency assessment of COLE); empirical (interviews with students of an institute of postgraduate pedagogical education; direct, indirect, and participant observation of learning process of teaching staff during in-service teacher training in COLE).

Expert evaluation method was used to estimate the numeric value of coefficient of importance of every criterion characteristic of COLE efficiency. A questionnaire [13] was developed on Google Forms basis. This questionnaire was used in an expert evaluation experiment. The subjects have been selected among the most qualified instructors in ICT-related disciplines, coming from the following educational establishments: of Vinnytsia Regional Institute of Postgraduate Pedagogical Education, Donetsk Regional Institute of Postgraduate Pedagogical Education, Zaporizhia Regional Institute of Postgraduate Pedagogical Education, Municipal Higher Educational Establishment Zhytomyr Regional Institute of Postgraduate Pedagogical Education, Municipal Higher Educational Establishment Vasyl Sukhomlynskyi Kirovohrad Regional Institute of Postgraduate Pedagogical Education, Luhansk Regional Institute of Postgraduate Pedagogical Education, Teacher Resource Complex Institute of Postgraduate Education of the National Technical University of Ukraine Kyiv Polytechnic Institute, K.D. Ushynskyi Chernihiv Regional Institute of Postgraduate Pedagogical Education.

It should be mentioned that group evaluation can only be considered sufficiently valid when the opinion consistency among all the experts is very good. Therefore, statistical processing of data received from the experts also contains the evaluation of opinion consistency among the experts, which was measured using the concordance coefficient W [2].

Based on the determined coefficients of importance of characteristics ($V_{p_{yji}}$), the coefficients of importance of the corresponding criteria ($V_{k_{yj}}$) and factors (V_{f_y}) of efficiency assessment of COLE were calculated (see Table 1), where indices y, j, i indicate the relation of various parameters to a corresponding factor, criterion or characteristic.

3 Basic Notions and Definitions

First of all, to conduct efficiency assessment of COLE, the notion of 'efficiency of COLE' should be defined.

Dictionaries say that the notion 'efficiency' means 'something that produces an effect (result)' [23], 'something that brings required results, outcomes, has the biggest effect' [8], 'successful in reaching the set goals' [18].

A.M. Dolgorukov says that 'efficiency' is a judgemental category which is a subjective measure that can vary with the change of our judgments [9].

Amitai Etzioni believes that 'efficiency of an organization is defined by the level of realization of its goals' [1].

R.L. Daft stresses that 'efficiency' is a broad notion, the use of which requires the consideration of defined variables that characterize individual departments and the entire establishment. Work efficiency correlates with the internal environment of an establishment, which means that its past achievements should be taken into account while setting new goals and defining future strategic course [7].

V.H. Kremen` and V.Yu. Bykov provide a rationalization for measuring 'learning environment efficiency' by the degree of correspondence between the quality and quantity characteristics of the created or planned learning environment and the specific target functions according to which it was designed and continues to evolve [16].

Therefore, the term 'efficiency of computer-oriented learning environment of an institute of postgraduate pedagogical education' means a judgmental category of COLE, that is defined by the level of realization of its tasks leading towards the main goal of the system of postgraduate pedagogical education and is conditioned by the key characteristics of individual structural components and COLE in general.

According to dictionaries, term 'factor' (from German faktor, Latin factor - the one who acts, creates) means cause, driving force of any change, event [10] or process that defines its nature or certain features [6].

Since the key task of COLE is creation and provision of the required and sufficient conditions to the students of in-service teacher training courses facilitating their personal and professional growth, the efficiency assessment of COLE should be carried out through evaluation of determinant factors of COLE, namely: (1) the efficiency of LP of in-service teacher training in COLE; (2) COLE infrastructure sufficiency; (3) ICT competence of COLE academic staff (research and teaching); (4) ICT competence of students.

The measurement of every COLE factor in its turn requires the institute of corresponding criteria for the assessment of COLE efficiency.

Term 'criterion' (from Greek kriterion, means of judgment) is defined in dictionaries as 'a feature, reason for evaluation taken as a basis of classification' [3]; 'a feature, sign, which serves as a basis for judgment, means of verification, measure of assessment; in the theory of cognition it is a feature that allows distinguishing between true and false and makes the judgment possible' [24].

In The Encyclopedia of Education, 'quality criteria of learning' is defined as 'features determining the degree of correspondence of learning to the set goals, standards and norms' [15].

The National Education Glossary of Higher Education defines the notion of assessment criteria regarding the compliance with educational and didactic requirements as 'descriptions of what should be accomplished for achieving learning results and at what level' [17].

Thus, the notion of 'criteria of efficiency assessment of COLE' signifies the descriptions of features of COLE, on the basis of which the efficiency assessment of the demonstrated quality of in-service teacher training provision in COLE is carried out.

A corresponding system of criterion characteristics is the manifestation of every efficiency assessment criterion in COLE at a certain development stage of COLE.

4 Criteria Factor Model

The need for objective efficiency assessment of COLE requires the use of qualimetry approach, which envisages comprehensive development and description of methodology of quantitative and qualitative assessment of COLE efficiency.

In order to perform this measurement, the criteria factor model of assessment of COLE should be designed (Table 1), which, in the first place, requires the selection of factors characterizing COLE, their criteria, and the determination of coefficients of importance of factors and criteria.

Decomposition of criteria determined within the criteria factor model of COLE efficiency assessment (Table 1) to the corresponding characteristics selected according to the didactic requirements of COLE [12] will allow to display qualitative and quantitative changes of COLE, to check the compliance and establish the connection between the levels of criteria, factors and characteristics of assessment and efficiency of COLE in general.

Decomposition of principal efficiency assessment criteria of COLE to the corresponding characteristics will be done factor by factor.

4.1 Learning Process

Let us pay a closer attention to the criteria characteristics based on the efficiency factor of learning process of in-service teacher training.

The basis of learning process of in-service teacher training in COLE consists of 'a system of organizational and didactic measures aimed at the realization of education at

Table 1. Criteria factor model of efficiency assessment of COLE.

Factor			Criterion		
#	Name	Coeff. of importance (V_{f_y})	#	Name	Coeff. of importance ($V_{k_{yj}}$)
1	Efficiency of LP of in-service teacher training at COLE	0.27	1.1	Goal corresponding	0.16
			1.2	Incentive motivation	0.18
			1.3	Organizational and practical	0.18
			1.4	Control and regulating	0.14
			1.5	Health-preserving	0.16
			1.6	Result and reflexive	0.18
2	COLE infrastructure sufficiency	0.26	2.1	Financial (expense and value-added)	0.45
			2.2	Fail-safe and backup	0.14
			2.3	Organization and management	0.14
			2.4	Information dissemination	0.12
			2.5	Assessment and reflexive	0.15
3	ICT competence of COLE academic staff	0.22	3.1	Quantitative	0.18
			3.2	Appropriateness	0.19
			3.3	Educational and qualification	0.19
			3.4	Research and practice	0.22
			3.5	Assessment and reflexive	0.22
4	Student's ICT competence	0.25	4.1	Motivational and axiological (value-conscious)	0.21
			4.2	Cognitive (knowledge)	0.21
			4.3	Praxeological (pragmatist)	0.21
			4.4	Academic	0.17
			4.5	Assessment and reflexive	0.20

a certain qualification level according to the state standards of education' [22] and modern prospective tendencies related, first of all, to the intensification of LP while using pedagogically considered information and communications technologies.

Therefore, the efficiency of learning process in general is estimated by the integrated characteristic of the system of organizational and didactic measures of LP in COLE aimed at the proper improvement of skills of the teaching staff.

First of all, learning process of in-service teacher training courses of COLE should take place according to the set goals and tasks of COLE. Thus, it is crucial for the organization of LP to estimate the number of students able to take an in-service teacher training course in COLE over an academic year. The key role during the process of academic group formation is played by the correlation between the number of potential students, based on the preliminary recommendation of town and district education boards, to the optimal capacity of COLE in terms of training a certain number of students on the in-service teacher training courses. Personal and professional features of the students, such as special field, qualification category, etc. should be considered. In accordance with this, the academic load is formed, and topics for classes, techniques and methods of successful LP and courses time frame are approved.

The efficiency of learning process in COLE also depends on its compliance with didactic requirements of COLE, namely [12]:

- creation of comfortable, healthy and life saving conditions facilitating the influence on the students' professional improvement;
- consideration of professional and personal features of students;
- general tasks of LP, development of professional competencies of students in COLE should comply with the goals and tasks of the high-school teaching process;
- creation of conditions for efficient realization, support and control of the unassisted work of students of in-service teacher training;
- creation of conditions for transition from the existing partial use of ICT to the efficient systematic application and development of COLE while using various forms of organization of LP;
- optimization of innovative approaches during LP of in-service teacher training;
- implementation and rational use of the credit-unit system to carry out interim and final monitoring of students' progress on in-service teacher training courses on the basis of state standards and with the consideration of special regional features of teaching work;
- rational use of distance education elements while organizing LP of in-service teacher training for full-time, part-time, or external studies, or a combination of such;
- integration of individual, micro-group, group, collective and mass organization forms of LP, the basis of which is provision of in-service teacher training courses in group form;
- transformation of the teacher (teaching methods specialist) as a source of knowledge into the knowledge manager, organizer, coordinator, consultant;
- creation and constant update of subject matter of COLE as well as providing students of the in-service teacher training courses with an access to the topical educational information and resources of professional interest.

These requirements determine basic characteristics of the goal corresponding criterion (1.1) of efficiency assessment of learning process of in-service teacher training in COLE (Table 2).

Table 2. Characteristics of goal corresponding criterion.

#	Characteristic	Coeff. of importance ($V_{p_{11i}}$)
1.1.1	The conformity of the number of enrolled students to the optimal number of them in terms of successful completion of in-service teacher training in COLE	0.30
1.1.2	The conformity of LP to the didactic requirements of COLE	0.35
1.1.3	Keeping to a unified pedagogical position with regard to the professional trend in the use of pedagogically balanced ICT in LP of in-service teacher training in COLE	0.35

LCP efficiency in COLE to a large extent is defined by the motivational level of in-service teacher training participants who want to master practical ICT skills.

4.2 Motivation and Incentives

Let us single out basic characteristics of the incentive motivation criterion (1.2) of efficiency assessment of learning process of in-service teacher training in COLE (Table 3).

Organizational and methodological work in learning process of in-service teacher training in COLE, as one of the basic factors of COLE efficiency, is determined, first of all, by adhering to the set course period. It characterizes the quality of organizational and methodological work planning. Particularly, it concerns the level of COLE standardized documents that determine organizational measures of LP of in-service teacher training. Among them, curriculum is the most important.

Table 3. Characteristics of the incentive motivation criterion.

#	Characteristic	Coeff. of importance ($V_{p_{12i}}$)
1.2.1	The facilitation of pedagogical conditions in COLE that motivate the participants of in-service teacher training to use ICT in LP	0.35
1.2.2	Management of events intended to acquaint the academic staff of COLE with innovations and new tendencies in ICT application in LP	0.30
1.2.3	Conducting of practical classes for the academic staff of COLE aimed at gaining and developing ICT skills	0.35

4.3 Characteristics of Organization and Tool Support

The correspondence of technologies, content, forms, and methods involved in LP of in-service teacher training to the planned ones determines the quality level of academic work, its justification, planning and implementation of learning process in COLE.

With the help of open access to electronic learning resources, such as libraries, professional editions and depository, etc. in COLE, the participants of LP are directed through information environment. They can choose the most suitable educational information, ICT means, etc. for further use in learning, cognitive and professional activities.

A significant amount of time in LP of in-service teacher training is dedicated to the student's individual work including project preparation, essays writing, and other tasks that require a clear and easy system of proper consulting and reviewing, especially during webinars, phone calls, textual and video communication, etc.

COLE academic staff should handle many tasks that require clear planning and periodic reports. They are generated by significant teaching, methodological, academic and organizational work that should be done by COLE academic staff. The staff also should quickly and thoroughly master the latest ICT means, assess their suitability for learning process of in-service teacher training and at a general education institute. For this reason, academic staff and other workers at higher educational institutes should prepare individual plans and make periodic reports on work they have done during school year. However, such type of planning and reports often repeat the information that can be found in different documents, particularly in the individual plan of the academic worker and their reports on the planned work, or combined work planned and done by the faculty, department or postgraduate pedagogical educational institute. This requires automation of reports on the accomplishment of educational, methodological, academic, and organizational work done by the academic staff of COLE. It is reasonable to organize this automation with the help of modern cloud resources offered by Google and Microsoft.

In such a way, the staff can cooperatively prepare documents, draw up and approve plans of educational activities like learning sessions, consulting hours, conferences, webinars, workshops, round-table discussions, etc., save video/audio reports on them, use file databases in further learning and cognitive activity.

So, let us single out the basic characteristics of the organizational and practical criterion (1.3) of efficiency assessment of learning process of in-service teacher training in COLE (Table 4).

4.4 Characteristics of Students Performance Control and Assessment Criterion

To improve the quality of learning process of in-service teacher training in COLE, it is necessary to automate entry-test-, ongoing-, final- and self-assessment of students with the help of dynamic and fully integrated ICT means for supporting LP and performing students' development assessment. Such measures will significantly reduce the time for assessment, and save it for learning and cognitive activity.

Teaching program and assessment management systems provide LP participants with the access to the students' performance ratings according to the credit-unit system. They also allow automatic tracking of students' attendance. Not only does it help gather group performance data, it also helps analyze them online. This will add more information while making decisions that can influence the performance of any student,

Table 4. Characteristics of the organizational and practical criterion.

#	Characteristic	Coeff. of importance ($V_{p_{13i}}$)
1.3.1	Compliance with organizational and practical data published on the COLE website referring to the periods during and between courses, namely: terms, in-service teacher training enrollment requirements, class schedules, etc.	0.20
1.3.2	Conformity of content, form, methods, and means to the curriculum	0.22
1.3.3	Open access to electronic learning resources of COLE: libraries (EPrints), professional editions (OJS), depository, etc.	0.20
1.3.4	Organization of the system of proper consulting and reviewing, especially during webinars (Moodle, WizIQ), VoIP phone calls, textual and video communication (Skype, Google Talk), etc.	0.20
1.3.5	Automation of reports on the accomplishment of educational, methodological, academic, and organizational work done by the academic staff of COLE	0.18

group, department, educational institute, as well as the whole system of postgraduate pedagogical educational institute.

This way, basic characteristics of control and regulating criterion (1.4) of efficiency assessment of learning process of in-service teacher training in COLE are determined (Table 5).

Table 5. Characteristics of control and regulating criterion.

#	Characteristic	Coeff. of importance ($V_{p_{14i}}$)
1.4.1	Organization of surveys of the participants of LP using ICT tools (like Google forms)	0.25
1.4.2	Automation of introductory, ongoing, exit and self-diagnostics of students	0.30
1.4.3	Organization of learning achievements assessment of the students of in-service teacher training in COLE according to the credit-unit system with the help of ICT tools, LearnBoost in particular	0.27
1.4.4	Automation of students' attendance control	0.18

4.5 Characteristics of the Health Preservation Criterion

Health is one of the vital resources of personal and professional development for every professional. That is why special role in efficiency assessment of in-service teacher training LP in COLE is allotted to health-preserving criterion (1.5). Its principle characteristics are provided in Table 6.

Table 6. Characteristics of health-preserving criterion.

#	Characteristic	Coeff. of importance ($V_{p_{15i}}$)
1.5.1	Organization of the instant response system in cases when participants of in-service teacher training LP in COLE have some health problems	0.25
1.5.2	Adherence to health and safety regulations during LP in COLE	0.35
1.5.3	Adherence to safety standards of ICT use in LP of in-service teacher training in COLE	0.40

4.6 Characteristics to Assess the Results of Learning

Learning process of in-service teacher training should focus on its participants. When efficiency assessment of learning process in COLE is conducted, it is reasonable to take into consideration the results of learning and cognitive activity of in-service teacher training and students' satisfaction of LP in COLE.

This way, the characteristics of result and reflexive criterion (1.6) of efficiency assessment of learning process of in-service teacher training in COLE are determined (Table 7).

Table 7. Characteristics of result and reflexive criterion.

#	Characteristic	Coeff. of importance ($V_{p_{16i}}$)
1.6.1	Improvement of students' professional competence during in-service teacher training in COLE	0.50
1.6.2	Satisfaction of students of in-service teacher training with LP in COLE	0.50

5 Infrastructure Efficacy Facet

Let us proceed to the criteria characteristics according to the COLE infrastructure sufficiency factor.

Infrastructure sufficiency of COLE is characterized by the available space provided for LP during in-service teacher training in COLE, as well essential equipment needed for teaching rooms, timely installation, scheduled and proper routine maintenance of ICT facilities, which is basis for determining the main criteria and characteristics of infrastructure sufficiency of COLE.

5.1 Characteristics of the Financial Criterion

Rapid development of ICT sphere and high practical value of new ICT technologies in learning process trigger the need of COLE for ongoing maintenance, renovations, repair and setting up of new ICT means, other equipment and devices required for teaching and work rooms in COLE. This, in turn, requires financial support. However, demands of COLE could not be satisfied because of low funding which is mostly provided by regional budgets. On the other hand, efficient functioning of COLE could partially cover these necessary expenses, particularly through enrolling some teachers for passing postgraduate courses on a fee basis, or through commercial use of ICT facilities, etc.

This way, basic characteristics of financial (expense and value-added) criterion (2.1) for assessing COLE infrastructure sufficiency are determined (Table 8).

Table 8. Characteristics of financial (expense and value-added) criterion.

#	Characteristic	Coeff. of importance ($V_{p_{21i}}$)
2.1.1	Budget allocations for the functioning, maintenance, and development of COLE	0.16
2.1.2	Total payments to COLE academic staff related to their direct professional work: wages, bonuses, travel expenses, etc.	0.17
2.1.3	Total cost of equipment in teaching rooms, offices, etc. used in COLE: ICT, furniture, etc.	0.15
2.1.4	Total cost of energy consumed by ICT means and its full payment	0.12
2.1.5	Expenses on scheduled and routine maintenance of ICT	0.15
2.1.6	Total cost of new ICT and their installation	0.15
2.1.7	Profits gained from COLE: students' training fees and commercial use of ICT means, etc.	0.10

5.2 Characteristics of the Fail-Safe and Backup Criterion

For the proper functioning of COLE, it is highly important to guarantee high-level protection. It is particularly vital to reach the highest level of safety for the integrity of in-service teacher training LP.

This way, the characteristics of fail-safe and backup criterion (2.2) for assessing COLE infrastructure sufficiency are determined (Table 9).

5.3 Characteristics of the Organization and Management Criterion

Improved efficiency of COLE requires carefully organized infrastructure management system of COLE. Such system will include all subsystems, communication lines, and processes facilitating efficient COLE.

It has determined the characteristics of organization and management criterion (2.3) for assessing infrastructure sufficiency of COLE (Table 10).

5.4 Characteristics of the Information Dissemination Criterion

The main feature of postgraduate pedagogical education system is an immediate reaction to the modern demands of new information society. This system should take

Table 9. Characteristics of fail-safe and backup criterion.

#	Characteristic	Coeff. of importance ($V_{p_{22i}}$)
2.2.1	Information security of electronic resources	0.24
2.2.2	Automation of regular external backup, reduplication, and deduplication (detection and deletion of obsolete data) of electronic resources: creation and retention of backup copies of important data on external devices, particularly on additional HDDs or using cloud services, etc.	0.22
2.2.3	Reliability assurance of all parameters of COLE within certain time frame that supports COLE functioning in set working conditions	0.27
2.2.4	Adherence to safety standards during ICT installation and use	0.27

Table 10. Characteristics of organization and management criterion.

#	Characteristic	Coeff. of importance ($V_{p_{23i}}$)
2.3.1	Management of the local COLE network	0.15
2.3.2	Management of personal (identity) records of the participants of in-service teacher training LP on the basis of ICT means	0.13
2.3.3	Automation of application preparation and students' registration for in-service teacher training	0.12
2.3.4	Automated analysis of introductory, exit and self-diagnostics	0.15
2.3.5	Organization of independent and interdependent students' activity with cognitive and psychological support, which is highly important during individual and distance forms of studies	0.14
2.3.6	Open access to modern pedagogically balanced ICT while organizing and conducting in-service teacher training	0.16
2.3.7	Development and support of distance courses based on the most popular e-learning platforms: Moodle, Claroline, ATutor, etc.	0.15

them into consideration while solving current and possible tasks of teaching and organizational processes. That is why the infrastructure of an institute of postgraduate pedagogical education should provide participants of in-service teacher training LP with free access to the wide range of high-quality modern teaching, learning, and educational aids. This way, the characteristics of information dissemination criterion (2.4) are determined (Table 11).

5.5 Characteristics of the Assessment and Reflexive Criterion

COLE infrastructure should create favorable conditions and opportunities for efficient LP. While assessing COLE infrastructure sufficiency, it is reasonable to take into

Table 11. Characteristics of information dissemination criterion.

#	Characteristic	Coeff. of importance ($V_{p_{24i}}$)
2.4.1	Systematic accumulation and open access to the wide range of modern electronic learning resources intended for LP of in-service teacher training in COLE, owing to the institute and maintenance of electronic teaching professional editions, research libraries, depositories, etc.	0.30
2.4.2	Impact factor of professional editions of COLE	0.20
2.4.3	The count of views and downloads of products published in digital research libraries or depositary of COLE	0.20
2.4.4	Organization of the system of timely acquaintance of LP participants with current and valid educational data regarding in-service teacher training in COLE. For example, using Google Sites we can launch and maintain a website that can be the original learning data source where you can find terms, forms, academic ranking, electronic classes schedule and online support of academic staff of COLE, as well as themes, requirements for course papers and projects, methods of proper academic writing, allocation of the chosen course paper themes to students, etc.	0.30

consideration the assessment of COLE infrastructure sufficiency made by students and academic staff on COLE.

This way, the characteristics of assessment and reflexive criterion (2.5) for assessing COLE infrastructure sufficiency are determined (Table 12).

Table 12. Characteristics of assessment and reflexive criterion.

#	Characteristic	Coeff. of importance ($V_{p_{25i}}$)
2.5.1	Assessment of COLE infrastructure sufficiency by academic staff	0.50
2.5.2	Assessment of COLE infrastructure sufficiency by students	0.50

6 Academic Staff ICT Competence Facet

Let us determine the criteria characteristics based on the factor of ICT competence of COLE academic staff.

6.1 Characteristics of the Quantitative Criterion

Organization and provision of learning process of in-service teacher training is performed by the academic staff of COLE. It means that the efficiency level of COLE depends on the degree of ICT competence of COLE academic staff.

This way, the characteristics of quantitative criterion (3.1) for assessing ICT competence of COLE academic staff are determined (Table 13).

Table 13. Characteristics of quantitative criterion.

#	Characteristic	Coeff. of importance ($V_{p_{31i}}$)
3.1.1	The number of academic staff, involved in LP of in-service teacher training of COLE, based on the primary place of employment	0.38
3.1.2	The number of academic staff, involved in LP of in-service teacher training of COLE, based on the second job	0.28
3.1.3	The number of academic staff, involved in LP of in-service teacher training of COLE, on an hourly basis	0.34

6.2 Characteristics of the Appropriateness Criterion

Modern postgraduate pedagogical educational institute should not act as a unitary unit, but as a center of different educational degrees [11]. For this reason, the degree of its academic staff involvement in the teaching process at general educational establishments and higher educational establishments (of a pedagogical kind) determines the degree of improvement of uninterrupted pedagogical education. This way, the characteristics of appropriateness criterion (3.2) of ICT competence of COLE academic staff are determined (Table 14).

Table 14. Characteristics of appropriateness criterion.

#	Characteristic	Coeff. of importance ($V_{p_{32i}}$)
3.2.1	The number of teaching staff, involved in COLE, who also work as computer science teachers at general educational establishments and/or teach in COLE	0.49
3.2.2	The number of teaching staff, involved in COLE, who teach ICT subjects and/or teach in COLE of other HEE (of a pedagogical kind)	0.51

6.3 Characteristics of the Educational and Qualification Criterion

COLE academic staff's ICT competence is, first of all, defined by the degree of their academic achievements in using information and communications technologies. Obtaining of further education by academic staff of COLE determines its orientation to constant knowledge enrichment, harmonious individual and professional development.

For this reason, let us single out the basic characteristics of educational and qualification criterion (3.4) for assessing ICT qualification of COLE academic staff (Table 15).

Table 15. Characteristics of educational and qualification criterion.

#	Characteristic	Coeff. of importance ($V_{p_{33i}}$)
3.3.1	The number of COLE academic staff with junior university degree in computer science teaching, etc.	0.30
3.3.2	The number of COLE academic staff with a scientific degree of Candidate, Doctor of Science in 13.00.10 'Information And Communications Technologies In Education', 13.00.02 'Theory and Methods of Teaching' (Computer Science)	0.20
3.3.3	The number of COLE academic staff mentioned in characteristic 3.3.2 with an academic degree of Assistant Professor, Professor	0.20
3.3.4	Obtaining of further education by COLE academic staff, subject matter and mode of study of which relates to the use of ICT in education confirmed by the appropriate documents, such as diplomas, certificates, etc.	0.30

6.4 Characteristics of the Research and Practice Criterion

Disclosure on the basis of academic and/or organizational and methodological experience, and combination of results of theoretical and practical research of COLE academic staff concerning the use of information and communications technologies in learning process during in-service teacher training, creates regularities of further development of COLE. This way, the characteristics of the research and practice criterion (3.4) for assessing ICT qualification of COLE academic staff are determined (Table 16).

6.5 Characteristics of the Assessment and Reflexive Criterion

It should be mentioned that the level of organizational and teaching work of COLE academic staff is determined by the level of measures relating to teaching work administration, which is done at two levels: at the level of the department as the principal unit of an institute of postgraduate pedagogical education that organizes teaching work around academic subjects; and at the level of postgraduate pedagogical educational institute, where problems are solved and teaching activities of general institutional and inter institutional scale between departments are performed.

The extensive use of the capabilities of information and communications technologies in LP of in-service teacher training requires high level of ICT competence from COLE academic staff.

To determine the level of ICT competence of COLE academic staff, self-evaluation and students' evaluation of ICT competence of COLE academic staff should be taken into consideration. This way, the basic characteristics of assessment and reflexive criterion (3.5) are determined (Table 17).

To do a detailed evaluation of ICT competence level of COLE academic staff according to every indicator, it is needed to make a distribution of academic and

Table 16. Characteristics of research and practice criterion.

#	Characteristic	Coeff. of importance (V_{p34i})
3.4.1	Experience in using ICT in academic work	0.15
3.4.2	Publishing of research work (RW) results: the number of published (issued) works in line with the research paper topic confirmed by electronic full-text versions (copies), published online; the number of web-oriented electronic resources in line with the research paper topic confirmed by URLs and web access to their main components [26]	0.13
3.4.3	Distribution of research work results: the count of views and downloads of electronic versions (copies) of research, developmental, teaching, information products in line with the RW topic published online; the number of requests (visits) of URLs of electronic resources created within the framework of a particular RW and considered as an intermediate or final result of such work; the rating of web-resources created within the framework of a particular RW and considered as an intermediate or final result of such work; the number of publications of RW results in national and foreign professional editions included into international electronic scientometric and abstract databases, particularly those defining the impact factor of editions) [26]	0.12
3.4.4	The use of research work results: index of citation or reference to product in domestic editions according to scientometric platforms and databases like Google Scholar, SciVerse Scopus, Scholarometer, positive feedback (reviews, comments, recommendations, etc.) about such products, target audience survey results regarding the peculiarities of its use, documented proof of its introduction (certificates, acts, letters of support, inclusion to the list of recommended sources, etc.) [26]	0.12
3.4.5	Approbation of research work results: reports at international, all-Ukrainian and interregional research and practice conferences, ICT seminars, online conferences	0.16
3.4.6	Recognition of achievements, related to ICT development and introduction to in-service teacher training courses, by the scientific society, including participation and achievements in research and teaching competitions, state programs, etc.	0.16
3.4.7	Organizational and methodological work using ICT, namely: introductory and exit diagnostics of students, timetable scheduling and regular update, web-site content, etc. (Google resources); students' attendance monitoring and current evaluation of their academic activity according to the credit-unit system (LearnBoost); distance learning realization (Moodle); organization and support of teaching community (Drupal); conducting of online seminars, conferences, tutorials, etc. (WizIQ), etc.	0.16

Table 17. Characteristics of assessment and reflexive criterion.

#	Characteristic	Coeff. of importance ($V_{p_{35i}}$)
3.5.1	ICT competence self-assessment by COLE academic staff, involved in LP of in-service teacher training	0.45
3.5.2	Students' assessment of ICT competence of COLE academic staff, involved in LP of in-service teacher training	0.55

assisting staff based on their involvement in the postgraduate pedagogical educational institute work: based on the primary place of employment, on the second job, on an hourly basis, which will intensify LP and increase the work level of an institute of postgraduate pedagogical education regular staff, and also become fundamental in staff correction of an institute of postgraduate pedagogical education.

7 Students ICT Competence Facet

Let us determine the criteria characteristics based on the factor of ICT competence of COLE students.

All COLE academic staff, to a greater or a lesser extent, is responsible for the quality of student advanced training: all academic, teaching methods and administrative workers of an institute of postgraduate pedagogical education involved in LP (headed by the department chairs, dean's office of an institute of postgraduate pedagogical education). That is why defining the level of ICT competence of students passing postgraduate courses has a great importance in evaluating the efficiency of COLE, as well as in decision making by COLE academic staff (that is responsible for students' training level) regarding the optimality of using information and communications technologies during learning process of in-service teacher training.

7.1 Characteristics of the Motivational and Axiological Criterion

In-service teacher training in an institute of postgraduate pedagogical education is done by individual education plan (drafted on the basis of curriculum). The responsibility for its implementation lies with the students themselves. For this reason, COLE students' academic success is determined by the level of their motivation and their targeting at the appropriateness of ICT use in learning, cognitive and professional activity.

This way, basic characteristics of motivational and axiological (value-conscious) criterion (4.1) for assessing ICT competence of in-service teacher training students in COLE are determined (Table 18).

7.2 Characteristics of the Cognitive Criterion

The peculiarity of in-service teacher training in COLE is contribution to the students' mastering of new knowledge about the basics of functioning and peculiarities of ICT use in education.

Table 18. Characteristics of motivational and axiological (value-conscious) criterion.

#	Characteristic	Coeff. of importance (V_{p41i})
4.1.1	Students' aspiration for mastering the ICT recommended by the academic staff of COLE	0.34
4.1.2	Students' interest in the use of innovative ICTs in the teaching process of a general educational establishment	0.34
4.1.3	Students' interest in doing in-service teacher training in distance or mixed mode of learning (full-time, correspondence, etc.)	0.32

For this reason, let us single out basic characteristics of cognitive (knowledge) criterion (4.2) for assessing ICT competence of in-service teacher training students in COLE are determined (Table 19).

Table 19. Characteristics of cognitive (knowledge) criterion.

#	Characteristic	Coeff. of importance (V_{p42i})
4.2.1	Basic knowledge about the functioning of ICTs	0.32
4.2.2	Students' knowledge about public health rules and regulations of ICT use in the teaching process at a general education establishment	0.30
4.2.3	Students' knowledge about the opportunities, perspective tendencies and expediency of use of innovative information and communications technologies in the teaching process of an educational establishment	0.38

7.3 Characteristics of the Praxeological Criterion

The attainment level of in-service teacher training students in COLE also depends on the level of their skills in using ICT at the general educational establishment. This way, basic characteristics of praxeological (pragmatist) criterion (4.3) for assessing ICT competence of in-service teacher training students in COLE are determined (Table 20).

7.4 Characteristics of the Academic Criterion

When evaluating ICT competence of in-service teacher training students in COLE, it is important to consider the students' use of information and communications technologies in research and educational activities. For this reason, let us single out the basic characteristics of the academic criterion (4.4) (Table 21).

Table 20. Characteristics of praxeological (pragmatist) criterion.

#	Characteristic	Coeff. of importance ($V_{p_{43i}}$)
4.3.1	Students' ability to use information and communications technologies rationally while organizing and learning at a general educational establishment	0.16
4.3.2	Students' ability to install and configure the hardware and software necessary for learning at a general educational establishment	0.14
4.3.3	Students' ability to find, process and use electronic resources in learning at a general educational establishment	0.15
4.3.4	Students' ability to create original pedagogical practice using advanced ICT tools	0.15
4.3.5	Students' ability to organize distance learning and use its elements in the class at a general educational establishment	0.12
4.3.6	Students' ability to receive and provide consultative assistance through the ICT means of feedback	0.13
4.3.7	Students' experience in the use of ICTs in the teaching process of a general educational establishment	0.15

Table 21. Characteristics of the academic criterion.

#	Characteristic	Coeff. of importance ($V_{p_{44i}}$)
4.4.1	Number and volume (in printed papers) of students' published scientific works and their publications in scientific journals related to the implementation of ICT in the teaching process at general educational establishments	0.30
4.4.2	Students' participation in research and practice conferences on ICT, also over the Internet	0.35
4.4.3	Students' obtaining further education, subject-matter of which and/or mode of study relates to the use of ICT	0.35

7.5 Characteristics of the Assessment and Reflexive Criterion

The constantly increasing level of ICT competence among in-service teacher training students stimulates the renovation of technologies of professional development of COLE teaching staff.

That is why the characteristics of assessment and reflexive criterion (4.5) of ICT competence evaluation of in-service teacher training students in COLE (Table 22) are among the key parameters of efficiency assessment of COLE.

Table 22. Characteristics of the assessment and reflexive criterion.

#	Characteristic	Coeff. of importance ($V_{p_{45i}}$)
4.5.1	ICT competence self-assessment by students of in-service teacher training in COLE	0.48
4.5.2	Evaluation of students' ICT competence by COLE academic staff	0.52

8 Assessment Based on the Proposed Model

On close examination of COLE efficiency on the basis of the built criteria factor model, the expert group needs to determine the levels of each COLE efficiency criterion's manifestation. It is relevant to do through measurement of appropriate criterion characteristics based on the following system of evaluation: 0 points – the characteristic is not exhibited; 1 point – the characteristic is more not exhibited than exhibited; 2 points – the characteristic is more exhibited than not exhibited; 3 points – the characteristic is exhibited completely. If the coefficient of criterion manifestation ($K_{k_{yj}}$), which is defined as a sum of products of assessment of each criterion's manifestation by their corresponding importance: $K_{k_{yj}} = \sum_{i=1}^{n} \left(O_{p_{yji}} \cdot V_{p_{yji}} \right)$, where n is the number of k_{yj} criterion characteristics not less than 1.5, then such criterion is considered positive and the level of its manifestation is determined by the numerical value of the $K_{k_{yj}}$ coefficient, i.e.: $1{,}50 \leq K_{k_{yj}} \leq 1{,}65$ – critical level, $1{,}66 \leq K_{k_{yj}} \leq 2{,}25$ – satisfactory level, $2{,}26 \leq K_{k_{yj}} \leq 3{,}00$ – high level of criterion manifestation.

If the share of positive criteria of the measurable factor is not less than 50 %, such factor is considered positive. The level of its manifestation is determined based on the numerical value of the factor's manifestation coefficient (K_{f_y}), which is calculated according to the formula: $K_{f_y} = \sum_{j=1}^{m} \left(K_{k_{yj}} \cdot V_{k_{yj}} \right)$. If $1{,}50 \leq K_{yj} \leq 1{,}65$, then the factor of manifestation is of critical level, with $1{,}66 \leq K_{f_y} \leq 2{,}25$ it is satisfactory, while $2{,}26 \leq K_{f_y} \leq 3{,}00$ means it is of high level.

If in the process of measuring of the efficiency of COLE at least one of its factors is negative, such COLE is considered ineffective. On the contrary, if all the factors of COLE efficiency are positive, such COLE is considered efficient. Its efficiency level is determined through the numerical value of the efficiency coefficient (K_e): $K_e = \sum_{y=1}^{4} \left(K_{f_y} \cdot V_{f_y} \right)$, – i.e.: $1{,}5 \leq K_e \leq 1{,}65$ as critical, $1{,}66 \leq K_e \leq 2{,}25$ as satisfactory or $2{,}26 \leq K_e \leq 3{,}00$ as high.

Criteria factor model of efficiency assessment of COLE, built this way, allows determine the level of efficiency of COLE in Ukraine.

However, COLE is a complicated educational system. It must correspond both to the state standards and the current needs of the rapidly changing world. Therefore, when conducting criteria factor evaluation of COLE efficiency, we need to correlate it to the average and the highest level of efficiency of computer-oriented learning environment of an institutions of postgraduate pedagogical educational in Ukraine.

9 Conclusions and Future Work

To conclude, the efficiency of COLE is a judgmental category of COLE defined by the level of realization of its tasks leading towards the main goal of the system of postgraduate pedagogical education and conditioned by the key characteristics of individual structural components, and COLE in general.

The need for objective efficiency assessment of COLE determined the development of the criteria factor model and description of the system of COLE efficiency assessment.

The use of the built criteria factor model of efficiency assessment of COLE (Table 1) provides criterion measurement of the efficiency of COLE based on four factors: (1) the efficiency of LP of in-service teacher training in COLE; (2) COLE infrastructure sufficiency; (3) ICT competence of COLE academic staff; (4) students' ICT competence. To conduct such an assessment 80 criteria were suggested, each of them containing 2–7 characteristics (Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, and 22).

There is a need in further research into the efficiency assessment of COLE in Ukraine. It will allow to identify advantages and disadvantages of computer-oriented learning environment of an institutions of postgraduate pedagogical educational in Ukraine and suggest the desired trends.

References

1. Etzioni, A.: Modern Organizations. Prentice-Hall, Englewood Cliffs (1964)
2. Beshelev, S.D., Gurvich, F.G.: Mathematical and Statistical Methods of Expert Assessments, 2nd edn. Statistika, Moscow (1980) (in Russian)
3. Busel, V.T. (ed.): Big Dictionary of Modern Ukrainian. Irpin`: VTF «Perun», Kyiv (2005) (in Ukrainian)
4. Cabinet of Ministers of Ukraine: On Licensing the Education Service Provision Business: Resolution #1019 (2007) (in Ukrainian)
5. Cabinet of Ministers of Ukraine: On Approval of the Accreditation Regulation of Higher Educational Establishments and Special Fields in Higher Educational Establishments and Vocational Schools: Resolution #978 (2008) (in Ukrainian)
6. Comptoirs: Pädagogische Real-Encyclopädie oder Encyclopädisches Wörterbuch des Erziehungs- und Unterrichtswesens und seiner Geschichte: für Lehrer an Volksschulen und andern Lehranstalten, für Eltern und Eryieher, für Geistliche, Schulvorsteher und andere Freunde der Pädagogik und des Schuwesens. Bearbeitet von einem Vereine von Predigern und Lehrern und redigirt von Hergang, K.G., Grimma: Verlags-Comptoirs, 1. Bd. 1843/2 (1847) (in German)
7. Daft, R.L.: The Theory of Organizations: Textbook for Undergraduate Students Majoring in «Organization Management», [lane. from English. ed. E.M. Korotkova]. YuNITI-DANA, Moscow (2006) (in Russian)

8. The Dictionary of Ukrainian. Ukrainian Academic Glossary (1970–1980). http://sum.in.ua/s/zasib (in Ukrainian)
9. Dolgorukov, A.M.: Strategic Art: Targeting in Business Development Stratagems, the Incarnation. OOO «1-S-Pablishing», Moscow (2004) (in Russian)
10. Kodzhaspirova, G.M., Kodzhaspirov, A.Yu.: Dictionary of Pedagogy. Y`KCz «MarT»; Rostov n/D: Publishing Center «MarT», Rostov on Don (2005) (in Russian)
11. Kolos, E.R.: Model of the organization of computer-oriented learning environment of an institute of postgraduate pedagogical education. Educ. Technol. Soc. **16**(1), 527–535 (2013). http://ifets.ieee.org/russian/depository/v16_i1/pdf/9.pdf (in Russian)
12. Kolos, K.R.: Didactic requirements for computer oriented training environment of postgraduate teacher education. Inf. Technol. Learn. Tools **35**(3), 11–21 (2013). http://journal.iitta.gov.ua/index.php/itlt/article/view/834/619 (in Ukrainian)
13. Kolos, K.R.: Importance of Efficiency Assessment Characteristics of Computer-Oriented Learning Environment of an Institute of Postgraduate Pedagogical Education (2013). https://docs.google.com/forms/d/1BV4aMiqvZH9GD-GTxRL0j75pfqwkj0myus92JfDa1nM/viewform (in Ukrainian)
14. Kolos, K.R.: The main components of computer-based learning environment post-graduate teacher education. In: Proceedings of Scientific Conference of the Institute of Information Technology and Training, pp. 170–171. NAPS of Ukraine, Institute of Information Technology and Training, Kyiv (2013). http://lib.iitta.gov.ua/1422/1/%D0%9A%D0%BE%D0%BB%D0%BE%D1%81_21_01_2014.pdf (in Ukrainian)
15. Kremen`, V.G. (chief ed.): Encyclopedia of Education, Acad. Ped. Sciences of Ukraine, Yurinkom Inter, Kyiv (2008) (in Ukrainian)
16. Kremen`, V.G., Bykov, V.Yu.: Categories of «Space» and «Environment»: characteristics of the model representation and educational use. Theory Pract. Soc. Syst.: Philos. Psychol. Educ. Soc.: Q. Sci. J. **2**, 150 p. (2013) (in Ukrainian)
17. Kremen`, V.G., Tabachny`k, D.V. (eds.): National Educational Glossary: Higher Education. TOV «Vy`davny`chy`j dim «Pleyady`», Kyiv (2011) (in Ukrainian)
18. Longman: Dictionary of Contemporary English. http://www.ldoceonline.com/dictionary/effective
19. Ministry of Education and Science, Youth and Sports of Ukraine: On Approval of Licensing Conditions of Education Service Provision in Higher Education: Order #1377 (2011) (in Ukrainian)
20. Ministry of Education and Science, Youth and Sports of Ukraine: On Approval of State Requirements for the Accreditation of Degree Programs, Special Fields and Higher Education Establishments (Qualitative Characteristics): Order #689 (2012) (in Ukrainian)
21. Ministry of Education and Science, Youth and Sports of Ukraine: On Approval of Document Forms to be Submitted with Applications for the Licensing of Education Services in Preschool, High School, Extracurricular, Vocational and Higher Education: Order #1021 (2012) (in Ukrainian)
22. Ministry of Education of Ukraine: Regulations on the Organization of the Educational Process in Higher Education (1993). http://zakon4.rada.gov.ua/laws/show/z0173-93 (in Ukrainian)
23. Ozhegov, S.I., Shvedova, N.Ju.: Dictionary of Russian language. Azbukovy`k, Moscow (1999) (in Russian)
24. Philosophical Dictionary. http://www.insai.ru/slovar/kriterii-0 (in Russian)

25. President of Ukraine: About the National Strategy for the Development of Education in Ukraine for the Period till 2021 (2013). http://www.president.gov.ua/documents/15828.html (in Ukrainian)
26. Spirin, O.M.: ICT monitor the results of scientific research. Inf. Technol. Learn. Tools **36** (4), 132–152 (2013). http://journal.iitta.gov.ua/index.php/itlt/article/view/890/655 (in Ukrainian)

Design of Electronic Learning Courses for IT Students Considering the Dominant Learning Style

Nataliia V. Morze[1] and Olena G. Glazunova[2(✉)]

[1] Borys Grinchenko Kiev University, Kiev, Ukraine
nmorze@kubg.edu.ua
[2] National University of Life and Environmental Sciences of Ukraine, Kiev, Ukraine
o-glazunova@nubip.edu.ua

Abstract. Methods of using e-learning courses to support learning activities of students at higher educational institutions are the subject of a large number of scientific and educational studies. In particular, much attention is paid to the structure, content, and format of educational resources of e-learning courses. However, the problem of dependency of their efficiency on the learning styles of students still needs to be further researched. This paper deals with the learning styles characteristic for IT students on the basis of determining their leading modality, designs the structure of e-learning courses for IT students considering the dominating learning styles, and provides the results of pedagogical experiment by measuring the performance and satisfaction in learning activity.

Keywords: Learning style · Leading modality · E-learning course · MOOC

1 Introduction

Modern universities are developing in accordance with the concept of Smart-education, one of the main trends of which is openness of learning systems - placing educational content openly available to students around the world, the development of the open source system, development of knowledge sharing according to the following patterns: “student – student”, “teacher – teacher”, “students - teacher” and “students - teachers” [1, 2]. One of the most illustrative examples of open learning systems for mass advancement and training of students are MOOCs (mass open online courses). An important step in the development of the idea of using the MOOC was the adoption of the UNESCO Declaration on global public policy on the issue of open electronic courses, which aims to develop standards for electronic courses, provide synergy access, conduct educational seminars on the design and the use of e-courses, develop collaboration of scientists and teachers, educational quality assurance [3]. Furthermore, for the last decade, there has been a significant demand for online learning from students as well as all population strata, which proves the need for lifelong learning on the basis of non-formal and informal rather than formal learning. In this respect we can identify a number of economic factors and observe real competition among commercial

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 261–273, 2014.
DOI: 10.1007/978-3-319-13206-8_13

institutions [4]. Paying attention to the quality of e-learning, we focus more on the quality of developing educational resources, tasks for independent work, organization of knowledge control. However, the effectiveness of perception of educational resources also depends on learning styles and needs of students. Therefore, it is essential to develop resources for the curriculum in the most appropriate form for students, considering their main channel of information acquisition or personality. A teacher or curriculum designer develops the material in order to fit the students with different learning styles. This directly affects the level of learning new information. Methods of teaching certain subjects are often based on specific learning styles. Sometimes it is best to learn the information by using a combination of both. Justin Ferryman recommends considering different styles since it is useful not only for a better acquisition of the material, but to preserve residual knowledge as well [5].

2 Problem

The problem of taking into consideration the styles has been well studied in traditional teaching. However, in order to organize online learning, the problem requires additional analysis. In other words, there is a need to define learning styles when determining the content and structure of a distance course, forms of presenting material, a set of training activities to study the discipline, etc. In addition, it is essential to research into the dominant learning styles of students of different specialties, including IT students, and the level of their assessment of e-learning courses of a specific format.

3 Related Work

A learning style describes individual factors that may remain relatively stable over time. Scientists define a learning style as “a biological and mental set of personal characteristics that make the same teaching method effective for some students and ineffective for others” [6].

Different people apply different approaches to solving problem of similar content and complexity or performing the same operations. In this case, a person carries out different activities applying their own peculiar method and performance efficiency (e.g. a type of work similar in content and complexity can be performed by different people with different efficiency – they obtain results of different quality, spend different time to do the task of equal complexity etc.). Conversely, if a person possesses several different methods of activity and is able to adequately use either of them to solve a particular set of problems (tasks) or their stages, a range of effective operation can be greatly extended and complicated problems (including sub problems of different content and complexity or their stages), on the whole, can be solved quite effectively, or at least, more successfully than when a complex problem is solved by a person able to apply only one method. [7]. It should be noted that there is a direct correlation between the style a person uses to solve almost all range of problems he faces in life and the style he applies when learning, and vice versa [8].

A Kolb experiential learning model [9] measures a student's strengths and weaknesses according to the four learning abilities: concrete experience, reflective observation, abstract conceptualization and active experimentation. Based on the combination of these four learning abilities the Kolb model identifies four dominant learning styles: divergence, assimilation, convergence and accommodation (Fig. 1). People of conveying style are strong in abstract conceptualization and active experimentation. In divergence style concrete experience and reflective observation prevail. Assimilators are strong in abstract conceptualization and reflective observation, while the accommodators are strong in concrete experience and active experimentation.

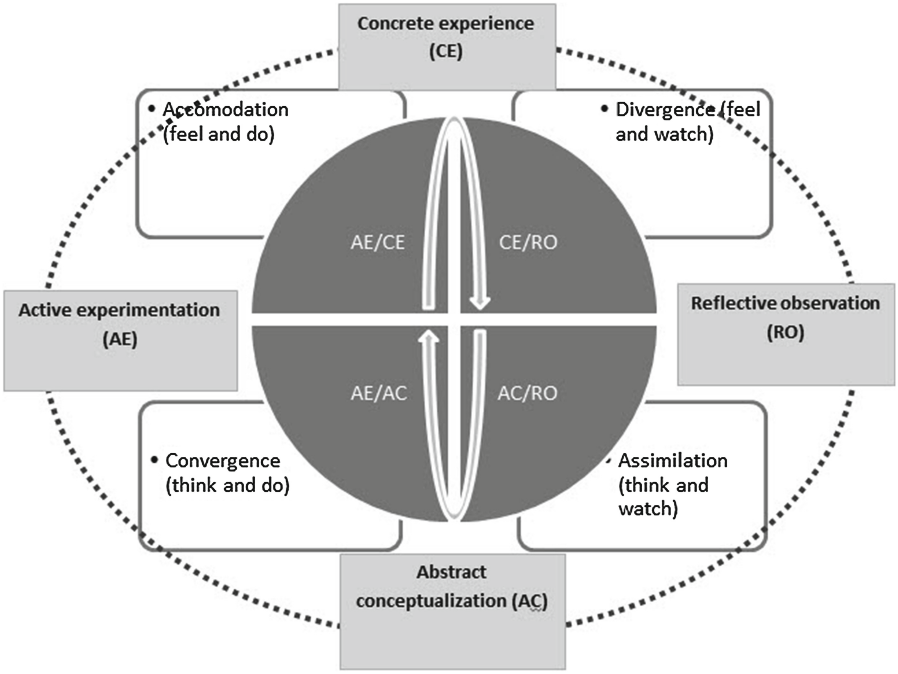


Fig. 1. Kolb experiential learning styles model

Another classification of learning styles is VARK (visual, audial, reading/writing, kinesthetic) developed by Fleming and Mills in 1992 [10]. Psychology categorizes learners with leading learning modality: sensor – visual learners, with prevailing audial modality – auditory learners, with muscular activity – kinesthetic learners, with reading-writing preference – reading/writing learners (Fig. 2). And, accordingly, the highest percentage of material has been acquired on the basis of leading modality. Visual learners prefer graphical and symbolic form of gaining data information. Auditory learners prefer attending lectures, training programs and participating in discussions. The reading/writing learners prefer text-based materials. Kinesthetic students prefer hands-on experience and learn best by touching and doing.

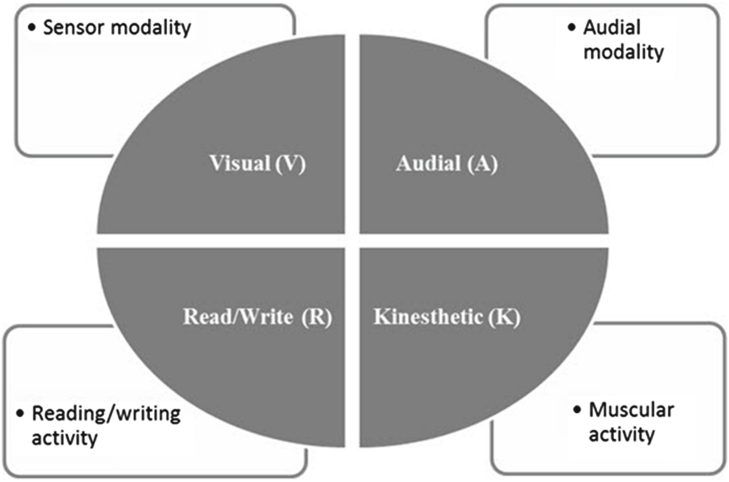


Fig. 2. VARK learning styles model

The first research into the comparison of learning styles and format of distance learning course was conducted in 1999 [11]. Researchers Diaz and Kartnel divided groups of students to study according to a traditional system and on-line, depending on whether students choose to study in social interaction mode or independently in a confined space. As a result of the research the strategy for online learning based on the needs of students and their peculiarities was developed.

Downing and Chim (2002) conducted a study on student satisfaction with online training by categorizing students into active, reflexive, theorists and pragmatists [12]. Activists prefer direct experience, reflexive students prefer just observing. Theorists are keen on analyzing and synthesizing course materials, whereas pragmatists prefer acting quickly to try out new ideas. These studies have shown that reflexive students were more satisfied with online learning, whereas groups with other styles of learning showed no significant relationship between learning styles and satisfaction.

In 1988 Richard M. Felder and Linda K. Silverman [13] investigated the models of learning and teaching. They proposed to define a student's learning style to the 5 questions:

1. Which type of information does a student preferentially perceive: *sensory* (external) – sights, sounds, physical sensations, or *intuitive* (internal) – possibilities, insights, hunches?
2. Through which sensory channel is external information most effectively perceived: *visual* – pictures, diagrams, graphs, demonstrations, or *auditory* – words, sounds? (Other sensory channels–touch, taste, and smell – are relatively unimportant in most educational environments and will not be considered here.)
3. Which organization of information is a student most comfortable with: *inductive*–facts and observations are given, underlying principles are inferred or *deductive*–principles are given, consequences and applications are deduced?

4. How does a student prefer to process information: *actively* – through engagement in physical activity or discussion, or *reflectively* – through introspection?
5. How does a student progress toward understanding: *sequentially* –in continual steps, or *globally* – in large jumps, holistically?

Gardner's theory of multiple intelligences differentiates from specific (primarily sensory) "modalities", rather than seeing intelligence as dominated by a single general ability. In 1983 Howard Gardner has proposed The Theory of Multiple Intelligences in which he suggested that people possess at least seven different forms of intelligence: musical, visual–spatial, logical–mathematical, bodial–kinesthetic, interpersonal, intra-personal [14]. In 1997 he added the naturalistic intelligence. Although the distinction between intelligences has been set out in great detail, Gardner opposes the idea of labeling learners to a specific intelligence. Each individual possesses a unique blend of all the intelligences. Gardner firmly maintains that his theory of multiple intelligences should "empower learners", not restrict them to one modality of learning.

Several researchers have attempted to determine the relationship between VARK learning styles and on-line course efficiency. In particular, Drago and Wagner [15] in their experimental studies claimed that the efficiency of online learning for students with leading "reading-writing" modality was significantly lower than for students with other leading modalities. However, the students combining auditory and "reading-writing" styles showed better results in online courses in comparison with others.

However, not all researchers have observed a relationship between style and learning outcomes. Kratzig and Arbuthnott's research [16] show no significant correlation between learning style and objective memory performance. Participant's learning styles were identified using self-report, questionnaire and after that they have passed standardized memory tests. According to the self-report questionnaire and the self-assessment guest 40 % and 60 % of participants were identified as visual learners through the questionnaire, but only 23 % performed best on the visual test. The percentages were 16 % and 8 % for kinesthetic-learning preferences. When the participants completed the standardized memory test 52 % performed best with the tactile test. These results Kratzig and Arbuthnott conclude that "such styles may indicate preferences and motivations rather than inherent efficiency at taking in and recalling information through specific sensory modalities".

In 2010 an experiment was conducted on the impact of learning style on student performance and satisfaction using the online environment according to Kolb model [17]. Students were split into groups who studied two types of Web modules - textual and multimedia information. The study found that the learning style in this case did not affect the satisfaction and the progress of students.

Also, Christopher Pappas debunked The Myth of Learning Styles [18]. He says that "there is no convincing evidence to prove that when an instructor changes the presentation mode of his course to match the learning style of his students actually helps them learn". He also recommend to create a course based on the motivational characteristics of the students and not their learning styles, and always be ready to adjust the learning methods and techniques and engage multiple senses rather than just one.

Despite the considerable amount of research in this area, the problem of presenting educational information in a distance course for students with different leading modality still needs to be studied. First of all, it is caused by the rapid development of information and communication technologies, particularly Web 2.0 technologies that provide brand new possibilities in a wide choice of presenting educational materials, videos and audio materials and their structuring. The aim of our study is to analyze the leading modality dominating in IT students of higher educational institutions, and on the basis of the results obtained to offer the structure, format of e-learning courses content and the assessment system of learning outcomes of IT students.

4 Experimental Setting and Research Methods

Students of the Faculty of Computer Sciences and Economic Cybernetics at the National University of Life and Environmental Sciences of Ukraine participated in the pedagogical experiment. The first experimental group of students was trained using blended learning technologies which combine traditional educational technologies and the elements of distance learning, the second group of students was trained using traditional technologies and printed textbooks. The leading modality or a combination thereof has been determined for each participant of the experiment. The results of the experiment were presented in the guidelines to the methods of teaching students with different learning styles through the use of e-learning courses. In addition, the impact of the structure and format of presenting training material on the progress and the level of satisfaction of students and organization of training activities was investigated. During the study we used the following methods: theoretical analysis of the methods of determining the sources of learning styles of students, studying and summarizing best practices of the use of e-learning courses to support the learning process, analysis, assessment, comparison, pedagogical experiment.

5 Learning Styles of IT Students

Consideration of a learning style is an important issue for both teachers and students using e-learning technologies. Teachers need to know the learning styles of students to further adapt teaching methods apply appropriate resources and determine the types of student activities when using e-learning courses and to better understand the differences between the needs and peculiarities of students. Students who are aware of their learning style learn better as they choose to study those resources, the perception of which has the largest educational effect. If a teacher's style is different from the style of the majority of his students, it is necessary to change the method of teaching and the organization of training activities. Within the framework of the research a survey of IT students was conducted according to the questionnaire by VARK model [19], so that the teachers of professionally-oriented courses had an idea of the student's learning styles. The purpose of this survey was to determine the leading modality of students who have chosen to obtain IT profession. Characteristics of each style are listed in Table 1.

Table 1. The characteristics of learning styles according to the VARK model.

Name of the style	Characteristics
Visual	Information is predominantly presented in the form of pictures, diagrams, flow charts and all those symbolic lines, circles, trees and other elements that are used instead of teacher's words to represent teaching materials. Visual learners prefer receiving information in a holistic rather than simplistic way. As a rule, the appearance of an object, color matching, information placing and spatial design are of critical significance
Auditory	The mode of perception describing the advantage of obtaining information materials aurally. Students with such modality are better taught through lectures, seminars, listening to recorded lessons, group discussions, web chats or talks about the object. Auditory learners most value words spoken verbally rather than written, listening to expert's explanations rather than reading a textbook paragraph
Verbal	It is preferable to present information materials in the form of words. It is no secret that many academic methods are exclusively designed for this style. The style perception tends toward textual introduction and reproduction, reading and writing of teaching materials in different forms. Unlike others, verbal learners believe in the power of words. They like to have information systematized in written form
Kinesthetic	Refers to perceptional benefit of using experience and practice (simulation or reality). Since this description may also fit the other modalities, the key is that the student is always "connected to" reality through experiments, examples, practice or simulation. Kinesthetic learners need to experience the situation in order to understand it. Practical importance and feasibility are valued. As a rule, such learners remember only those things that took place in reality

In order to offer IT students the e-learning courses, the structure of which will be most effective for perception according to their learning styles, we carried out the social and psychological diagnosis of students in specialty "Computer Sciences". We used surveys and questionnaires, proposed in the sources [20, 21] by VARK model. Students are encouraged to fill in a form with 16 (49) questions. The questionnaire helps to determine the students' leading modality and, correspondingly, their main learning style.

The survey results allowed us to identify 15 groups of students with different modality of information perception. The results of the survey (Fig. 3) indicate that the IT students possess the learning styles that combine auditory and kinesthetic modality (AK) – 18 %, visual and kinesthetic modalities (VK) – 17 %, auditory, visual and kinesthetic modalities (VAK) – 16 %. This statistics concludes that the future IT professionals learn best with the help of practical experience, verbal and visual explanation of learning material.

6 Structure of E-Learning Courses Based on Learning Styles

One of the objectives of the experimental study was to determine the optimal structure of e-learning courses and the ways of presenting learning resources for IT students

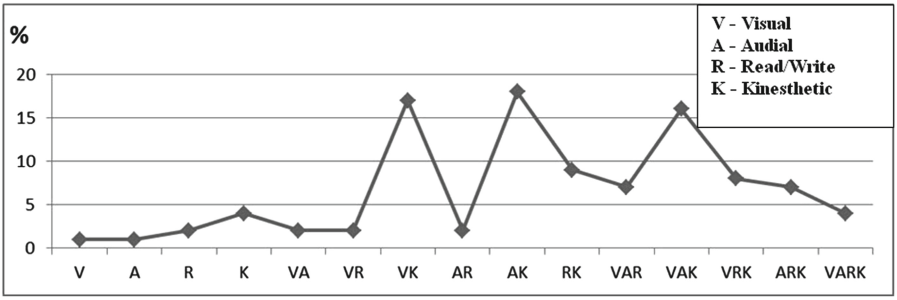


Fig. 3. Correlation of leading modalities of students in specialty "Computer Sciences"

according to their learning style. An e-learning course (ELC), which is used to support the learning process in studying a specific discipline when training future IT professionals, is a set of learning resources and types of activity to provide the study of theory, carrying out practical tasks, independent work, student/teacher cooperation, monitoring academic performance etc. The general approach to the structure of ELC, presented in [22] is to determine different types of e-learning resources appropriate at different stages of the learning process, though it does not consider the individual characteristics of student's perception of learning material. In particular, when studying theoretical material the students are encouraged to use the following resources: an electronic textbook, an electronic copy of the printed textbook, video lectures, video lesson, presentation, virtual excursion, links to external resources etc. In order to develop practical skills in e-learning course teachers can use laboratory workshops, tasks for laboratory (practical) work, individual work, virtual seminars, webinars, group research projects, forums, chats etc. Besides, the e-learning course should include the activities to monitor academic performance and reflection, as well as such resources as: tests, assignments, surveys, questionnaires, checklists etc.

Taking into consideration the dominant learning style of the student group it is vital to choose a different format of learning resources and activities in an electronic course. For visual students the resources of ELC should provide an opportunity to view the lecture material in a structured way in which the information is classified according to various criteria, key points are underlined or highlighted, there are images, video clips, posters, slides, flowcharts, graphs and charts. The tasks for independent work should have the materials where a student will be able to convert visual images into verbal ones thus creating vivid memory of events and actualizing corresponding visual sequence in memory. As an example, students may be given the task to design classification tables (charts, graphs, charts) on the basis on the review of learning material in textual, tabular, graphical and video formats. These students appreciate the form of presentation of learning material, design, color gamma. Everything is supposed to be harmonious and attractive. All types of activities designed for students when organizing the training on the basis of e-learning course, suggest that the learning material and the tasks should be structured. This course should include the presentation of learning content, video lessons, memory cards, electronic textbooks, tasks with a clear structure and procedure, examples of task performance. The e-learning technologies may include video conferences using the interactive white board to visualize material.

The resources for students with a dominant auditory modality should provide the theoretical material in the form of video and audio recordings of lectures. To acquire practical skills it is recommended to use webinars recorded in the video format and placed in the course resources. Group work tasks can be effective, since through discussion, brainstorming and talk's auditory learners expand the amount of the material by receiving verbal information from others. Besides, quite effective can be the task to create audio recordings in which the learning material is analyzed and synthesized through spoken language.

For students whose dominating style is "reading-writing" the resources of the training course should be built on the principle of compulsory working up of the material with the help of a pen or keyboard. Theoretical resources can be presented in the form of video lectures, full-text document, definitions, alphabetically built glossary, the list of concepts etc. Practical tasks should include guidelines for assignments and be based on the conversion of the forms of presentation, such as diagrams, flowcharts and graphs into words and statements. When carrying out the tasks students are expected to use references, supporting materials, theses, reports, guidelines, instructions. While studying for the seminars students can use the following tasks: to make a plan of presenting material, its structure, indicating paragraphs and subparagraphs. Among the e-learning technologies it is preferable to use task with written answers, essay tests to monitor learning outcomes.

Students with kinesthetic modality percept educational materials more effectively if e-learning courses include resources in the form of virtual workshops, virtual tours and exhibitions with further analysis. E-course tasks should be based on the examples of solutions performed in previous years for similar tasks. Practical tasks should apply approaches that allow mastering knowledge in practice, use the trial-and-error method. To explain abstract concepts in electronic sources it is recommended to provide lots of examples from real life.

In our research we suggested another structure of electronic course: the resources that take into consideration the dominant leading modality best suited to successful learning of IT students. When designing a new e-course pattern we used such elements that are best suited for people who combine visual, auditory and kinesthetic modalities. To highlight the theoretical material it is proposed to use video format, electronic textbooks as well as additional links to external resources and virtual tours (Fig. 4). To acquire practical skills in electronic courses there should be virtual laboratory workshops with methodological recommendations, individual assignments with clear assessment criteria and the examples of solutions to these assignments. These examples of solutions are advisable to file in screen cast format, which illustrate the process of solution as well as have a soundtrack explaining. To organize independent work it is recommended to provide resources with practical video lessons and assignments for individual study, research group projects, which make it possible to develop teamwork skills in students through the use of information and communication technology. The resources that contribute to the formation of practical skills should include video conferences to discuss issues related to the solution of practical problems. Methods of control should contain tests to assess the level of theoretical knowledge, control tasks to assess practical skills, questionnaires, checklists and other tools for self-assessment and reflection.

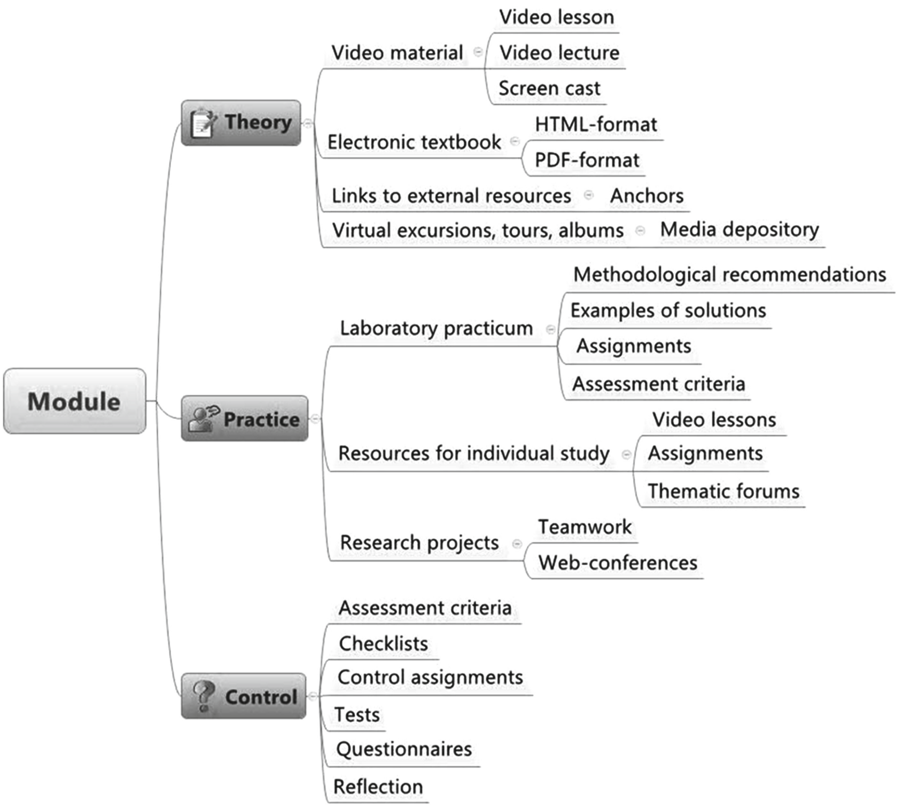


Fig. 4. The structure of a module in the experimental course

In pedagogical theory and practice, there are different approaches to the definition of criteria and indicators of the effectiveness of learning activity. As a measure of the effectiveness of any human activity one can specify two universal criteria - productivity of work and satisfaction of its participants. [23]. As a rule, the participants of the learning process are interested not only in the result of training, but in their satisfaction or dissatisfaction with the process of learning and their relationships. It is not always possible to assess satisfaction by observation. Thus it is necessary to develop diagnostic methods in order to determine the level of satisfaction in the relevant activity.

In the pedagogical experiment, the students studied the first module using blended learning technology and a new model of e-learning course. In the second module, the traditional learning technology was applied, though electronic textbooks were used to study the theoretical material. To determine whether the use of e- learning course influences the efficiency of the learning process, we used the rate of learning progress, namely the results of modular control assessed by 100 grades scale. The results of the modular control included the following activities: laboratory work - 40 points, independent work - 20 points, modular test - 20 points, control task - 20 points. To measure the level of satisfaction with learning activity, we compiled assessment questionnaires according to the criteria of the organization of the process of learning when studying the module. Each criterion of satisfaction questionnaire was assessed by students using a 5- point scale of assessment: completely satisfied, generally satisfied, satisfied but with some comments, generally dissatisfied, strongly dissatisfied. Comparison of progress and satisfaction are shown in Table 2.

Table 2. The results of progress and satisfaction.

Courses	Module 1 (using ELC)		Module 2 (without ELC)	
	Average grade	Assessment level	Average grade	Assessment level
System Analysis	72	4,6	66	3,5
Information Systems Design	84	4,7	63	3,2
Information Technology	78	4,5	67	3,4

While analyzing the results of the study, we have come to the following conclusions: the progress of students in learning outcomes for module 1 on the basis of ELC was significantly higher than for module 2 – on average by 12 %. The level of student assessment of learning material quality when using ELC was also higher by 24 % on the whole. Similar studies were conducted among the students of distance learning. The results of the progress of students who were trained in the intersession period on the basis of ELC increased even further – by 21%, compared with those who were trained according to the traditional techniques of distance learning.

7 Conclusions

When developing learning materials and designing the whole process of training, it is important for teachers to take into account the learning styles of their students to ensure the quality of their learning outcomes and increase their motivation to study. According to the results of the research, the awareness of learning styles can yield benefits for students today. Questionnaires by VARK tool is easy and fast and can provide the important information which can be successfully used in the creation of effective learning environment for students. The information about the learning styles of students and their impact on the learning environment can contribute to a vitally important understanding of the students and create an effective learning environment, in particular, to make decisions on the design and content solutions of e-learning courses at all stages of the course. The obtained results need further confirmation and study. In particular, the promising areas of research, we believe, is to study the relationship between the dominant modality and presentation e-course of theoretical resources, methods and forms of independent work of students using e-learning courses.

In addition, it is useful to conduct a study and compare the results of using e-learning courses for other models of learning styles.

References

1. Helmer, J.: A Pair of Key Trends of This Year Learning: MOOCs and OA. http://www.smart-edu.com/moocs-and-oa.html
2. Pawlowski, J.M., Hoel, T.: Towards a Global Policy for Open Educational Resources: The Paris OER Declaration and its Implications. In: White Paper, Version 0.2, Jyväskylä, Finland (2012)
3. Open Educational Resources. http://www.unesco.org/new/en/communication-and-information/access-to-knowledge/open-educational-resources/
4. Allen, I.E., Seaman, J.: Class Differences: Online Education in the United States (2010). http://www.sloanconsorium.org/publications/survey/class_differences
5. Ferriman J.: 7 Major Learning Styles - Which One are You? (2013). http://www.learndash.com/7-major-learning-styles-which-one-is-you
6. Dunn, R., DeBello, T., Brennan, P., Krimsky, J., Murrain, P.: Learning style researchers define differences differently. Educ. Leadersh. **38**(5), 372–375 (1981)
7. Bykov, V.: Modeli Organizatsiinyh System Vidkrytoi Osvity. Monograph. Atika, Kyiv (2009) (in Ukrainian)
8. Kolb, D.A.: Experiential Learning: Expirience as the Source of Learning and Development. Prentice-Hall, Englewood Cliffs (1984)
9. Ishkov, A.D.: Uchebnaya Deyatelnost Studenta: Psihologicheskie Faktory Uspeshnosti. Monografiya, ASV, Moscow (2004). (in Russian)
10. Fleming, N.D., Mills, C.: Not another inventory, rather a catalyst for reflection. Improv. Acad. **11**, 137–155 (1992)
11. Diaz, D.P., Cartnal, R.B.: Students' learning styles in two classes. Coll. Teach. **47**(4), 130 (1999)
12. Downing, K., Chim, T.M.: Reflectors as online extraverts? Educ. Stud. **30**(3), 265–276 (2004)
13. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Engr. Educ. **78**(7), 674–681 (1988). http://www4.ncsu.edu/unity/lockers/users/f/felder/
14. Theory of Multiple Intelligences. http://en.wikipedia.org/wiki/Theory_of_multiple_intelligences#cite_note-Gilman-1
15. Drago, W.A., Wagner, R.J.: VARK preferred learning styles and online education. Manag. Res. Rev. **27**(7), 1–13 (2004)
16. Kratzig, G.P., Arbuthnott, K.D.: Perceptual learning style and learning proficiency: a test of the hypothesis. J. Educ. Psychol. **98**(1), 238–246 (2006). http://psychology.illinoisstate.edu/cbs/readings/kratzig_arbuthnott2006.pdf
17. Kozub, R.M.: An ANOVA analysis of the relationships between business students' learning styles and effectiveness of web based instruction. Am. J. Bus. Educ. **3**(3), 89–98 (2010)
18. Pappas, C.: The Myth of Learning Styles. E-Learning Industry (2013). http://elearningindustry.com/the-myth-of-learning-styles
19. Fetiskin, N.P., Kozlov, V.V., Manuilov, G.M.: Sotsialno-Psihologicheskaya Diagnostika Razvitiya Lichnosti s Malyh Grup. Izdatelstvo Instituta Psichoterapiyi, Moscow (2005) (in Russian)
20. The VARK Questionnaire. http://www.vark-learn.com/English/page.asp?p=questionnaire
21. Lyubimov, A.: Diagnostika Dominiruyushchey Pertseptivnoy Modalnosti. http://brainmod.ru/tests/typology/perceptual-modality/ (in Russian)

22. Morze, N.V., Glazunova, O.G.: What should be e-learning course for smart education. In: Ermolayev, V. et al. (eds.) Proceedings of the 9th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, CEUR Workshop Proceedings, vol. 1000, pp. 411–423 (2013). http://ceur-ws.org/Vol-1000/ICTERI-2013-MRDL.pdf
23. Stepanov, Y., Andreev, O.: Vyvchennia Zadovolenosti Uchasnykiv Navchalnoho Protsesu. Osvita.ua (2005). http://osvita.ua/school/theory/1376 (in Ukrainian)

ICT in Research and Industrial Applications

On Industrial Strength Bio-design Automation

Wolf-Ekkehard Matzke[1(✉)], Chris P. Barnes[2], Eyck Jentzsch[1], Thorsten Mascher[3], and Michael Stumpf[4]

[1] MINRES Technologies GmbH, Keltenhof 2, 85579 Neubiberg, Germany
{wolf,eyck}@minres.com
[2] University College London, Gower Street, London WC1E 6BT, UK
christopher.barnes@ucl.ac.uk
[3] Ludwig-Maximilians-University Munich, Grosshaderner Str. 2-4, 82152 Planegg-Martinsried, Germany
mascher@bio.lmu.de
[4] Theoretical Systems Biology, Imperial College London, 317 Sir Ernst Chain Building, London SW7 2AZ, UK
m.stumpf@imperial.ac.uk

Abstract. Bio-Design Automation (BDA) denotes the nascent domain-specific Information and Communication Technology (ICT) discipline for synthetic biology, which constitutes the core technology of the Knowledge-Based Bio-Economy (KBBE). Ultimately, the success or failure of synthetic biology and the emerging KBBE equates to the progress or lack of progress in establishing an industrial strength BDA discipline. In this paper, we seek answers to the question "What does it take for BDA to become an industrial strength discipline?" Our goal is to stimulate a broad community discussion including Business Managers, Computer Scientists, ICT professionals, Synthetic Biologists, etc. around this question. To jump-start the debate, we will provide four core hypotheses covering what we believe are the most important aspects to be considered. Given that industrial strength is a composite aggregate of several technical and managerial variables, we have chosen to take a holistic approach and not restrict ourselves a priori to any particular viewpoints. Last, but not least, we will apply our findings and provide a prototypical industrial implementation of a BDA platform.

Keywords: Bio-design Automation · Bio-Design System · Business models · Information and Communication Technology · Synthetic Biology

1 Introduction

The keynote given to the ICTERI 2013 conference at Kherson, Ukraine, [26], highlighted the defining role of Information and Communication Technology (ICT), Biotechnology and Synthetic Biology for the emerging Knowledge-Based Bio-Economy (KBBE). Bio-design Automation (BDA) was identified as the key domain-specific ICT for Synthetic Biology (and the KBBE). The paper at hand goes one step further from the keynote by providing a discussion framework for the industrial implementation of

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 277–299, 2014.
DOI: 10.1007/978-3-319-13206-8_14

BDA that can perhaps serve as a guide for BDA practitioners and researchers. Some initial thoughts on industrial strength bio-design automation were presented at 5th International Workshop on Bio-Design Automation (IWBDA 2013) [28].

To shortly recapitulate: The KBBE aims at the "sustainable production and conversion of biomass, for a range of food, health, fiber and industrial products and energy", where "renewable biomass encompasses any biological material to be used as raw material."[1] Biotechnology is defined by the Organization for Economic Co-operation and Development (OECD) as "the application of science and technology to living organisms, as well as parts, products and models thereof, to alter living or non-living materials for the production of knowledge, goods and services" [41]. Synthetic biology is defined as:

"(A) the design and construction of new biological parts, devices, and systems; and
(B) the re-design of existing, natural biological systems for useful purposes"[2].

ICT is the Economy-Defining Technology (EDT) of the Knowledge-Based Economy (KBE) and it will continue to be an EDT in the KBBE. This statement is undoubtedly true in terms of general purpose ICTs. Every year, Gartner, Inc.[3], a world-renowned information and technology research and advisory company, compiles a list of the top ten strategic general purpose ICT trends for the following year, thus providing an annual fresh perspective. For 2014 Gartner lists the following top ten ICTs[4]: (1) Mobile Device Diversity and Management; (2) Mobile Apps and Applications; (3) The Internet of Everything; (4) Hybrid Cloud and IT as Service Broker; (5) Cloud/Client Architecture; (6) The Era of Personal Cloud; (7) Software Defined Anything; (8) Web-Scale IT; (9) Smart Machines; (10) 3-D Printing. For preceding years Gartner included for example areas such as Strategic Big Data (2013)[5], Next-Generation Analytics (2012)[6] and Cloud Computing (2011)[7]. It is perhaps easy to imagine the application of these general purpose ICTs in the context of the KBBE as the advancement of the KBE. Clearly, the KBBE will benefit from extensive deployment and use of general purpose ICTs. However, the KBBE and in particular synthetic biology require a sophisticated domain-specific ICT solution. Contrary to general-purpose ICT, domain-specific ICT is created to solve problems within a particular area of concern. In the case at hand, the area of concern is synthetic biology. As noted above, the associated domain-specific ICT is comprised under the label BDA. Note that, unlike general purpose ICTs, domain-specific ICTs usually don't receive the public awareness - even among ICT professionals - they merit. For example: Most likely many ICT professionals are highly interested and knowledgeable in general purpose ICTs such as Mobile Apps and

---

[1] THE EUROPEAN BIOECONOMY IN 2030. Delivering Sustainable Growth by addressing the Grand Societal Challenges: http://www.epsoweb.org/file/560.

[2] http://syntheticbiology.org/

[3] http://www.gartner.com/technology/home.jsp

[4] http://www.gartner.com/newsroom/id/2603623

[5] http://www.gartner.com/newsroom/id/2209615

[6] http://www.gartner.com/newsroom/id/1826214

[7] http://www.gartner.com/newsroom/id/1454221

Applications, but only very few might have ever heard about Electronic Design Automation (EDA). EDA is the domain-specific ICT for electronics, which is the ICT enabling the design and development of electronic systems. What if there would be no EDA? There would be no electronics and no mobile apps and applications. It's as simple as this. Last, but not least, it should be mentioned that domain specific ICTs represent really fascinating research and engineering fields, provided one manages to overcome the domain-specific barrier to entry.

Exploitation of the industrial potential of synthetic biology has only recently begun. Currently, we witness the progression of synthetic biology from an emerging technology to an emerging industry. However, it will take many years for synthetic biology to evolve into a fully established industry. To which extent and when these transitions will be completed depends on a multitude of challenges and constraints facing synthetic biology. The siliconization of synthetic biology, that is the degree to which so-called *in silico*[8]-based design penetrates synthetic biology, is amongst the fundamental technical challenges to be addressed. Siliconization is driven by the necessity to increase the productivity of design, which is in turn a requirement for the successful industrialization of synthetic biology. Eventually, *in silico* design has to become the principal design approach of synthetic biology.

Siliconization of synthetic biology is the realm of the nascent field of BDA. It should be noted, however, that the application of *in silico* approaches in biology is by no means a new idea. Bioinformatics and computational biology are well-established *in silico* disciplines. In fact, both are cornerstones of BDA. Bioinformatics is a data-centric discipline. It focuses on the application of computational techniques to understand and organize biological data [23]. Computational biology on the other hand is concerned with computational models of biological phenomena [20]. It is therefore a model/hypotheses-centric discipline. The third cornerstone of BDA is formed by the broad spectrum of established (non-bio) design automation approaches, such as EDA, Architecture, Engineering and Construction (AEC), Mechanical Computer Aided Design (MCAD) etc. Note that, in general, design automations are the disciplines (and domain-specific ICTs) devoted to computerized design processes [7]. Like all domain-specific ICTs, design automations are design productivity-centric.

The positioning of BDA relative to bioinformatics, computational biology and (non-bio) design automation is depicted in Fig. 1. BDA builds on these three established *in silico* disciplines as cornerstones and reuses and integrates the underlying technologies whenever feasible and appropriate. Additionally, BDA provides its own set of solutions to address unique challenges of synthetic biology design.

BDA's ultimate promise is to increase the productivity of synthetic biology. To fulfill this promise, BDA needs to develop industrial strength solutions. Industrial strength is defined as a system's ability to work capably and dependably in an operational setting [36].

---

[8] *In silico* is a term popular among synthetic biologists. Wikipedia explains (see http://en.wikipedia.org/wiki/In_silico): "*In silico* is an expression used to mean 'performed on computer or via computer simulation.' The phrase was coined in 1989 as an analogy to the Latin phrases *in vivo*, *in vitro*, and *in situ*, which are commonly used in biology … and refer to experiments done in living organisms, outside of living organisms, and where they are found in nature, respectively."

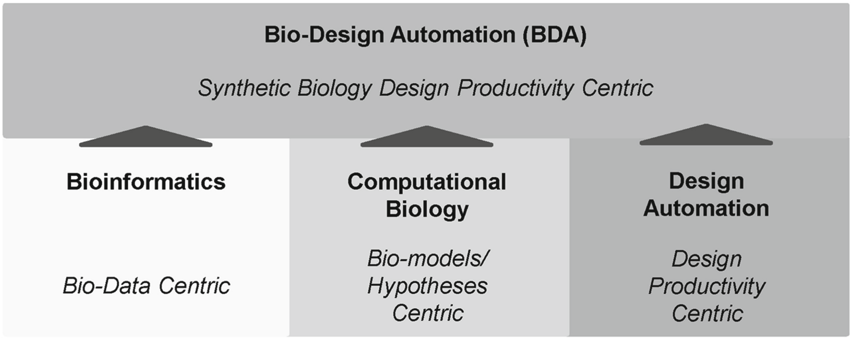


Fig. 1. Positioning of BDA. The figure details the position of *BDA* relative to *bioinformatics*, *computational biology*, and (non-bio) *design automation*. *BDA* reuses and integrates their underlying technologies whenever feasible and appropriate. Additionally, *BDA* provides its own set of solutions unique to synthetic biology design.

In the following we will present what we believe are the four most important aspects to reach industrial strength for BDA. We arrived at these four core hypotheses by examining the BDA industrial strength challenge from the perspectives of various engineering and management disciplines, namely engineering management [37], design theory [17], complexity engineering [6] and, business management theory [43]. In particular, the engineering management perspective led to The Bio-Design System Paradigm hypothesis. The design theory viewpoint revealed The Rational Design Fallacy conjecture. Taking a complexity engineering standpoint resulted in the Uncertainty Rules supposition. Utilizing a business management theory frame of reference led to the It's The Business Model (, Stupid)[9] hypothesis. Given the goal of

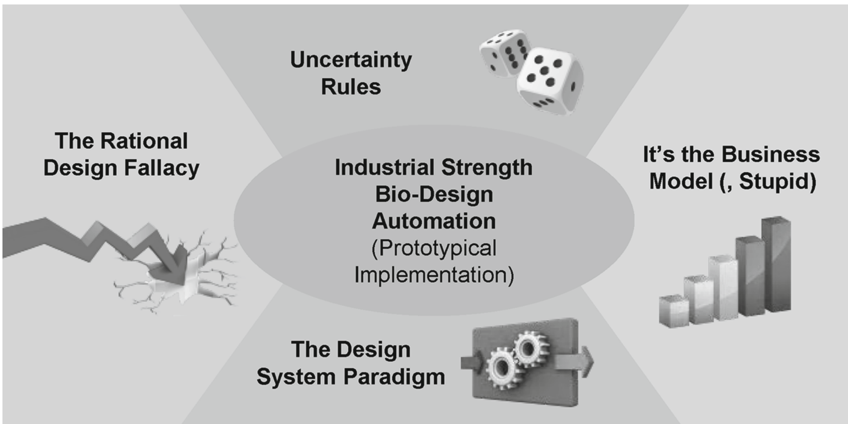


Fig. 2. Graphical abstract for the Table of Contents of this paper

[9] The labeling of the business related core hypothesis as The Business Model (, Stupid) hypothesis will be explained in Sect. 5 of this paper.

stimulating a community discussion, we will keep the following discussion at the conceptual level, the rational or the basics for what it takes to create an industrial strength BDA industry. Getting the concepts right for an emerging discipline is difficult, but also absolutely essential. Note that the authors come from four different perspectives: general purpose ICT, BDA industry, BDA research, and synthetic biology. The aim of the conceptual discussion is to integrate these different perspectives into a single framework for industrial strength BDA. For practical purposes this framework has to be translated into concrete methodological and implementation agendas. Therefore, we will present a prototypical implementation of a BDA platform in the final section of this paper.

Figure 2 provides a graphical representation of the table of contents of this paper.

2 The Bio-design System Paradigm

Let us define a bio-design system as the totality of resources required to perform synthetic biology design processes, with the goal to transform bio-design specifications into bio-production designs. Note that, in general, design systems are holonic systems [13]. Holonic systems are composed of distinct entities (called holons) which interact with each other in order to achieve global system goals. Self-organization is a key property of holonic systems. This allows a holonic system to evolve over time to optimize the achievement of its system goal. Let us recall that a paradigm is a general perspective or way of thinking that reflects the fundamental beliefs and assumptions about the nature (principles) of the subject under consideration [16]. The Bio-Design System Paradigm hypothesis is then the idea that a bio-design system is a holonic system that represents the frame of reference for all synthetic biology engineering activities. Thinking in bio-design systems terms is perhaps the key to industrial strength BDA. In fact, experiences – both good and bad - from established design automation industries (and beyond) underpin the imperative to take a system approach when developing design systems. Indeed, out of our four hypotheses, the Bio-Design Paradigm hypothesis might be the most important one.

Present-day synthetic biology is still largely devoted to solve fundamental (and persistent) problems of engineering biological artifacts [14]. This might lead to the impression that The Bio-Design System Paradigm hypothesis may be far beyond any practical relevance. But perhaps the opposite is true. In fact, the potential benefits of applying this paradigm can be numerous. To discuss this in more detail it is necessary to refine the definition of a bio-design system. As Martin and Odell stated [25]: “If we wish to build systems that perform correctly and consistently, we will want to be clear, concise, and unambiguous in our specifications” and concepts.

Figure 3 proposes a hierarchical definition of a bio-design system. At the highest level the system is treated as a whole, accepting design specifications and producing production designs (forward engineering) or vice versa (backward/reverse engineering). The next level is comprised of the design environment and synthetic biology engineering. Synthetic biology engineering denotes the team of synthetic biologists that perform the design processes. Synthetic biologists are human resources and are therefore carriers of highly valuable tacit knowledge. Tacit knowledge is defined by

Polanyi as knowledge that cannot be expressed explicitly in (whatever) codified form – "We know more then we can tell" [33]. It is work-related practical knowledge acquired informally through experience on the job [42]. Tacit knowledge can be passed between engineers "by personal contact but cannot be, or have not been, set out or passed on in formulae, diagrams, or verbal descriptions and instructions for action" [9]. Note that tacit knowledge constitutes often a company's decisive competitive advantage. Consequently, a bio-design system is not a purely technical system, but a socio-technical system where human, organizational and technical factors closely interact and evolve. Therefore the role of BDA is not solely one of providing automation tools. The BDA challenges faced in the context of bio-design systems are both technical and social in nature. In fact this holds true for any domain-specific ICT. This is important to acknowledge and understand.

Back to the bio-design system hierarchy, the next level down in the bio-design system hierarchy details the composition of the design environment. The function of the design environment is to enable and support the design of biological artifacts. It is composed of the design infrastructure and design environment engineering. The design infrastructure encompasses the *in silico* and the wet lab function. Note that the design infrastructure embodies the explicit knowledge of the bio-design system. This knowledge represents the antithesis to tacit knowledge. Explicit knowledge is highly codified and is easily transmittable [29]. It comes in a wide variety of codifications such as books, documents, policy manuals, standards, intellectual property, databases,

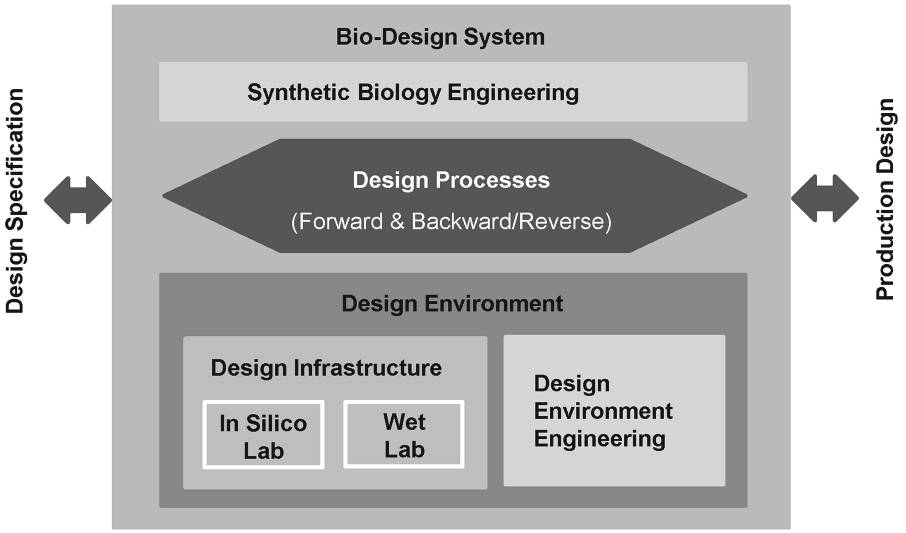


Fig. 3. The *bio-design system* hierarchy. *Synthetic biology engineering* and *design environment engineering* represent human resources who are the carriers of the *bio-design system*'s tacit knowledge. The *design infrastructure* represents the explicit/codified knowledge of the *bio-design system*. *Design processes* are supported bi-directionally (forward and backward/reverse). BDA technology is localized in the *in silico lab*. BDA engineering is part of *design environment engineering*.

software tools, wet lab equipment etc. Similarly like synthetic biology engineering, design environment engineering represents human resources that act to as carriers of tacit knowledge. All the synthetic biology engineering tacit knowledge remarks apply without limitation. Design environment engineering fulfills two major functions: first, it supports synthetic biology engineering in the application of the design infrastructure; second, it codifies its own and synthetic biology engineering's tacit knowledge into explicit concepts and therefore creates the basis for the future evolution of the bio-design system.

Based on the above we are ready to discuss some potential high-level practical benefits of applying The Bio-Design System Paradigm hypothesis:

(a) The Tower-of-Babel problem denotes the potential communication problem between experts that come from different disciplines and speak their own domain-specific languages. This impedes the process of establishing a shared understanding among these experts and hampers collaboration. Cross functional disciplines, such as synthetic biology and BDA, are especially prone to this problem. And indeed, as of now it seems that the BDA community started to build its own Tower-of-Babel. For example there is already a competing set of terms for BDA, namely bioCAD [18], BioCAD [4], Genetic Design Automation [19], etc. - more to come? This might be not an issue for the BDA community, but it is clearly a way to confuse synthetic biologists (as experienced by some of the authors multiple times). The Bio-Design System Paradigm provides the foundation to develop a shared, agreed upon and committed vocabulary that can be used in a coherent and consistent manner by all stakeholders, a prerequisite for true collaboration. One ongoing effort focusing on developing such a vocabulary is the BioParts Terms initiative[10];
(b) Performance management is a prime management concern in any industrial setting [27]. Note that productivity is the single most important performance measurement. The proposed bio-design system hierarchy provides a unifying framework within which the various resources (incl. and most importantly human resources) and processes of design can be situated and their roles and relations can be explicitly identified. Such a framework is pivotal for any performance management and measurement beyond the strategic level [12];
(c) Transforming tacit into explicit knowledge is key to the advancement of a bio-design system. This process needs to be actively managed. The proposed bio-design system hierarchy allows identifying the flow of knowledge and taking concrete actions to facilitate the transformation process. Although it might be a truism to some people, it needs to be stated that the knowledge transformation requires very tight collaboration between synthetic biology engineering and BDA engineering;
(d) Most of the literature defines BDA in the narrow sense that is purely technical. As such BDA is a technology situated in the *in silico* lab. However, given the central role assigned to be BDA in the process of siliconizing synthetic biology (the BDA mission) this is perhaps a too narrow definition. BDA in the broad sense

[10] http://www.minres.com/wiki/index.php/BioParts_Terms

encompasses technical and human resources. BDA is connected to every element of a bio-design system;

(e) The proposed bio-design system structure can serve as starting point for the development of a BDA roadmap. Such a roadmap is important to achieve industrial strength BDA for two reasons: first, it serves as an agreed-upon requirements document between the synthetic biology engineering community and the BDA community; second, it gives BDA researches and practitioners alike direction where to focus on to develop and deliver high-impact solutions. This will become even more important as the BDA community grows.

Some might consider The Design System Paradigm hypotheses still a "philosophical" issue. However, we believe that ignoring this paradigm will have the effect of making the collaboration between the communities of synthetic biology engineering and BDA engineering much more difficult, no matter how pragmatic it might seem to ignore this "philosophical" issue. It can be expected that the recently formed Bio-Design Automation Consortium (BDAC), a non-profit association which also organizes the IWBDA events, will take a leadership role in addressing and promoting the above benefits of bio-design systems[11].

3 The Rational Design Fallacy

Synthetic biology is founded on the notion of rational design, which implies the following propositions [32]:

(a) Design processes (what to do) and design methods (how to do it) are known explicitly and in detail;

(b) Given a well-defined goal, the design process progresses along the optimal path on the basis of logical principles.

Clearly, neither of these propositions will pass the reality test, which renders rational design a utopian vision. This is not a bad thing in itself, provided it is recognized for what it is: a guiding dream with positive transformative significance. However, if rational design is used as a mental model that directs practical actions [15], then it can turn straight into a fallacy. In fact, all design activities are rationally bounded [38].

The major pitfall for BDA caused by this fallacy is over-automation. The labeling of this pitfall is meant to express the respective BDA experience of synthetic biologists. The over-automation pitfall occurs when the level of abstraction for a design automation is raised beyond the essence horizon for the sake of automation: You cross the essence horizon if your abstractions alienate the substance of the system to be designed. Established classical engineering domains are as a rule relatively self-contained and mature. They offer defined representations of both the design artifact under development and the environment it is supposed to operate in. Therefore, it is possible to build comprehensive domain models that enable and support model-driven design approaches

[11] http://www.iwbdaconf.org/2014/

which proceed more or less smoothly from architectural to physical design. However, the situation in synthetic biology is completely different. Even so it is possible to create (to some extent) design artifacts following classical engineering approaches the overall synthetic biology experience is that there is no guarantee that the design artifact will operate as specified in the target environment. This is because bio-designs exhibit significant context-dependency and low predictability. Perhaps this is a prime challenge for BDA to focus on. Currently, the synthetic biology engineering way to cope with this challenge is to follow a minimum design approach.

Figure 4 details the above discussion visually. Every design process starts with a design specification which describes the initial design space including design metrics and constraints. The design progresses further along defined levels of abstractions where the design space is narrowed by proceeding from a higher level of abstraction to a lower one. Here, the biological system to be designed is a gene-regulatory network (GRN). Note that a GRN is a collection of deoxyribonucleic acid (DNA) segments in a cell which interact with each other indirectly - through their ribonucleic acid (RNA) and protein expression products - and with other substances in the cell[12]. Figure 4(a) shows an exemplary synthetic biology design process. The design proceeds from the design architecture, perhaps captured by a functional block diagram, to the detailed design, perhaps captured by the network topology, to the design implementation in form of abstract gene-regulatory network (AGRNs) to the physical design, represented by physical a GRN, and, finally to the assembly plan. Figure 4(b) captures the rational design (illusion): Rational design progresses straight from the initial design space captured in the design specification to the optimal production design along the optimal design path on the basis of purely logical decisions. That is, the designer is only concerned with the design specification. Everything else is done fully automatically and nontransparent (black-box) to the designer. The final production design would be optimal by definition. Of course, rational design does not pass the reality test. Figure 4 (c) represents the design process based on the bounded rational design methodology: All design activities are rationally bounded and transparent (no automations) or non-transparent (automations) to the designer. Design automations are only introduced when appropriate and their scope is fully described (to ensure that they are essence-preserving). All other activities in the design process are considered to be fraught with uncertainty and ultimately dependent on the tacit knowledge of the synthetic biologists. In general the bounded rational design process is iterative. The final production design will be acceptable (that is as a rule not optimal). Figure 4(d) depicts the concept of essence preserving design automations: A design automation introduced in the design process must comply with the entry essence horizon and the exit essence horizon of the design activity it automates. In practical terms an essence horizon is a validated design space at certain level of abstraction. Validated means that the design space complies with design metrics and constraints along the hierarchy of abstraction down to the final product. This way it is ensured that the essence of the system to be designed is maintained. A design space at a certain level of abstraction might very well fulfill the design metrics and constraints at this level of abstraction. However, if it fails to fulfill

[12] http://en.wikipedia.org/wiki/Gene_regulatory_network

the design metrics and constraints down the hierarchy, it does not constitute an essence horizon. Last, but not least, note that the exit essence horizon of an automation constitutes the entry essence horizon of the next design activity down the abstraction hierarchy. Figure 4(e) visualizes the over-automation pitfall: As explained above this pitfall occurs when the level of abstraction is raised beyond the essence horizon for the sake of automation. The scenario is as follows: Given a bounded rational design process, a design automation is introduced at an abstraction level which is not appropriate for automation. Such a design automation will prevent the synthetic biologist to make necessary engineering decisions based on his or her tacit knowledge. The necessity for engineering decisions is a consequence of having precisely not enough explicit knowledge to justify a design automation. The essence horizon compliance requirement is violated at the entry level and therefore produces results that constitute a

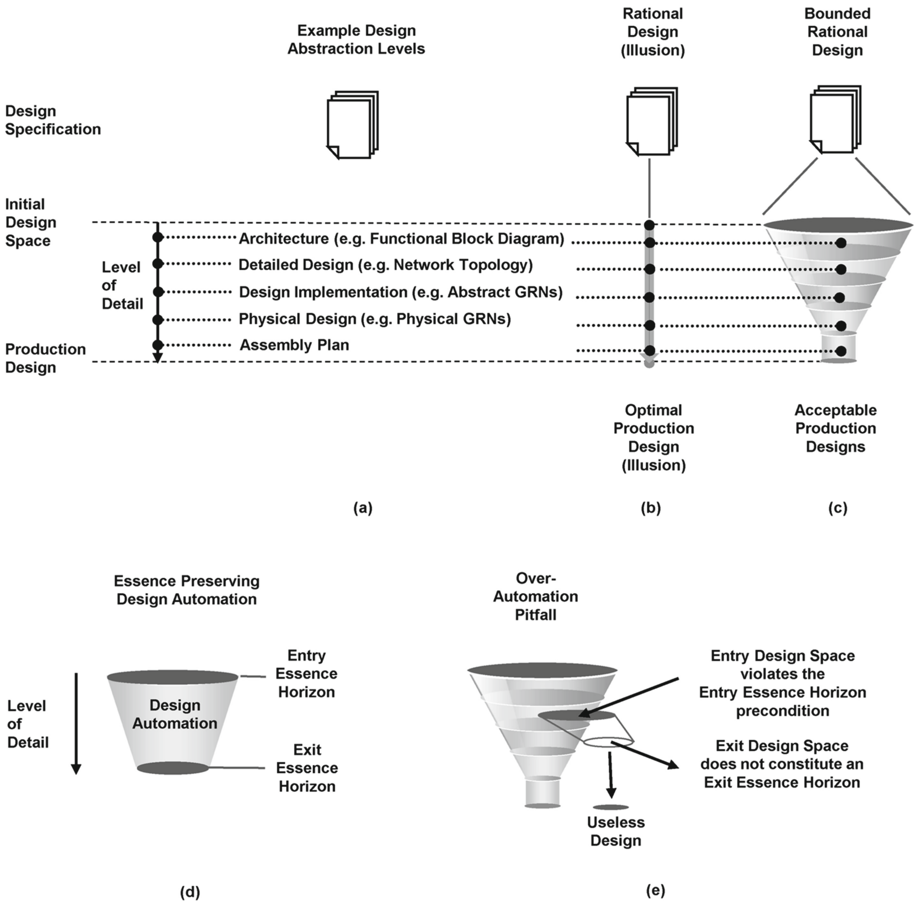


Fig. 4. Bio-design process. The system to be designed is gene regulatory network (GRN). (a) Example design abstraction levels. (b) Rational design illusion. (c) Bounded rational design. (d) Essence preserving design automation. (e) The over-automation pitfall.

design space which is off from the exit essence horizon. As a consequence the resulting "production designs" will be almost guaranteed to be useless.

To avoid the over-automation pitfall and others industrial strength BDA should adhere to the following principles and guidelines:

(a) Focus on the development of solutions for today's synthetic biology problems. Of course, there also need to be investments in longer-term R&D efforts that may not mature for years;
(b) Respect that the (baseline) design processes are established by synthetic biologists. This implies that synthetic biologists define the essence horizons. Indeed, in several instances this might bring some high-flying BDA aspirations down to earth;
(c) Support both forward and reverse engineering. Note, that in today's synthetic biology settings, reverse-engineering dominates;
(d) Given a set of essence horizons support the right levels of remaining computational abstractions along the design process;

Last, but not least, the perhaps most important rule is this:

(e) If your automations do not improve the BDA experience for synthetic biologists', start all over again!

4 Uncertainty Rules

This section emphasizes that BDA technology must be capable to cope with uncertainty. Generally, uncertainty is a second order effect in many classical engineering design settings, and is mostly related to concerns such as product yield, product lifetime, etc. The opposite is true, although often not explicitly acknowledged, for synthetic biology. Quantum physics aside, the world is in principle knowable and deterministic [11]. It follows that uncertainty is due to ignorance of the underlying causes. Ignorance is currently ubiquitous in synthetic biology, and it does not seem that this situation will change in the near future. Not surprisingly, the prevailing sentiment in the world of synthetic biology is that the inherent complexity of the field largely prevents the use of computational design approaches. This position caused a considerable debate between the BDA and the synthetic biology community [10]. Unfortunately, discussants often seem to talk at cross-purposes when they might well be seeing the same problem from different angles.

To arrive at some level of consensus among the communities consider the following:

(a) If we interpret complexity as a function of our ignorance about the reality's work principles, then synthetic biology is devilishly complex;
(b) Complex systems are inherently uncertain, that is stochastic, in nature.

Both statements should be viewed as consensual propositions. The immediate conclusion is that stochasticity is an essential property of biological systems. Indeed, more recent research has shown that stochasticity is a generic feature of many biological systems, and may, in fact, be actively exploited by ensuring heterogeneity in a population of cells. Understanding the causes – in addition to the effects – of stochasticity is thus pivotal for applications in synthetic biology. Statistical inference provides us with the computational tools necessary to dissect noise and stochasticity in biological systems; such a statistical angle is essential, as we are no longer dealing with a single type of behavior but with a probability distribution over different (though related) types of behavior; this poses severe challenges to e.g. conventional optimization procedures.

Let us consider the design of biochemical reaction network (BRN) according to some stated design objectives. A BRN is defined by (e.g. [31, 40]):

(a) A set of variables which represent the amounts of biochemical species (molecules) in an reactor under consideration;
(b) A set of rules of temporal changes (reaction equations) of these variables.

A design objective might be for example the specified change of these variables over time. Let us further assume we have identified a set of potential designs, but we don't know the one that best meets the design objectives. Let us finally assume we have captured these designs by some appropriate modeling formalism. However, we face an additional challenge. In practice, it will be the rule rather than the exception that we have to cope with parameter uncertainty, often quite considerable (i.e. spanning orders of magnitude). The task is now to identify the design that best meets the design objectives (model selection) under conditions of parameter uncertainty (parameter inference). This requires an inference engine that is capable to handle model selection and parameter inference. Conceptually the inference engine needs to "sit" on top on whatever computational model we apply. Figure 5 depicts the positioning of the inference engine in the design process.

Unquestionably, it is clear that for any industrial strength BDA solution an inference engine is a prerequisite to cope with stochasticity. One superior inference engine (both in terms of conceptual and computational performance) is the Approximate Bayesian Computation – Sequential Monte-Carlo (ABC SMC) algorithm presented in [1]. The algorithm supports both forward- and reverse engineering and is flexible with respect to the underlying computational models. As long as appropriate computational models are available, it is possible to apply the algorithm at any design abstraction level (architecture, detailed design etc.). Perhaps the most important feature is that the algorithm can handle parameter uncertainty and model selection at once.

Figure 6 visualizes a typical design scenario in synthetic biology that is the design of a BRN, as discussed above. In this scenario the synthetic biologist has chosen to enter the design process at the network topology level. Biochemical adaptation is used as an example [24]. This example was investigated thoroughly in one of the authors' research group (see [1, 2]). The inference engine exploits the advantages of Bayesian statistics. The applied Approximate Bayesian Computation Sequential Monte-Carlo (ABC-SMC algorithm) is based on [2]. Parameter inference is accomplished by the efficient exploration of design and high-dimensional parameter spaces. Model selection, Fig. 6(a), is enabled by the ability to rank competing designs with respect to their ability to bring about the desired behavior. The synthetic biologist identified several design options (network realizations) which in principal will be able to implement the

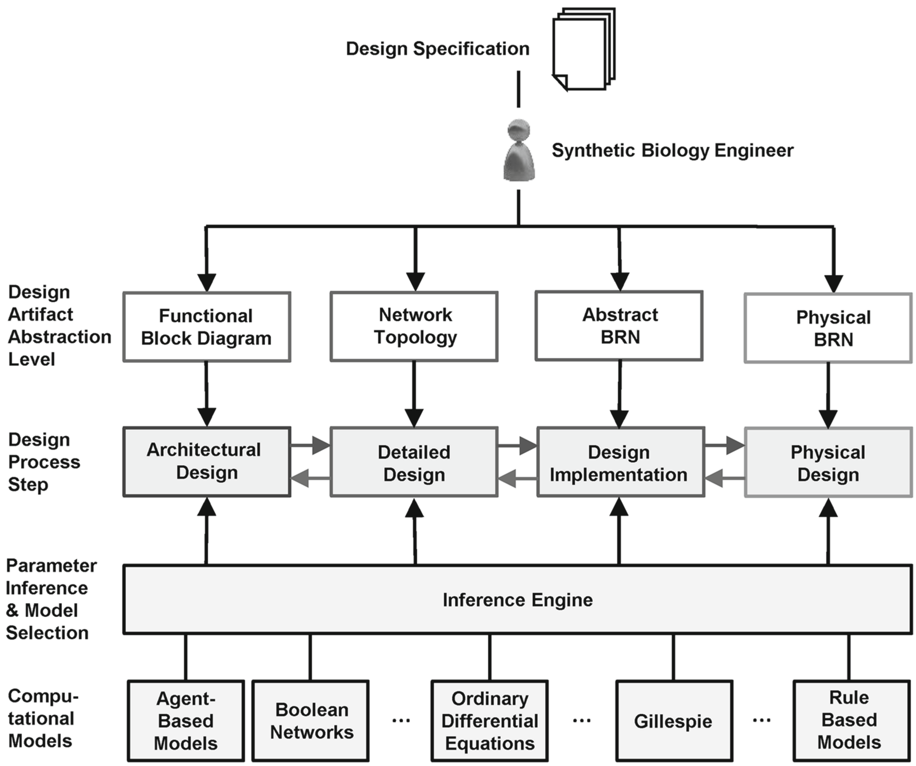


Fig. 5. Inference Engine. Given the *design specification*, the synthetic biologist chooses the appropriate design entry level among the available design artifact abstractions. In general it can be assumed that there are different design options available, which are represented by different design models. It is further assumed that these models can be represented in a computational form. For this, there exist several computational approaches, such as *agent based models*, *Boolean networks*, *ordinary differential equations*, etc. Furthermore, it is usually the case that these models will contain parameters with uncertain values. The *inference engine* allows to infer the parameter values and to select the model which has the highest probability to fulfill the design specification.

design objectives [24]. Computational models (deterministic or stochastic) are developed for each design model. Each model contains a set of uncertain kinetic parameters θ and associated prior distributions π (θ |M) on the parameters. A distance function ρ (y, O) relates model output, y, to the desired output characteristic O. The design space is explored using Sequential Monte Carlo (SMC). The desired behavior is more accurately approximated with each new population. The ability of each design to achieve the specified behavior is represented by the model posterior p (M|D). In the shown case, Model M3 encodes the specified behavior with the highest probability. Eventually, the parameter posterior p (θ |M, D) allows to identify parameters that are sensitive or insensitive to the targeted behavior.

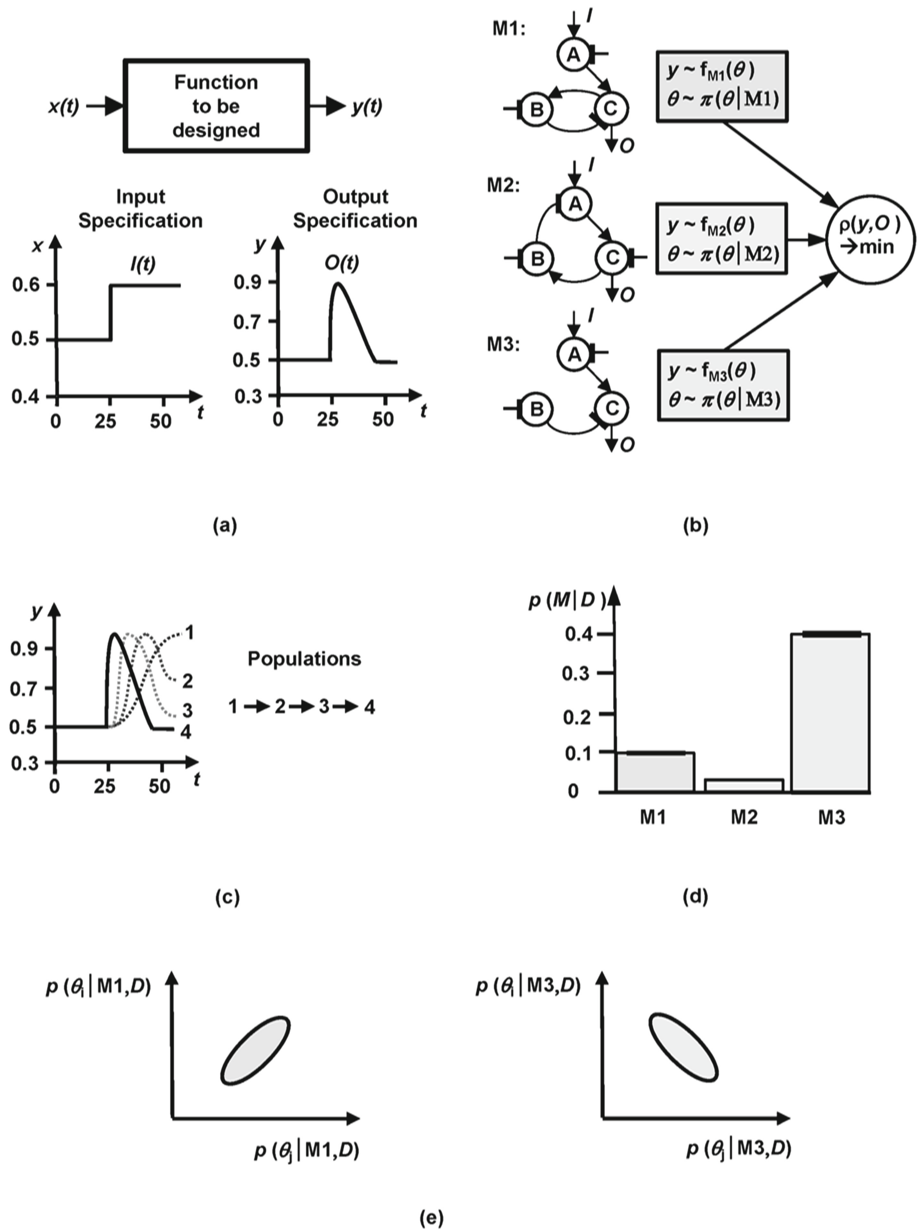


Fig. 6. Design of a biochemical adaptation function. Adopted from Fig. 1 in [1] with permission of the copyright owner (PNAS - Proceedings of the National Academy of Sciences of the United States of America). (a) Specification and encoding of the design objectives. (b) Design entry of competing designs at the network topology level. (c) Design space exploration. (d) The model posterior *p* (M|D). (e) The parameter posterior (sensitivity analysis).

5 It's the Business Model (, Stupid)

As clearly pointed out in [26], there ought to be no engineering discipline without economics. "It's the economy, stupid" was the famous 1992 U.S. presidential campaign slogan of Bill Clinton that helped him get elected. The slogan was meant to emphasize the economy as the number one issue to be addressed. Paraphrasing this motto, the title of this section aims to draw attention to the fact that business models are of prime concern for industrial strength. It is worth mentioning that the intrinsic relation between industrial strength and business models matches most people's basic intuition. However, it might not be obvious to everyone in the BDA community why he or she should care about business models, which is in most cases a distant concept for engineers and researchers alike. After all engineers develop technical artifacts and managers are supposed to develop business models. The straightforward answer is that businesses are the foundation of any professional activity. The more specific answer is that BDA is in its early formative stage, where everyone has an important role in defining the trajectories for tomorrow's business models, knowingly or unknowingly, willingly or unwillingly. Today's dominant BDA business model is public funding. Clearly, this is an unsustainable model. We already witness much of the "free" software developed in public funding settings end up as "Internet-Zombies". Nevertheless, the public funding model defines the starting point and direction of trajectories to future business models.

Let us recall the popular "There's no such thing as a free lunch" adage, which clarifies that is impossible to get something for nothing. That is, for BDA to become and stay industrial strength there needs to be adequate and sustained funding. This presupposes that BDA solutions are enabled, supported and, maintained by governance models which ensure the continuity of funding. A governance model is defined as a set of policies and practices that outline the responsibilities of the stakeholders (providers and users).

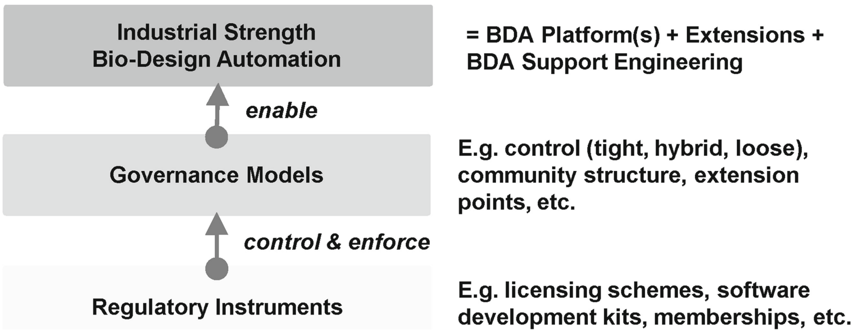


Fig. 7. BDA industrial strength business model framework. *Industrial strength BDA* is represented by the availability of *BDA platforms* with appropriate *extension* points for third party contributions and *BDA support engineering* function that endorses maintains and develops these platforms and their extensions. *Governance models* and *regulatory instruments* play a key role in ensuring the sustained availability of industrial strength BDA. This requires a well-orchestrated interplay between *governance models* and *regulatory instruments*.

It is probably safe to proceed on the assumption that governance models that enforce BDA solutions that are closed, highly complicated and, proprietary and which are controlled by a few vendors should be avoided by all means. That is perhaps a key lesson to learn from established design automation industries. With reference to Fig. 7, Industrial strength BDA needs to encompass (non-proprietary) BDA platforms with easily accessible extension points for third parties as well as an active BDA engineering function responsible for development and support of the community. Such a function will certainly be distributed over several companies, research groups etc. Governance models come in multiple forms, such as tight, loose and hybrid control. Other properties of a governance model are the community or stakeholder structure, the openness for extensions by third parties and so on. For more details the reader is referred to [30]. Governance models need to be enforced by regulatory instruments to be sustainable. Regulatory instruments are based on the use of some form coercive power. However, this does not mean that regulatory instruments are limited to legal enforceable instruments (such as license schemes). In fact, there is a wide range of managerial and technical regulatory instruments available, including mandatory development processes, software development kits, membership schemes, normative authority of the platform owner and many more. Note, that different types of regulation entail different models of governance. It is now the right time to start a discussion in the BDA community about sustainable innovative business models. The provision of open source solutions will probably have to be a key consideration in these discussions.

In any case, the BDA community should be open-minded about any and every possible business model proposed. For example, consider the single-vendor commercial open source business model. Prima vista, this sounds like an economic paradox. However, as explained by Riehle [34, 35] this turns out to be a superior business model in certain settings. If the BDA community does not manage to develop sustainably business models, we might very well end up in an unproductive open source jungle, but we will not arrive at industrial strength BDA. After all, the success or failure of BDA is a question of the right business models and their ability to adapt to changes.

6 Mendel – A Prototypical BDA Platform

The aim of this section is to present the prototypical BDA platform Mendel[13]. We take the position of a start-up aiming to develop and market a BDA solution. While reading this section, the reader is invited to consider himself or herself to be part of this start-up endeavor. In this way, it becomes perhaps more personal and relevant to the reader without losing the big picture. Note that a prototype is a small-scale working model of the final system, built to develop and test design ideas. To this end, we introduce the notion of minimum viable BDA product. In general, “a minimum viable product (MVP) is a strategy used for fast and quantitative market testing of a product or

[13] Mendel is the registered trademark of the BDA platform of MINRES Technologies GmbH. The name was chosen in honor of Gregor Mendel, the father of genetics.

product feature."[14] That is, our start-up follows the lean-start-up methodology. This methodology "favors experimentation over elaborate planning, customer feedback over intuition, and iterative design over traditional 'big design upfront' development" [5][15]. There are two important points to be made here. Ryan Dancey, the CEO of Goblinworks, Inc.[16], concisely expressed them recently as follows[17]:

(a) "The point of an MVP is not to make one and stop work. The point is to make one and use it as the very first point of interaction with real customers, and then start iterating on the design by tightly integrating the users' feedback into the development of new features and expansion of existing features."
(b) "A 'minimum viable product' is not a 'minimum' product. The key word is 'viable,' not 'minimum.'"

Clearly, the prototype must conform to the minimum requirements imposed by our four core hypotheses. It is beyond the scope of this paper to present and discuss the complete set of requirements. Here we will focus on some exemplary requirements from which we are able to develop our BDA prototype.

Let us start with a business and design system view. The backbone of the synthetic biology industry will be formed by small and medium enterprises (SMEs). Consequently the costs for introducing and maintaining BDA solution for a synthetic biology SME must be within the means of such a company. Our research shows that costs in the upper three digit, lower four digit price range seem to be acceptable for the combined hardware and software cost prices. In any case, a well-conceived product tiering[18] strategy is ab initio required. This in turn demands a highly scalable product architecture. Further, it needs to be highlighted that the confidentiality requirements of biotechnology companies are very strict. In fact, they prohibit to utilize recent ICT trends such as cloud computing, web-scale IT, mobile apps and application, etc. But also note that openness and confidentiality in a system context are not necessarily conflicting. Last, but not least, customers are increasingly concerned about the vendor lock-in problem. In this scenario, "the coupling between the customer and the provider become so high to a point that it is no longer economically viable to move from one provider to another" [3]. The imperative overarching requirement following from the above is openness. From a business point of view the openness requirement is represented by the requirement for open business models. Readers who are interested in more details and comprehensive descriptions of open business models are referred

[14] http://en.wikipedia.org/wiki/Minimum_viable_product

[15] Lately, the MVP approach and the lean-start-up methodology became very prominent among high-tech start-ups.

[16] https://goblinworks.com/, Goblinworks, Inc. is a start-up that develops Pathfinder, a massive-multiplayer online game.

[17] http://massively.joystiq.com/2014/03/03/pathfinder-onlines-ryan-dancey-on-crowdforging-a-minimum-viabl/

[18] "Product tiering is a pricing structure that is … used by producers, in which" customers "are segmented by willingness" and ability "to pay for specific (added) product benefits." See: Breetz C (2014) Product Packaging as Tool to Demand a Price Premium: Does Packaging Enhance Consumers 'Value Perception to Justify a Price Premium. Anchor Academic Publishing.

to [21]. From a design systems perspective the openness requirement translates into the requirement for open design systems (ODSs). ODSs are characterized by the reuse open source software whenever it is feasible (technically and legally), open standards, a platform approach, etc. Readers who are seeking a more thorough understanding of ODSs are referred to [8].

Next we look on The Rational Design Fallacy hypothesis. This hypothesis requires us to ensure that the design automations we introduce are essence preserving. This can only be accomplished by close collaboration of synthetic biology engineers and BDA engineers. The requirement is therefore to ensure that an appropriate collaboration process is in place and is continuously exercised. Such a process is highly iterative and interactive. It is instructive to note that the MVP approach and the lean-start-up methodology are perfect instruments to ensure the accomplishment of this requirement.

The key requirement of The Uncertainty Rules hypotheses is straightforward. Any BDA solution needs to provide a computational inference engine and the capability to deal with stochasticity.

Let us now discuss some implementation aspects of our prototype BDA platform Mendel. For this purpose we use the notion of a minimum viable design process based on the MVP concept. Here we consider the case of the design of BRNs. From a technological point of view, a BDA product is comprised of a hardware and a software part. The software part can be implemented straightforward based on a few design decisions. We decided to use Eclipse [39] as the basis for our BDA platform. Some of advantages of Eclipse are reusability, trustworthiness, confidentiality, quality, clarity, longevity, flexibility, and rapid development on the basis of a plug-in system. Figure 8 shows the minimum valuable design process for the design of BRNs and the Mendel GUI along the stages of this design process.

By their very nature stochastic computations are highly compute intensive tasks requiring significant computational resources. Practical cases will therefore principally require the use of massively parallel algorithms which in turn require an appropriate and affordable (!) high-performance computing (HPC) hardware. To this end we have analyzed the feasibility of various hardware acceleration approaches[19]. These days, there are three cost-effective different hardware acceleration options:

(a) GPU based HPC accelerators (NVidia Tesla, AMD Firepro),
(b) Many-Core based HPC accelerators (Intel Xeon Phi).
(c) Reprogrammable logic chips (FPGA - field programmable gate arrays).

Performance figures for the first (a) and (b) are known while for the latter approach no re-ports can be found. Recent developments allow now to transform a high-level behavioral description, e.g. in SystemC[20], into configuration information. Thus FPGAs can be used as highly configurable application accelerators executing simulation and analysis algorithms directly while consuming a magnitude less power. From a

[19] As the intended products are targeted mainly to SMEs, super computers are not a valid attempt for financial reasons.

[20] SystemC is a set of C++ classes and macros which provide an event-driven simulation interface in C++; http://en.wikipedia.org/wiki/SystemC.

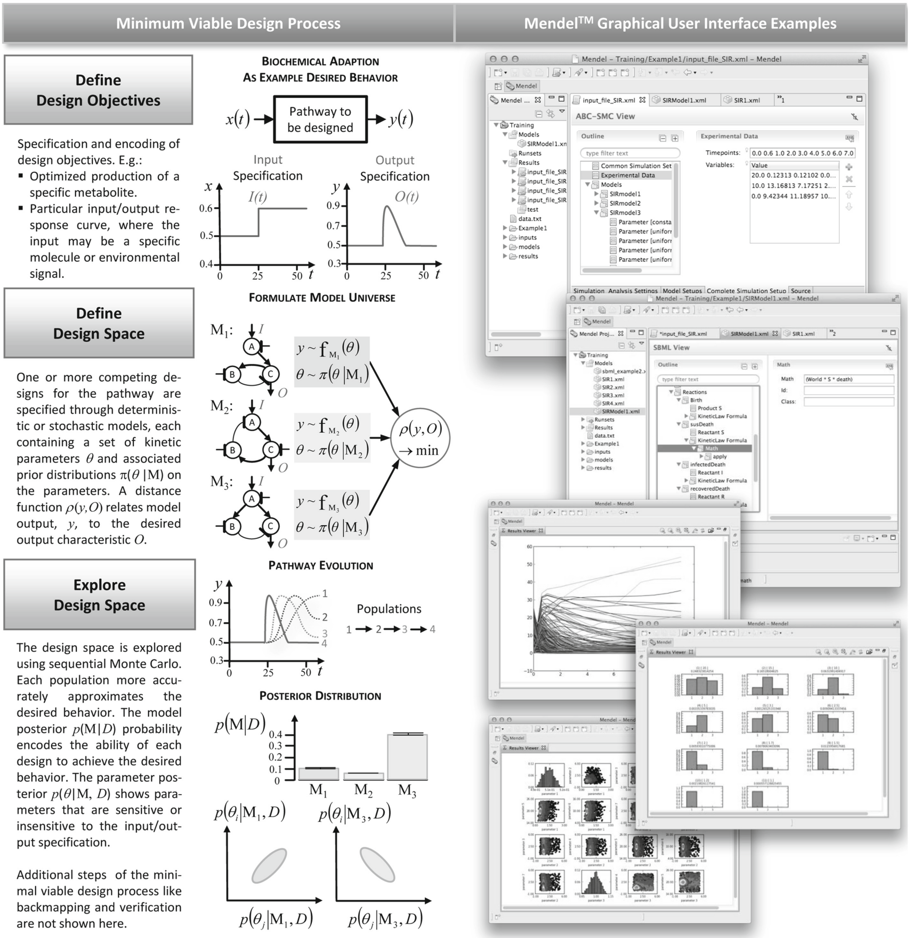
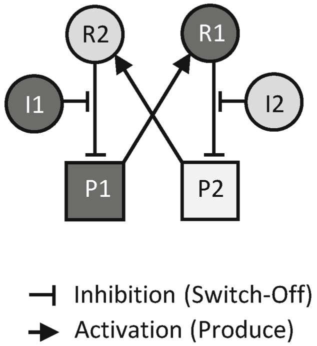


Fig. 8. Mendel GUI along the stages of a minimum viable BRN design process

Table 1. HLS Latencies. Duration of HLS as function of the unrolling factor[a]. HLS: High-Level Synthesis. ODES: First-order Ordinary Differential Equation System.

Unrolling factor	12	4	∞[a]	0
Duration of HLS	21 m47.418 s	2 m59.423 s	1 m52.367 s	1 m52.503 s
Time to solve a single ODES	868,19 µs	869,81 µs	868,54 µs	868,54 µs

[a]Unrolling, is a loop transformation technique that attempts to optimize a program's execution speed at the expense of its binary size. The unrolling factor can be controlled manually by the programmer or chosen by the software only (∞).



Toggle Switch Operation		
I1	**I2**	**Action**
0	0	No Change
0	1	R2=1
1	0	R1=1
1	1	Restricted Combination

For i=1,2
Pi: Constitutive Promoter i
Ri: Repressor i
Ii: Inhibitor I

(a)

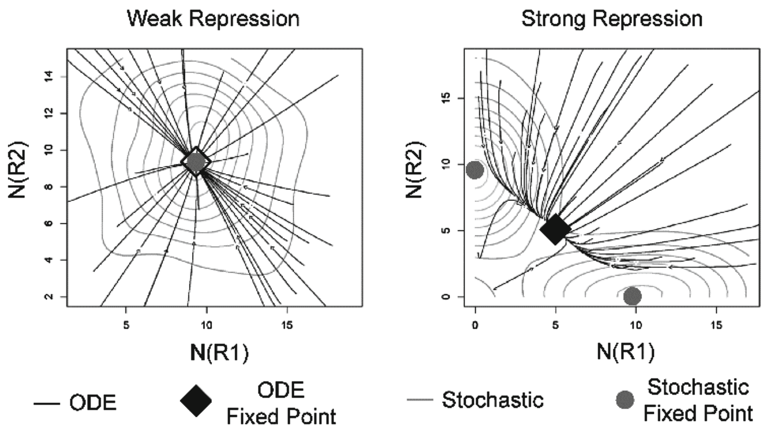
$$\frac{du}{dt} = \frac{\alpha_1}{1+v^{\beta}} - u$$

$$\frac{dv}{dt} = \frac{\alpha_2}{1+u^{\gamma}} - v$$

u: Concentration of repressor 1
v: Concentration of repressor 2
α_1: Effective rate of synthesis of repressor 1
α_2: Effective rate of synthesis of repressor 2
β: Cooperativity of repression of promoter 1
γ: Cooperativity of repression of promoter 2

$\beta, \gamma > 1$: Cooperativity; $\beta = \gamma = 1$: No Cooperativity

(b)



(c)

Fig. 9. Genetic toggle switch. (a) Schematic and Boolean operation table. (b) First-order differential equation system governing the dynamic behavior of the toggle switch. (c) Simulation results.

performance perspective our results show that FPGAs can compete with both GPU and Many-Core accelerators, while yielding higher power efficiency. However, one issue prevents the use of FPGAs in current solutions. Simulation and analysis are highly iterative and interactive tasks. Latency from starting an analysis until the first computation starts is governed by the runtime of the synthesis and mapping process. Currently this latency ranges from 2 to 20 min which is not acceptable in an interactive design process. Table 1 summarizes some results of the feasibility study. Recent announcements of FPGA vendors reveal a promising convergence towards OpenCL as lingua franca being supported by vendors of GPU as well as FPGA based solutions. This would allow to run the same algorithm description on different acceleration platforms utilizing their specific advantages (low latency, high efficiency) in a hybrid solution. However, currently we use a HPC-GPU solution.

Finally we want to present the results of analyzing a genetic toggle switch using our prototype platform Mendel. Toggle switches belong to the class of genetic circuits which received considerable interest in the synthetic biology community, e.g. [22]. Figure 9(a) presents a schematic of a toggle and the associated Boolean operation table.

The switch is composed of two repressors R1, R2 (with concentrations u, v) which negatively regulate each other's production. More details can be found in [22]. With I1 = S, I2 = R, R1 = Q, and R2 $= \overline{Q}$ it can be easily seen that the toggle switch represents a SR latch. Figure 9(b) presents the ODEs governing the dynamic behavior of the switch. Based on this, it was assumed that for the circuit to operate as a toggle switch strong repression and the cooperative binding $(\beta, \gamma > 1)$ is required. In the case of no cooperativity there should be no bistability (see Fig. 9(c)). However, if one uses a stochastic approach, the bifurcation diagram for strong repression and no cooperativity shows two fixed points, that is the switch is bi-stable. Without a stochastic approach this behavior would not have been revealed [22].

7 Conclusions

The four hypotheses of this paper aimed at highlighting what we believe are the most important current challenges for establishing industrial strength BDA, the domain-specific ICT for synthetic biology. Any BDA solution should comply with the requirements imposed by these hypotheses. There might be other important concerns that we have not touched. We ought to hear about them. We might have covered some aspects insufficiently or inaccurately to some reader. We want to learn about those. But most importantly, and in spite of these conceivable gaps, this paper will hopefully foster a broad community discussion about industrial strength BDA.

Acknowledgements. The HPC hardware related work, referred to in this paper, is supported in part by the Bavarian Ministry of Economic Affairs, Infrastructure, Transport and Technology, Grant No. TOU-1212-0002. We wish to thank particularly Vadim Ermolayev for encouraging us in writing this paper and for his stimulating discussions on the topic.

References

1. Barnes, C.P., Silk, D., Sheng, X., et al.: Bayesian design of synthetic biological systems. Proc. Natl. Acad. Sci. USA **108**, 15190–15195 (2011)
2. Barnes, C.P., Silk, D., Stumpf, M.P.H.: Bayesian design strategies for synthetic biology. Interface Focus **1**, 895–908 (2011)
3. Bessani, A., Correia, M., Quaresma, B., et al.: DepSky: dependable and secure storage in a cloud-of-clouds. Trans. Storage **9**, 1–33 (2013)
4. Bhatia, S., Densmore, D., Pigeon, A.: Design visualizer for synthetic biology. ACS Synth. Biol. **2**, 348–350 (2013)
5. Blank, S.: Why the lean start-up changes everything. Harvard Bus. Rev. **91**, 63–72 (2013)
6. Buchli, J., Santini, C.C.: Complexity engineering, harnessing emergent phenomena as opportunities for engineering. In: Reports of the Santa Fe Institute's Complex Systems Summer School (2005)
7. Cederfeldt, M., Elgh, F.: Design automation in SMEs-current state, potential, need and requirements. In: ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy, p. 1507. Engineers Australia (2005)
8. Christiansson, P., Svidt, K., Sørensen, K.B.: Future integrated design environments. J. Inf. Technol. Constr. **14**, 445–460 (2009)
9. Collins, H.M.: Tacit knowledge, trust and the Q of sapphire. Soc. Stud. Sci. **31**, 71–85 (2001)
10. Densmore, D.: Bio-design automation: nobody said it would be easy. ACS Synth. Biol. **1**, 296 (2012)
11. Earman, J.: A Primer on Determinism. Springer, Heidelberg (1986)
12. Ermolayev, V., Keberle, N., Matzke, W.-E.: An upper level ontological model for engineering design performance domain. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 98–113. Springer, Heidelberg (2008)
13. Ermolayev, V., Matzke, W.-E.: Towards industrial strength business performance management. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS (LNAI), vol. 4659, pp. 387–400. Springer, Heidelberg (2007)
14. Gardner, T.S.: Synthetic biology: from hype to impact. Trends Biotechnol. **31**, 123–125 (2013)
15. Gentner, D., Stevens, A.L.: Mental Models. Psychology Press, Hillsdale (1983)
16. Gioia, D.A., Pitre, E.: Multiparadigm perspectives on theory building. Acad. Manag. Rev. **15**, 584–602 (1990)
17. Gregor, S., Jones, D.: The anatomy of a design theory. J. Assoc. Inf. Syst. **8**, 312–335 (2007)
18. Habibi, N., Mohd Hashim, S.Z., Rodriguez, C.A., et al.: A review of CADs, languages and data models for synthetic biology. J. Teknologi **63** (2013)
19. Hassoun, S.: Genetic/bio design automation for (re-)engineering biological systems. In: Design, Automation and Test in Europe Conference & Exhibition (DATE 2012), pp. 242–247 (2012)
20. Huerta, M., Downing, G., Haseltine, F., et al.: NIH Working Definition of Bioinformatics and Computational Biology. US National Institute of Health (2000)
21. Lindgren, P., Rasmusssen, O.H., Poulsen, H., et al.: Open business model innovation in healthcare sector. J. Multi Bus. Model Innov. Technol. **1**, 23–52 (2012)
22. Lipshtat, A., Loinger, A., Balaban, N.Q., et al.: Genetic toggle switch without cooperative binding. Phys. Rev. Lett. **96**, 188101 (2006)
23. Luscombe, N.M., Greenbaum, D., Gerstein, M.: What is bioinformatics? A proposed definition and overview of the field. Methods Inf. Med. **40**, 346–358 (2001)

24. Ma, W., Trusina, A., El-Samad, H., et al.: Defining network topologies that can achieve biochemical adaptation. Cell **138**, 760–773 (2009)
25. Martin, J., Odell, J.J.: Object-Oriented Methods: A Foundation, UML edn., 2nd edn. Prentice-Hall, Upper Saddle River (1998)
26. Matzke, W.-E.: Biotechnology, synthetic biology, and ICT define the emerging knowledge-based bio-economy. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI 2013. CCIS, vol. 412, pp. 1–19. Springer, Heidelberg (2013)
27. Matzke, W.-E.: Engineering design performance management–from alchemy to science through ISTa. In: Proceedings of the 4th International Conference on Information Systems Technology and Its Applications. LNI, GI, pp. 154–179 (2005)
28. Matzke, W.-E., Stumpf, M., Mascher, T.: Towards industrial strength BDA - a short essay. In: 5th International Workshop on Bio-Design Automation. Imperial College, London, UK (2013)
29. Nonaka, I.: The knowledge-creating company. Harv. Bus. Rev. **85**, 162–171 (2007)
30. Noori, N., Weiss, M.: Going open: does it mean giving away control? Technol. Innov. Manag. Rev. (2013)
31. Otero-Muras, I., Szederkényi, G., Hangos, K.M., et al.: Dynamic analysis and control of biochemical reaction networks. Math. Comput. Simul. **79**, 999–1009 (2008)
32. Parnas, D.L., Clements, P.C.: A rational design process: how and why to take it. IEEE Trans. Softw. Eng. **SE-12**, 251–257 (1986)
33. Polanyi, M.: The Tacit Dimension. Routledge & K. Paul, London (1967)
34. Riehle, D.: Erratum to: The single-vendor commercial open source business model. Inf. Syst. E-Bus. Manag. **10**, 427 (2012)
35. Riehle, D.: The single-vendor commercial open sourse business model. Inf. Syst. E-Bus. Manag. **10**, 5–17 (2012)
36. Rouse, M.: Industrial strength. In: WhatIs.com (2005)
37. Shaw, W.H.: Engineering management in our modern age. In: Proceedings of the Engineering Management Conference. IEMC ‘02, pp. 504–509. IEEE (2002)
38. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
39. The Eclipse Foundation. Eclipse Foundation (2014)
40. Thiriet, M.: Conclusion. In: Intracellular Signaling Mediators in the Circulatory and Ventilatory Systems, pp. 911–918. Springer, New York (2013)
41. van Beuzekom, B., Arundel, A.: OECD Biotechnology Statistics 2009. OECD Publishing, Paris (2009)
42. Wagner, R.K., Sternberg, R.J.: Practical intelligence in real-world pursuits: the role of tacit knowledge. J. Pers. Soc. Psychol. **49**, 436 (1985)
43. Zott, C., Amit, R.: Business model design: an activity system perspective. Long Range Plan. **43**, 216–226 (2010)

Availability Assessment of Information and Control Systems with Online Software Update and Verification

Vyacheslav Kharchenko[1], Yuriy Ponochovnyi[2], and Artem Boyarchuk[1(✉)]

[1] National Aerospace University, Kharkiv, Ukraine
V.Kharchenko@khai.edu, a.boyarchuk@mail.ru
[2] Poltava National Technical University, Poltava, Ukraine
pnchl@rambler.ru

Abstract. The paper is devoted to the analysis of the verification objectives and features for information and control systems (ICS) software at the different stages of life cycle, in particular, on development and operation stages for space ICS software. Theoretic-set description of objectives, verified functions considering their criticality for space systems software and scenarios of online verification are proposed. Availability Markov chains based models are developed for different scenarios of space systems software online verification. Multi-fragment Markov models of ICS software for a serviced spacecraft is researched taking into account updating and removal of detected faults.

Keywords: Availability information and control systems · Online software verification · Markov's models · Online software updates

1 Introduction

1.1 Motivation

Successful development of aerospace industry relies on dependability and availability of information and control systems (ICS) for spacecraft (SC) as well as on quality and reliability of its hardware and software. In the first decade of the 21st century, 27 % of spacecraft failures that proved fatal or restrict their use were associated with hardware faults (6 %) and software faults (21 %) [1]. Therefore, it requires of additional measures to improve the processes of software development and maintenance, that is important task in point of view functional safety. Other aspect in this regard is to improve the quality and flexibility of software verification procedures considering the specific issues of the development and application of SC ICS.

The definition of the process of verification has no single term. For example, in [2] verification refers to the process of determining whether or not the products of a given phase of a software development process fulfill the requirements established during the previous phase.

At the same time authors of the research use the definition of the verification process taken from the standards of aerospace engineering [3–5]. In particular,

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 300–324, 2014.
DOI: 10.1007/978-3-319-13206-8_15

according to the standard of European Cooperation for Space Standardization ECSS-E-ST-40C6. Space Engineering. Software [4] "verification is a process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product".

Following a standard approach, the development of critical software is a costly process, and most of the costs are incurred not by development but its qualification trials [1, 3]. The use of software tools with the possibility of their modification allows verifying the set of non-critical software functions after the launch of a spacecraft and fixing the detected faults during the next software update. However, given the high criticality of the software, this choice must be justified by the mathematical models which take into account changing of the failure rates.

Existing models of systems with variable parameters use full-scale experiment [6], simulation methods [7], Bayesian analysis [8] and the most preferred method – apparatus of Markov's [9, 10] and semi-Markov's processes [11, 12]. In [13], a system approach is developed to the construction of multi-fragment models, but it does not provide modeling of procedures related to software updates and online verification.

The aim of this paper is to provide the theoretic-set description of the objectives, stages, scenarios of verification and models in order to assess the availability of ICS considering software faults, detected by online corrective verification during maintenance. Other one is development and analysis of availability models of ICS of manned spacecraft considering software updates. To estimate of availability function the calculation and research on multi-fragment Markov's models are performed.

1.2 Analysis of the Related Work

An overview of methods and technologies for software verification and system modeling for some critical applications considering their features is given in [3, 11, 14]. For spacecraft ICS software, the following verification methods are applied [4, 5, 15]: the testing, the method static analysis, the method of documentation analysis, the method of problem review, the method of inspection, etc.

Description of these methods is verbal and any formal procedures are not proposed in point of view online verification. Formal verification methods [16, 17], based on reporting and verifying systems specifications using special notations are severely limited in application. It is explained by complexity of the tasks and impossibility of correction and verification of software during a flight. The features related to software verification of spacecraft ICS are the following [3, 5, 18].

For some systems the complete verification functions cannot be made in full at the ground due to their specific conditions related to (a) the inability to fully simulate flight conditions and provide verification tasks to be performed onboard during take-off, orbital flight and landing; (b) the uncertainty related to both the flight conditions and target problems, the set of which may vary in its process, and (c) high complexity of testing tasks (volumes of input data which must be verified).

The required trustworthiness of verification must be ensured; otherwise the decision should be made about (a) the correction of the project in terms of scope of tasks; (b) continue the process of verification and modification to achieve acceptable risk level; (c) the possibilities and procedures of additional verification in flight conditions to improve (or ensuring the required) trustworthiness.

For commercial space projects, it is possible that onboard verification may be more cost effective and does not require: (a) time-consuming verification; (b) the need to develop or purchase expensive hardware; (c) to attract highly skilled professionals in the unique areas.

For safety reasons on the results of verification and due to software faults the decision may be made about: (a) system self-destruction or (b) blocking of certain functions because of failures occurred due to physical hardware faults, software faults and interaction faults [11, 13, 18]; (c) the redistribution of resources of ICS and execution of re-verification.

Thus, there is a need to improve existing and to develop new methods of software verification of spacecraft ICS. We shall call such verification as online verification (OV) of software; we shall call it online corrective verification (OCV) if the results of verification lead to correction of software.

The paper is structured as follows: the second section is devoted to the classification and description of ICS functions by criticality level (Subsect. 2.1); description of stages and operational basis for the verification process (Subsect. 2.2); description of the operations and objectives of verification with the possibility of continuation in flight mode (Subsects. 2.3, 2.4). Based on these steps the set-theoretic description of the functions of different criticality is given in the Subsect. 2.5. Besides a variety of scenarios to eliminate faults according with results of the OCV is proposed too (Subsect. 2.6). The third section offers an approach to developing availability models corresponding to scenarios. The fourth chapter contains the detailed model presentation and research of duplicated ICS with online software update. The fifth chapter is devoted to the research extended model with software update and collection of faults. In the last chapter the conclusions are made, directions for future works are outlined.

2 ICS Model with Online Verification and Update

2.1 Set-Theoretic Model of Functions, Objectives and Stages of Verification

Set of functions (MF) of spacecraft ICS differ by the level of criticality in accordance with international and national regulatory framework [3, 5]. Criticality is determined as ultimate damage that may occur as a result of the relevant event (emergency). They were marked as A, B, C, U (according to the criticality level).

Thus spacecraft ICS performs the set of functions MF which are divided at the four subsets considering the criticality level:

$$\mathrm{MF} = \mathrm{FA} \cup \mathrm{FB} \cup \mathrm{FC} \cup \mathrm{FU}. \quad (1)$$

Each set is described by the set of functions:

$$FA = \{f_{Ai}, i = 1, \ldots, b_A\},\ FB = \{f_{Bj}, j = 1, \ldots, b_B\},$$
$$FC = \{f_{Ck}, k = 1, \ldots, b_C\},\ FU = \{f_{UI}, I = 1, \ldots, b_U\}, \tag{2}$$

where: $\forall N,\ M \in \{A, B, C, U\}, N \neq M : FN \cap FM = \emptyset$,

$$\text{Card } MF = b_A + b_B + b_C + b_U. \tag{3}$$

Each of the functions f_z of MF set is described by criticality level (tolerated cases) u_{cr}, target description $a_{cr,}$ means m_{cr} and complexity w_{cr} of realization $f_z \sim \{u_{crz}, a_{crz}, m_{crz}, w_{crz}\}$.

Criticality attributes for functions $f_z \in MF$ are presented in Table 1.

Table 1. Characteristics of ICS functions considering their criticality

Criticality	Function attributes F			
	Criticality level, u_{cr}	Objective, a_{cr}	Means of implementation, m_{cr}	Complexity of implementation, w_{cr}
FA	Key functions for ensuring safety	Eliminate emergency situations	Automatic/ manual in exceptional cases	Simplest implementation
FB	Additional functions for ensuring safety	Do not initialize FA elements	Mostly automatic	More complex techniques than for FA implementation
FC	Supplement functions	To react at the situations tolerated for functions FA, FB	Automatic/ manual	Techniques of different complexity
FU	Non-critical functions	Do not take part at the elimination and tolerating of situations A, B, C	Automatic/ manual	Techniques of different complexity

2.2 The Stages of Development and Implementation of Spacecraft ICS

Spacecraft ICS provides performance verification on a set of the following stages:

$$MSt\ = \{StD,\ StQ,\ StA,\ StFP,\ StFE,\ StAL\}, \tag{4}$$

where: StD – development stage with the set $V_{StD} = \{v_{StD\,e},\ e\ =\ 1, \ldots,\ c_{StD}\}$;

StQ – qualification stage with the set of operations $V_{StQ} = \{v_{StQi}, i = 1, \ldots, c_{StQ}\}$;
StA – acceptance stage with the set of operations $V_{StA} = \{v_{StAi}, i = 1, \ldots, c_{StA}\}$;
StFP – flight preparation stage, $V_{SFP} = \{v_{StFPi}, i = 1, \ldots, c_{StFP}\}$;

StFE – flight execution stage with the set $V_{StFE} = \{v_{StFEi}, i = 1, \ldots, c_{StFE}\}$;
StAL – after landing stage with the set $V_{StAL} = \{v_{StALi}, i = 1, \ldots, c_{StAL}\}$.

Given set of stages is considered as enlarged one. Each of these stages can be represented by sets of sub-stages. In particular, the phase of flight (StFE) includes the sub-stages: pre-orbital (StPO), orbital (StO), landing (StL): StFE = {StPO, StO, StL}.

Listed steps are repeated for subsequent flights when verification is performed in full or in part depending on the changes in the functions of the system. The expression to define MSt will look as follows:

$$\begin{aligned} \mathrm{MSt} &= \{\mathrm{MSt_j},\ j = 1, \ldots,\ r;\ \mathrm{StAE}\} \\ &= [\{\mathrm{St_1D},\ \mathrm{St_1Q},\ \mathrm{St_1A},\ \mathrm{St_1FP},\ \mathrm{St_1FE},\ \mathrm{St_1AL}\}, \\ &\quad \{\mathrm{St_2D},\ \mathrm{St_2Q},\ \mathrm{St_2A},\ \mathrm{St_2FP},\ \mathrm{St_2FE},\ \mathrm{St_2AL}\}, \ldots, \\ &\quad \{\mathrm{St_rD},\ \mathrm{St_rQ},\ \mathrm{St_rA},\ \mathrm{St_rFP},\ \mathrm{St_rFE},\ \mathrm{St_rAL}\},\ \mathrm{StAE}], \end{aligned} \tag{5}$$

where: r – number of flights; StAE – stage of the system application ending.

2.3 Subset of Verification Operations

Set of verification operations (MV) is described as the union of operations' sets by stages:

$$\mathrm{MV} = V_{StD} \cup V_{StQ} \cup V_{StA} \cup V_{StFP} \cup V_{StFE} \cup V_{StAL}. \tag{6}$$

Operations performed during verification at different stages during re-launches may be repeated, thus

$$\begin{aligned} \mathrm{Card\ MV} \le \mathrm{Card}\ V_{StD} + \mathrm{Card}\ V_{StQ} + \mathrm{Card}\ V_{StA} + \mathrm{Card}\ V_{StFP} \\ + \mathrm{Card}\ V_{StFE} + \mathrm{Card}\ V_{StAL}. \end{aligned} \tag{7}$$

If the system functions do not change from one launch to another, the number and volume (for complex functions) of verification operations may be reduced:

$$\begin{aligned} \forall N \in \{D, Q, A, FP, FE, AL\}, \forall i \in \{2, \ldots, r\}\colon V_{StiN} \subseteq V_{Sti-1N}, \\ \mathrm{Card}\ V_{StiN} \le \mathrm{Card}\ V_{Sti-1N}. \end{aligned} \tag{8}$$

In case of changing the set of performed functions MF or their means of implementation, the verification is carried out at appropriate stages for that part of the system functions and tools that have changed (MF*), and configuration tools which ensure their integration into the modified system (MF_{conf}). For such situations:

$$\forall f_s \in MF^*, \forall i \in \{2, \ldots, r\}: \ V_{StiN(s)} = V_{Sti-1N(s)} \cup V_{StiN(s)conf}, \tag{9}$$

where: $V_{StiN(s)conf}$ – set of additional verification operations.

2.4 Set of Verification Objectives

Given the multi-stage ICS life cycle, the overall objective of verification is divided into many sub-objectives. The set of verification objectives may be defined as:

$$MO = \{MOD,\ MOQ,\ MO\ A,\ MO\ FP,\ MO\ FE,\ MO\ AL\}, \quad (10)$$

where each of the sets may be decomposed for the sub-sets in line with two features:

- criticality level of verification functions, u_{cr} (A,B,C,U);
- task of verification solved at the current stage, a_{cr}, etc.

Regulatory framework, the feasibility and possible economic efficiency considerations for commercial launches admit that some of the functions of the system can be verified during the flight. Let's make decomposition of verification objectives of ICS for the flight stage:

$$MO\ FE = \{OFE\ A,\ OFE\ B1,\ OFE\ B2,\ OFE\ B3,\ OFE\ C2,\ OFE\ C3\}, \quad (11)$$

where: OA1 – confirmation of trustworthiness of verification of function A for the decreasing of level of acceptable risk,

OB1 – confirmation of trustworthiness of verification of function B for the decreasing of level of acceptable risk,
OB2 – verification of function B which is impossible to check during ground operations,
OB3 – verification of function B in case it is possible within safety requirements,
OC2 – verification of function B which is impossible to check during ground operations,
OC3 – verification of function C in case it is possible within safety requirements.

So:

$$MO = \{\Delta MO_T, \Delta MO_P, \Delta MO_G\}, \quad (12)$$

where: $\Delta MO_T = \{OA1,\ OB1\}$ – set of objectives aimed on confirmation of trustworthiness of verification of functions A, B for decreasing of the risk level;

$\Delta MO_P = \{OB2,\ OC2\}$ – set of objectives aimed on verification of functions B, C in case it is possible within safety requirements;
$\Delta MO_G = \{OB3,\ OC3\}$ – set of objectives aimed on verification of functions B, C which is impossible to check during ground operations.

Conformance between objectives of verifications during the flight and their criticality is described in Table 2.

Indexes of the functions correspond to the objectives and conditions of the verification:

- increasing or confirmation of verification trustworthiness – index «T», otherwise «T'»;

Table 2. Conformance between objectives of verifications during the flight and their criticality

Criticality	$\Delta MO_T/F$		$\Delta MO_P/F$		$\Delta MO_G/F$	
A	OA1	FA_T				
		$FA_{T'}$				
B	OB1	FB_T	OB2	FB_P	OB3	FB_G
		$FB_{T'}$		$FB_{P'}$		$FB_{G'}$
C			OC2	FC_P	OC3	FC_G
				$FC_{P'}$		$FC_{G'}$
U						

– performing verification of functions that cannot be checked in ground conditions or impossible to provide the required assessment trustworthiness – index «P», otherwise – index «P'»;
– performing verification of functions in flight, if it is acceptable in point of view safety requirements and requires costs less than the ground conditions – index «G», otherwise – index «G'».

2.5 Graphical Interpretation of Functions Considering Verification Objectives and Their Set-Theoretic Description

Given the decomposition of verification objectives of spacecraft ICS functions of during the flight their graphic interpretation for different levels of criticality is shown in Fig. 1. In this figure the different lines separate functions FA (a), FB (b), FC (c), FU (d) in accordance with possible verification objectives.

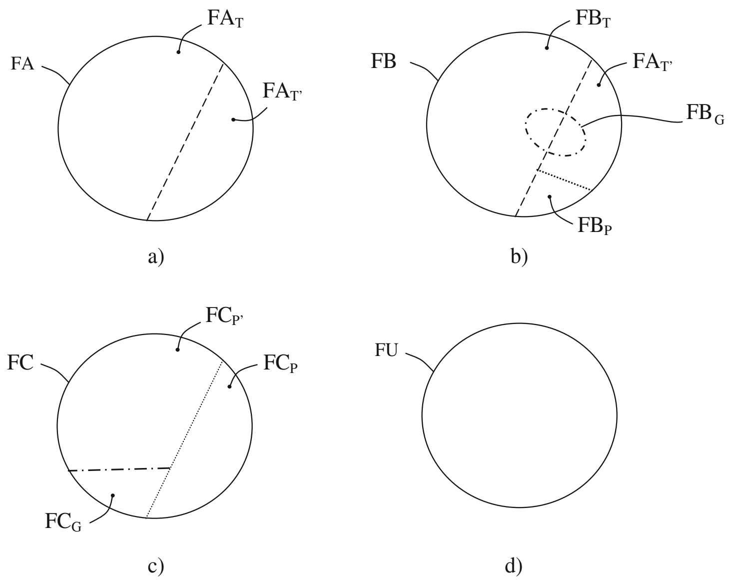


Fig. 1. Graphic interpretation of spacecraft ICS functions of different criticality considering the verification objectives (for the functions FA(a), FB(b), FC(c), FU(d))

The following formulas describing features of functions are valid:
(a) for function FA

$$FA = \{FA_T,\ FA_{T'}\},\ FA_T \cap FA_{T'} = \emptyset;. \tag{13}$$

(b) for function FB

$$\begin{gathered} FB = \{FB_T,\ FB_P,\ FB_G,\ FA_{T'},\ FB_P,\ FB_{G'}\}, \\ FB = FB_T \cup FB_{T'} = FB_P \cup FB_{P'} = FB_G \cup FB_{G'}, \\ FB_T \cap FB_{T'} = FB_P \cap FB_{P'} = FB_G \cap FB_{G'} = \emptyset,\ FB_P \subset FB_{T'} \\ FB_G \cap FB_P = \emptyset(FB_G \subset FB/FB_P),\ FB_G = FB_{TG} \cup FB_{T'G}, \\ FB_{TG} \cap FB_{T/G} = \emptyset, FB_{TG} = FB_T \cap FB_G, FB_{T'G} = FB_{T'} \cap FB_G; \end{gathered} \tag{14}$$

(c) for function FC

$$\begin{gathered} FC = \{FC_P,\ FC_{P'},\ FC_G,\ FC_{G'}\},\ FC = FC_P \cup FC_{P'} = FC_G \cup FC_{G'}, \\ FC_P \cap FC_{P'} = FC_G \cap FC_{G'} = \emptyset,\ FC_G \subset FC_{P'}, FC_G \cap FC_P = \emptyset, \\ FC_{P'} = FC_{P'G} \cup FC_{PG'}, FC_{P'G} \cap FC_{PG'} = \emptyset, \\ FC_{P'G} = FC_{P'} \cap FC_G, FC_{PG'} = FC_{P'} \cap FC_{G'}. \end{gathered} \tag{15}$$

2.6 Scenarios of Verification and Execution in Case of Fault Detection

Having considered the analysis and description of verification processes it is important to develop the potential scenarios of reaction for the detected faults and to define the basic set of availability models of spacecraft ICS.

The set of scenarios may be decomposed taking into account the following features described by the set H (H = {hi, i = 1,…,4}):

- possibility of OV for the functions which cannot be verified in full in ground conditions, h_1 (h_{11} – without correction during detection of faults, h_{12} – with correction faults and saving the full functionality – OCV case;
- possibility of OV for the functions with limited criticality which can be verified with significant expenses, h_2 (h_{21} – without correction, h_{22} – with correction, OCV case);
- possibility of software updates during the flight, h_3 (h_{31} – preventive maintenance, h_{32} – functional);
- possibility of faults tolerating by means of functions blocking and losing of quality, h_4 (manages degradation).

Features h_i (and sub-features h_{ij}) are Boolean variables and have the values of 0 or 1 depending of availability («1») or non-availability («0») to perform OV or OCV.

Considering the values of features $h_i \in H$ the set of scenarios $MSC = \{SC_q,\ q = = 1,\ldots,w\}$ is developed, which are described by the vector of values $SC_q \sim < h_{iq} >$. Here come some possible scenarios:

- SC1 – all possible (for ground conditions) types of verification are performed; OV and correction of faults are impossible during the flight; the set of features $H_{SC1}(\forall i == 1,\ldots,4 :\ h_i = 0)$ is assigned for this scenario;
- SC2 – all possible (for ground conditions) types of verification are performed; OV is impossible during the flight; some correction of faults is possible; correction is made only in case of detecting of fault; the set of features $H_{S2} = \{h_1 = h_2 = h_4 = 0,\ h_{31} = 1\}$ is assigned for this scenario;
- SC3 – all possible types of verification are performed; correction of all faults is performed (OCV case); the set of features $H_{SC3} = \{h_{12} = 1,\ h_2 = h_3 = h_4 = 0\}$ is assigned for this scenario;
- SC4 – all possible types of verification are performed (in ground conditions and during the flight); correction of detected faults is performed (OCV case); blocking of the part of functions and system degradation is possible; the set of features $H_{SC4} == \{h_{12} = h_4 = 1,\ h_2 = h_3 = 0\}$ is assigned for this scenario.

3 An Approach to Modeling

In order to assess the availability and other system indicators depending on scenarios set MSC there is a need to develop the corresponding models. They may be based on single- [10, 19] and multi-fragment Markov's models [11, 13]. Let's define some examples of these models.

Model 1 (scenario SC1). Graph of states for this model is shown at the Fig. 2(a). It is described by some set of states and transitions caused by failures and repairs of hardware and software considering the system architecture and possibilities of failure detecting and system repair.

Model 2 (scenario SC2, Fig. 2(b)). The difference is caused by possibility of detecting and partial fixing/tolerating of the software faults. Decreasing of their intensity is considered by α, (α < 1). Besides, new coefficient δ is added for the model (δ > 1), which considers the complexity of hardware for the online correction of software code during the flight.

Models 3 and 4 (scenarios SC3 and SC4, Fig. 2(c)). There is a tree-view graph of transitions for these models; the type of transition depends on the type of event occurred: detecting of fault or start of verification process. Nevertheless graph models for the scenarios SC3 and SC4 are equal, scenario SC4 models the degradation of system functions. Thus the intensity of transition into the state of verification (and detecting of fault) may be higher for the scenario SC4.

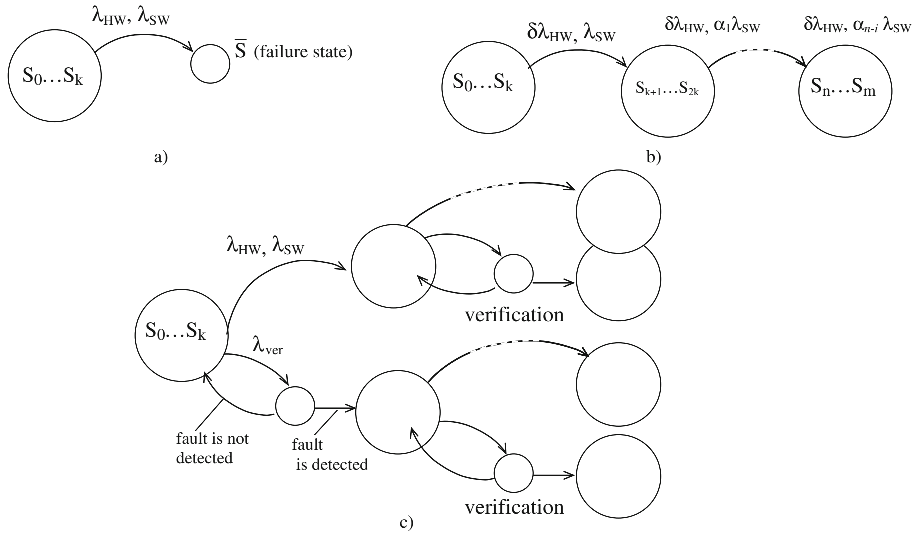


Fig. 2. Availability models of ICS with OV (a) – model 1; (b) – model 2; (c) – models 3, 4

4 Industrial Case: Availability Assessment of Onboard ICS

In this paper we examine manned spacecraft with the most common architecture of ICS, which includes two duplicated channels; each of them operates with the same software version (Fig. 3). It is foreseen to restore the functional state of ICS by spacecraft crew [5, 18]. Software of such spacecraft allows such modifications and software reengineering during spacecraft maintenance period.

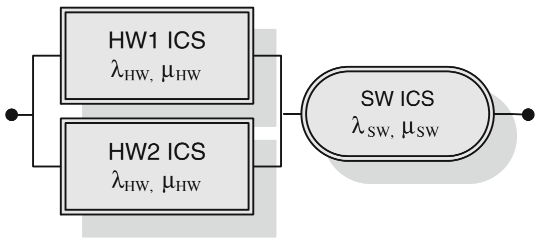


Fig. 3. Structured scheme of reliability of bi-channel one-version spacecraft ICS

The following assumptions are considered for the development of availability model of spacecraft ICS:

– ICS can be either operable or inoperable at any given time, and the flows of events that transform the system from one state to another function are simple;
– ICS recovery after failure caused by software fault is performed by restarting;

– during ICS operation time the updating of software is performing and software faults are fixed; state of system update is inoperative (state of system unavailability).

4.1 Availability Model of Spacecraft ICS with Software Updates

Markov chains-based analysis may be chosen for the research method and can be proved by the correctness of the assumptions. Variation of the failure rate λ_{SW} is processed by the mathematical apparatus of the regular multi-fragment Markov models (RMFM) [13]. In this case the Markov chain may be presented by the number of the separate fragments, which differ from each other by one parameter; this parameter is changing from one fragment to another one by fixed known value. Thus the basic RMFM model may be represented by the graph shown in Fig. 4.

The process of ICS functioning is arranged with the following way [21]. Initially the system implements all functions according with specification and stays at the S_0 state. Later, some hardware (physical) faults occur, and the system consequently transits to the S_1 state (failure of the one hardware channel, up-state), S_2 (failures of two hardware channels, down state) and then due to repair system is restored and transits to the S_0 and S_1 states). Then, in some period of time, the system failure occurs caused by software (design) fault, and the system transits to S_3 state. The system is restored by restart and transits to the S_0 state. In some time interval the software updates are executing and the system transits to S_4 state.

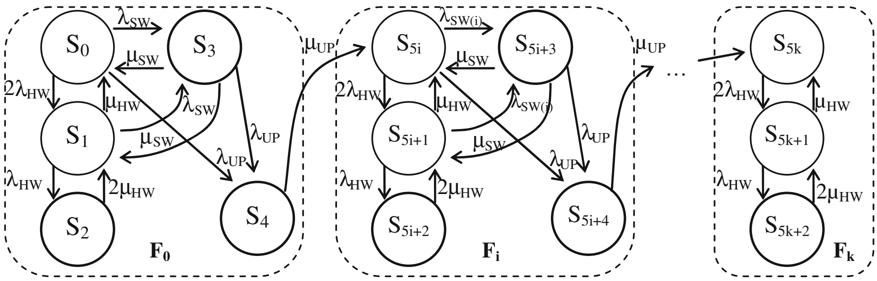


Fig. 4. Marked graph of ICS functioning with software updates

It is important to consider the S_4 state of updates. It ensures the adequacies of the model due to the fact that this state corresponds to the actual exploitation state of the system.

During software update procedures there procedure of fixing design faults is executing, and then system transits to the new fragment RMFM (state S_{5i}), which is characterized by new parameter $\lambda_{SW\,i}$. In the last fragment of the model all software faults are fixed, and there are only hardware faults are possible.

The system of differential Kolmogorov-Chapman's equations for the availability model from Fig. 4 will consist from the following regular blocks:

For the initial fragment F_0:

$$\begin{cases} \frac{dP_0(t)}{dt} = -(2\lambda_{HW} + \lambda_{SW} + \lambda_{UP})P_0(t) + \mu_{HW}P_1(t) + \mu_{SW}P_3(t), \\ \frac{dP_1(t)}{dt} = -(\lambda_{HW}+\lambda_{SW}+\mu_{HW})P_1(t) + 2\lambda_{HW}P_0(t) + 2\mu_{HW}P_2(t) + \mu_{SW}P_3(t), \\ \frac{dP_2(t)}{dt} = -2\mu_{HW}P_2(t) + \lambda_{HW}P_1(t), \\ \frac{dP_3(t)}{dt} = -(2\mu_{SW} + \lambda_{UP})P_3(t) + \lambda_{SW}P_0(t) + \lambda_{SW}P_1(t), \\ \frac{dP_4(t)}{dt} = -\mu_{UP}P_4(t) + \lambda_{UP}P_0(t) + \lambda_{UP}P_3(t). \end{cases} \tag{16}$$

For internal fragments Fi:

$$\begin{cases} \frac{dP_{5\cdot i}(t)}{dt} = -\left(2\lambda_{HW} + \lambda_{SW(i)} + \lambda_{UP}\right)P_{5\cdot i}(t) + \mu_{HW}P_{5\cdot i+1}(t) + \mu_{UP}P_{5\cdot i-1}(t) + \mu_{SW}P_{5\cdot i+3}(t), \\ \frac{dP_{5\cdot i+1}(t)}{dt} = -\left(\lambda_{HW} + \lambda_{SW(i)} + \mu_{HW}\right)P_{5\cdot i+1}(t) + 2\lambda_{HW}P_{5\cdot i}(t) + 2\mu_{HW}P_{5\cdot i+2}(t) \\ \qquad + \mu_{SW}P_{5\cdot i+3}(t), \\ \frac{dP_{5\cdot i+2}(t)}{dt} = -2\mu_{HW}P_{5\cdot i+2}(t) + \lambda_{HW}P_{5\cdot i+1}(t), \\ \frac{dP_{5\cdot i+3}(t)}{dt} = -(2\mu_{SW} + \lambda_{UP})P_{5\cdot i+3}(t) + \lambda_{SW(i)}P_{5\cdot i}(t) + \lambda_{SW(i)}P_{5\cdot i+1}(t), \\ \frac{dP_{5\cdot i+4}(t)}{dt} = -\mu_{UP}P_{5\cdot i+4}(t) + \lambda_{UP}P_{5\cdot i}(t) + \lambda_{UP}P_{5\cdot i+3}(t). \end{cases} \tag{17}$$

For the last fragment Fk:

$$\begin{cases} \frac{dP_{5\cdot k}(t)}{dt} = -2\lambda_{HW}P_{5\cdot k}(t) + \mu_{HW}P_{5\cdot k+1}(t) + \mu_{UP}P_{5\cdot k-1}(t) + \mu_{SW}P_{5\cdot k-2}(t), \\ \frac{dP_{5\cdot k+1}(t)}{dt} = -(\lambda_{HW} + \mu_{HW})P_{5\cdot k+1}(t) + 2\lambda_{HW}P_{5\cdot k}(t) + 2\mu_{HW}P_{5\cdot k+2}(t), \\ \frac{dP_{5\cdot k+2}(t)}{dt} = -2\mu_{HW}P_{5\cdot k+2}(t) + \lambda_{HW}P_{5\cdot k+1}(t). \end{cases} \tag{18}$$

Here: P_i - the probability of finding the system in state S_i; i – numbers of internal fragments; k – number of the last fragment. Availability function value is defined by the expression:

$$A(t) = \sum_{i=0}^{k} \left[P_{5\cdot i}(t) + P_{5\cdot i+1}(t)\right]. \tag{19}$$

4.2 Justification of Input Parameters of ICS Model with Software Updates

The objective of the research is the analysis of variation of availability function. Input parameters' values used for modeling were obtained from statistical data on exploitation of SC ICS [21]. Part of input parameters has the fixed values, the remaining parameters have changed within fixed intervals of values. Fixed values have the following parameters:

– failure rate of single hardware channel $\lambda_{HW} = 1.5e - 4$ (1/h);
– recovery rate of single hardware channel $\mu_{HW} = 1.5$ (1/h);
– initial software failure rate $\lambda_{SW0} = 4e - 3$ (1/h);
– system recovery rate after occurrence of software fault (software restart) $\mu_{SW} = 2$ (1/h);
– software failure rate after fixing of all faults is zero: $\lambda_{SWk} = 0$ (1/h).

The following values of input parameters were used:

Table 3. Variable values of the model input parameters

Model parameter	Variable values of parameters				
Δ_{SW} (1/h)	1e−3	5e−4		2e−4	1e−4
λ_{UP} (1/h)	4.63e−4		2.31e−4		1.14e−4
μ_{UP} (1/h)	0.33		0.5		1

The following "basic" values of input parameters were used:

– step of changing of software failure rate $\Delta\lambda_{SW} = 1e - 4$ (1/h);
– software update rate $\lambda_{UP} = 4.63e - 4$ (1/h);
– system recovery rate after software update procedures $\mu_{UP} = 0.5$ (1/h).

4.3 Research Results of ICS Model with Software Updates

While researching of the model the special interest was given to the initial stage of system functioning, that's why the time interval T = 15000 h was considered (about 20 month).

The results of calculations are presented as the graphical dependency of availability function on system functioning time in Figs. 5, 6, 7, 8.

Graph analysis (Fig. 5) shows that the value of parameter $\Delta\lambda_{SW}$ has the key impact on speed of the fixing of the faults (the greater the $\Delta\lambda_{SW}$, the sooner the availability function of multi-fragment transits to the stable mode). Besides this, the value of $\Delta\lambda_{SW}$ parameter has no impact on value of minimum of availability function on the initial stage of system functioning.

The value of λ_{UP} parameter substantially impacts on value of maximum of availability function (Fig. 6). It is cause by the fact that the greater the number of software updates the greater total time of the system state of software updating (this state is considered as non-working state).

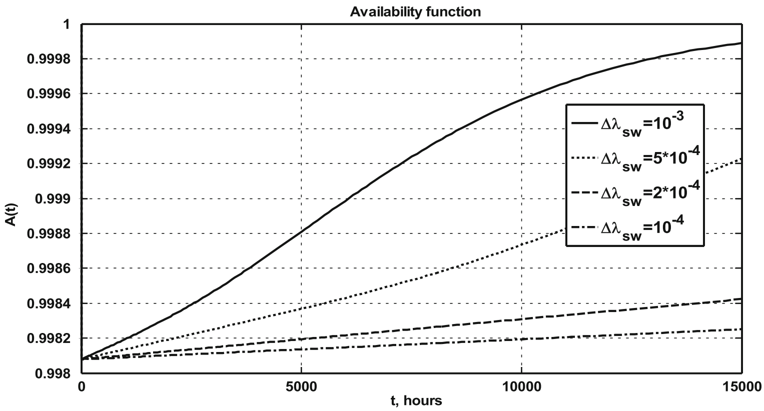


Fig. 5. Dependency of ICS availability function on time of system functioning with different values of $\Delta\lambda_{SW}$

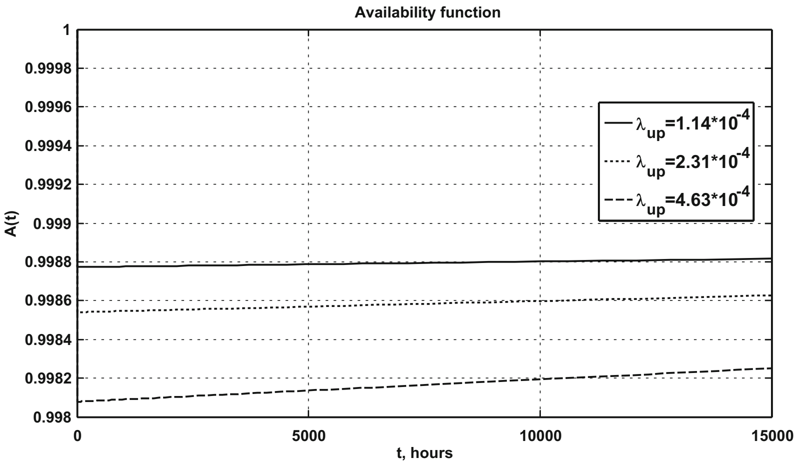


Fig. 6. Dependency of ICS availability function on time of system functioning with different values of λ_{UP}

Besides value of λ_{UP} parameter impacts on speed of transition of availability function into stable mode (curve for $\lambda_{UP} = 4.63e - 4$ has the bigger steepness then curves for $\lambda_{UP} = 1.14e - 4$ and $\lambda_{UP} = 2.31e - 4$ in Fig. 6). At the same time it is difficult to explain the speed of transition of availability function into the stable mode by the Fig. 6. It is important to carry on additional research on λ_{UP} parameter, greater than those listed at the Table 3.

This allows checking the complete fixing of software faults within the target time period of 15000 h.

Figure 7 clearly shows that value of μ_{UP} parameter impacts on minimum value of the availability function of multi-fragment model (having increased μ_{UP} value the

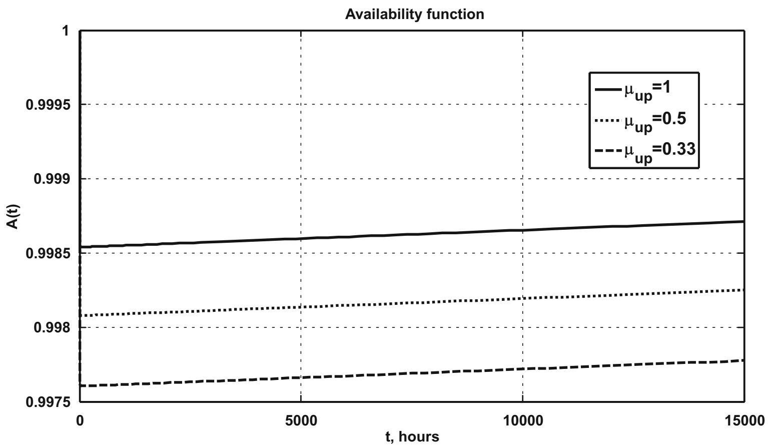


Fig. 7. Dependency of ICS availability function on duration of system exploitation with different values of μ_{UP}

duration of software updates procedure is decreasing). Impact of μ_{UP} parameter on speed of transition into stable mode cannot be caught from Fig. 7 (there are no transition into the stable mode at the graphs from Fig. 7), because there are no completed fixing of software faults during 15000 h with given parameters. So it is important to carry on further research with other values of μ_{UP} and λ_{UP} parameters.

Due to limited information obtained from Figs. 6 and 7 it is decided to carry on further research of the model with additional set of input parameters (Table 4).

The results of modeling with parameters from Table 4 are presented at the Fig. 8.

Obtained results of availability function are compared with stable value of availability function of duplicated single-version ICS with constant initial values of λ_{HW}, μ_{HW}, λ_{SW0}, μ_{SW} parameters using single-fragment modeling.

Table 4. Additional values of the model input parameters

Model parameter	Variable values of parameters			
λ_{UP} (1/h)	0.0014			0.014
μ_{UP} (1/h)	5	3	1	1

Figure 8 clearly shows the effectiveness of software updates procedures, because after 3000 working hours there is an increase of availability function of multi-fragment model in comparison with availability function of single-fragment model. Value of λ_{UP} parameter has an influence on maximum of availability function and its speed of transition to stable mode (curves ($\lambda_{UP} = 0.0014$, $\mu_{UP} = 1$) and ($\lambda_{UP} = 0.014$, $\mu_{UP} = 1$) in the Fig. 8). At the same time value of μ_{UP} parameter does not impact on speed of transition to stable mode (curves for $\mu_{UP} = 5$, $\mu_{UP} = 3$ and $\mu_{UP} = 1$ at the Fig. 8 transits to the stable mode within the same time interval).

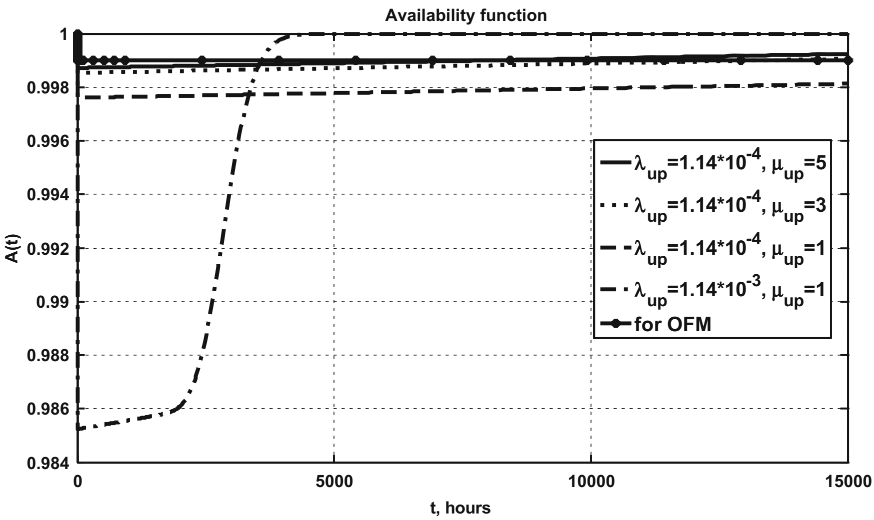


Fig. 8. Dependency of ICS availability function on time of system functioning for single-fragment system (OFM) and for additional variants of λ_{UP}, μ_{UP} values

The analysis of the modeling results of manned spacecraft ICS allows to formulate the following conclusions:

- to facilitate transition of availability function into the stable state it is necessary to increase the values of $\Delta\lambda_{\text{SW}}$ and λ_{UP} parameters, which means to increate the intensity of software updates and attempts to fix the maximum number of software faults with one update;
- at the initial period the system availability (with software updates procedures) is lower, then in the systems without software updates;
- availability rate may be increased at the initial period of functioning by increasing of μ_{UP} parameter (by speed up the system restoring).

5 Extended Availability Model of Spacecraft ICS with Software Updates and Collection of Faults

5.1 Development of the Extended Model

Unlike availability model studies in [13], the extended availability model provides that:

- after detecting of software fault in the software register and memory the accumulation of faults is in progress, which results in increasing of software failure rate λ_{SW} by a certain amount $\Delta\lambda_{\text{SW}}$;
- increasing of λ_{SW} is possible until some limit $\lambda_{\text{SW}\max}$.

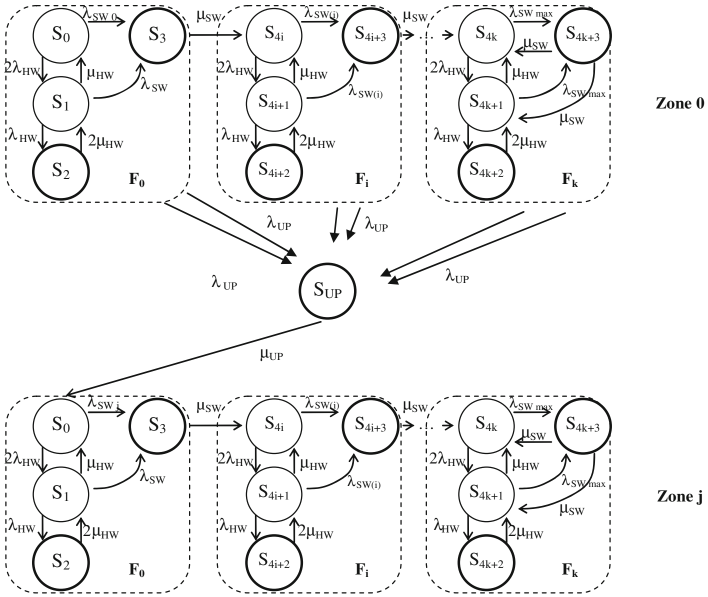


Fig. 9. Marked graph of states and transitions of spacecraft ICS taking into account increase of faults detecting and their eliminating by means of software updates

Having in mind the described assumptions the process of functioning of ICS may be presented as Markov's process [6] and modeled using multi-fragment Markov's model [9]; the marked graph of states is given in Fig. 9.

The graph may be conditionally separated at the initial zone Z0 and internal zones Zj. Then, each of these zones includes three types of fragments – initial type F0, internal type Fi and the latest type Fk.

The process of ICS functioning is defining with the following steps. At the initial stage the system implement all foreseen functions and stay at the S0(t) state of Z0 zone. During the functioning of the system the elements fail and recover. It leads to the transition to the states S1(t) – S3(t) of the initial fragment F0 of Z0 zone.

After the software fault detection and elimination the failure rate λ_{SW} decreases for the $\Delta\lambda_{SW}$. The system recovers and transits to the next fragment of zone Z0, which is characterized by new parameter $\lambda_{SW\,i}$.

After software updates completing the system transits from non-functioning state S_{UP} into the initial fragment of new zone Zj.

The value of availability function within one fragment zone may be calculated by the expression:

$$A(t)_{Z_j} = \sum_{i=0}^{k_j} \left[P_{4\cdot i}(t) + P_{4\cdot i+1}(t)\right]. \tag{20}$$

Thus, availability function in general may be calculated as sum of availability rates $A(t)_{Zj}$ within the limits of each zone with total number of zones as N_{UP}:

$$A(t) = \sum_{i=0}^{N_{up}} A(t)_{Z_i}. \tag{21}$$

The analysis of the model shows that the model has the regular structure. The regular structure means in repeating of fragment zones and fragments within corresponding zone, which allows tracking the changing of availability rate of ICS using the apparatus of Markov's chains.

5.2 Consideration of Input Parameters of ICS Models with Software Updates and Collecting of Faults

The input parameters which were considered for modeling are presented in Table 5. Special software blocks were designed in Matlab for the model research. Two options were considered for the development of models:

(1) during software updates the faults are fixed; transitions into new fragment zones are performed before complete eliminating of software faults: $\lambda_{SW\,k} = 0$;
(2) during software updates only collected faults are fixed ($\Delta\lambda_{SW\,UP} = 0$); additional parameter of limitation the size of model is introduced: nup – number of updates.

Table 5. Values of availability models input parameters

Symbol	Matlab-name	Value	Unit	Symbol	Matlab-name	Value	Unit
λ_{HW}	lambda_hw	1.5e−4	1/h	$\Delta\lambda_{SW}$	delta_lambda_sw	1e−3	1/h
λ_{SW0}	lambda_sw0	4e−3	1/h	$\Delta\lambda_{SW\,max}$	lambda_sw_max	1e−2	1/h
μ_{HW}	mu_hw	1.5	1/h	λ_{UP}	lambda_up	4.63e−4	1/h
μ_{SW}	mu_sw	2	1/h	μ_{UP}	mu_up	0.5	1/h
				$\Delta\lambda_{SW\,UP}$	delta_lambda_sw_up	5e−4	1/h

5.3 Availability Model of Spacecraft ICS with Elimination of Detected Doftware Faults

This model considers the first variant of ICS functioning in accordance with the graph at the Fig. 9. With this model it is foreseen to fix not only collected errors but to fix software faults as well. As a result, after the transition into new fragment zone Z_{j+1} the initial failure rate of software within zone $\lambda_{SW\,(j+1)}$ will be decreased by the value of $\Delta\lambda_{SW\,UP}$ depending to the software failure rate of the first fragment of the previous zone Z_j. The last fragment zone is characterized by complete fixing of software faults ($\lambda_{SW} = 0$). In order to estimate the number of updates (number of fragment zones) and the total number of fragments within each zone the following formulas may be applied:

$$N_{up} = \left]\frac{\lambda_{SW\,0}}{\Delta\lambda_{SW\,UP}}\right[, \quad (22)$$

$$N_{fr}(i) = \left]\frac{\lambda_{SW\,max} - \lambda_{SW\,0} + (i-1)\cdot\Delta\lambda_{SW\,UP}}{\Delta\lambda_{SW}}\right[. \quad (23)$$

Here "][" indicated integer division. Last zone includes only one fragment Nfr = 1 (by i = Nup + 1).

The following code is used to define the number of updates (zones of fragments) and the number of fragments within each zone during the fixing of software faults:

```
>> nup= round(lambda_sw0/delta_lambda_sw_up);
N_fr=[]; for i=1:nup
N_fr=[N_fr round((lambda_sw_max-(lambda_sw0-(i-1)*…
             delta_lambda_sw_up))/delta_lambda_sw)];
end; N_fr=[N_fr 1];
```

Having the agreed values of input data (Table 5) the estimated number of updates before complete eliminating of software faults is nup = 8. Respectively the model includes 9 zones of fragments with the number of fragments in each zone: N_fr = [6 7 7 8 8 9 9 10 1].

For development of oriented graph (Fig. 10) in Matlab the component grPlot.m was used [22]. Arrays V and E are used: V array contains the coordinates of graph states; E array contains transition rates between the states of the graph.

```
>> grPlot(V,E,'d','%d','',0.7) % digraph without weight;
```

Kolmogorov-Chapman's differential equations were solved using Matlab by method "ode15s" [23] for time interval of [0…50000] h. Function "matrixA" is used for filling in matrix of the coefficient of the system of differential equations; function "stiff" is used for formation of the array of derivatives.

```
function [A] = matrixA(Va, Ea)
index =[]; global A; A=[];
for i=1: length(Va)
for j=1: length(Va)
if i==j index = Ea(:,1)==i & (Ea(:,2)~=i); A(i,j)=0;
for k=1:length(Ea) A(i,j)= A(i,j)+-1*index(k)*Ea(k,3);
end
else
index = (Ea(:,1)==j) & (Ea(:,2)==i); A(i,j)=0;
for k=1: length(Ea) A(i,j)= A(i, j)+index(k)*Ea(k,3);
end end end end;
function dP=stiff(t,P)
global A; dP=A*P;
```

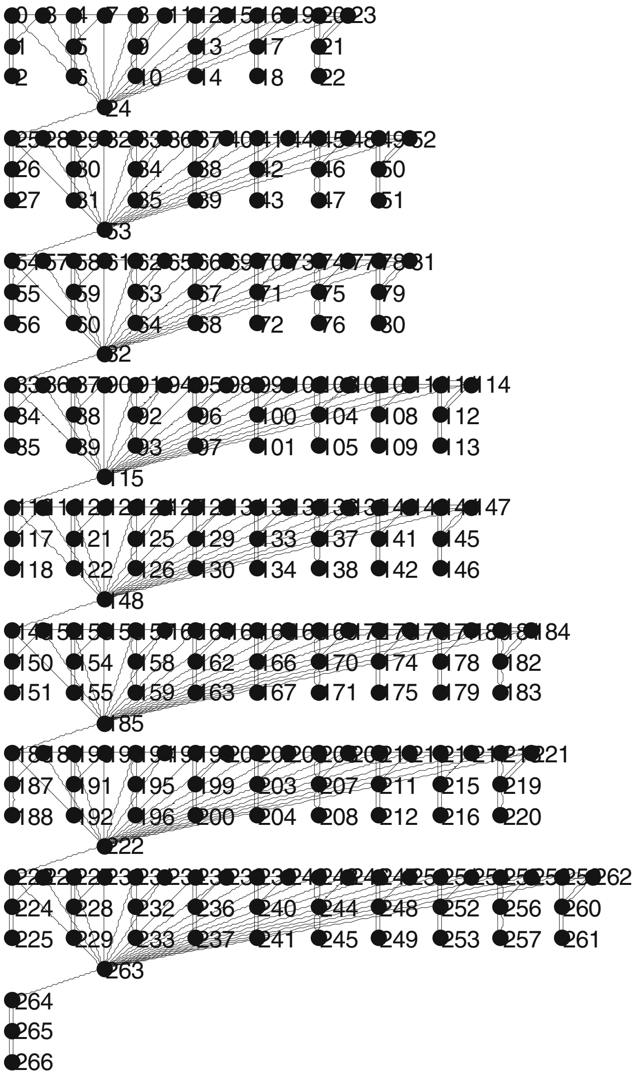


Fig. 10. Oriented graph of extended availability model of spacecraft ICS during execution of software updates

The value of availability function was defined by the program:

```
>> A=matrixA(V,E);
options = odeset('RelTol',1e-9, 'AbsTol',1e-9);
P0=[1 zeros(1, length(V)-1)];
[t,P] = ode15s(@stiff,[0 50000],P0,options);
Pg=[1:size(t,1)]'; Pg(:,1)=0; ng=0;
for j=0:nup
for i=1:N_fr(j+1)
Pg(:,1)= Pg(:,1)+P(:,ng+(4*i-3))+ P(:,ng+(4*i-2));
end;
ng=ng+4*N_fr(j+1)+1;
end;
```

The results of modeling are given in Fig. 11. The availability function of the system with updates is increased till the limit of the system with hardware faults. Wherein the effect of proceeding of updates will occur after 13500 h (first cross point), and gain in availability rate will occur after 20000 h (second cross point).

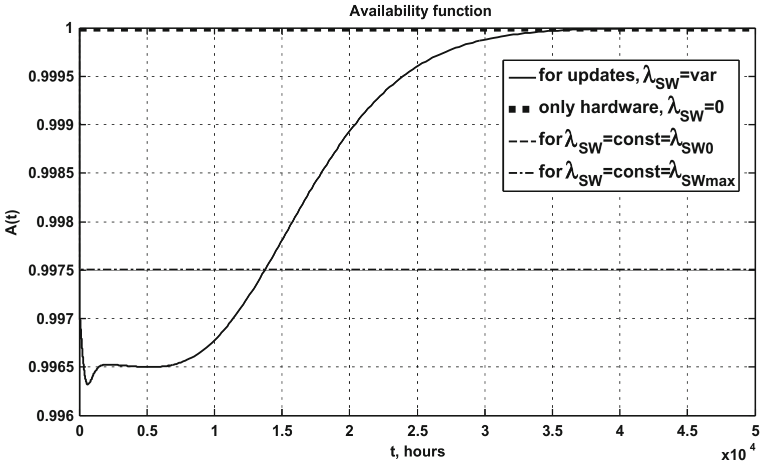


Fig. 11. Availability function of spacecraft ICS during performing of updates with fixing the software faults

5.4 Availability Model of Spacecraft ICS with Elimination of Collected Errors During Software Updates

The model considers the second variant of ICS functioning as shown at the Fig. 9. Only collected errors will be eliminated during software updates. As a result, during the transition into the new zone Z_{j+1} the initial rate of software failures of zone $\lambda_{SW\ (j+1)}$ will be equal to the software failures rate at the first fragment of the previous zone Z_j.

The model without fixing of software faults $\Delta\lambda_{SW\,UP} = 0$ the number of updates (zones of fragments) is the input parameter of model nup = 10.

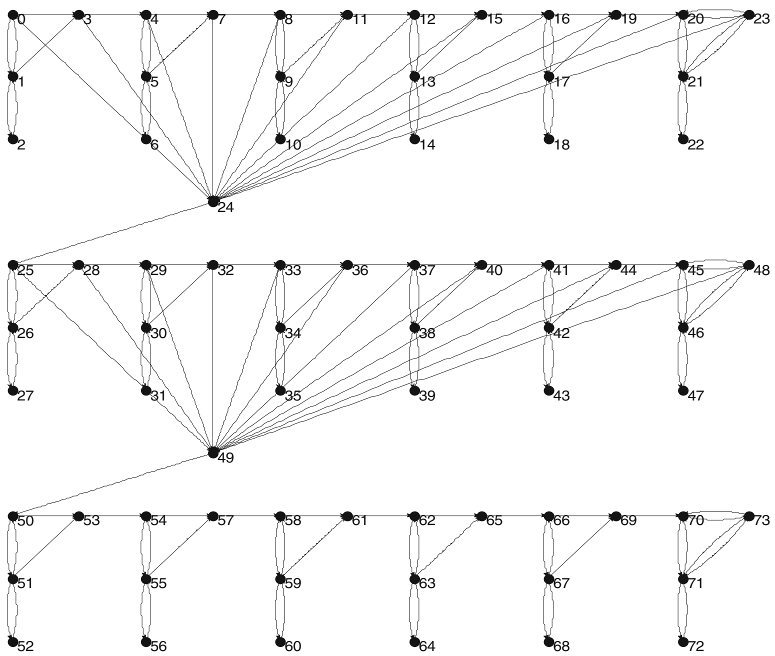


Fig. 12. Graph of the extended availability model of spacecraft ICS with two software updates

The number of the fragments in each zone is constant and is calculated as follows:

$$N_{fr}(i) = \left] \frac{\lambda_{SW\,max} - \lambda_{SW\,0}}{\Delta\lambda_{SW}} \right[. \tag{24}$$

The number of the fragments in each zone is constant and calculated using the following Mathlab code:

```
>> N_fr=[1:nup+1];
for i=0:nup
N_fr(i+1)=round((lambda_sw_max-
          lambda_sw0/delta_lambda_sw); end;
```

The oriented graph which describes the availability model with two updates is presented in Fig. 12.

Eliminating of collected faults only leads to speed up the system repairing after updating. Thus the parameters' values used for modeling are $\lambda_{UP} = 4.63e-2$ and $\mu_{UP} = 1$. The results of modeling are given in Fig. 13.

The analysis of the graph in Fig. 13 shows that the model with the highest availability rate is the model with the constant software failure rate, and gets the maximum value λ_{SWmax}.

The model with an increased probability rate λ_{SW} at the first stage shows a decreased rate, caused by the intervals of restart after the occurrence of the fault. At the

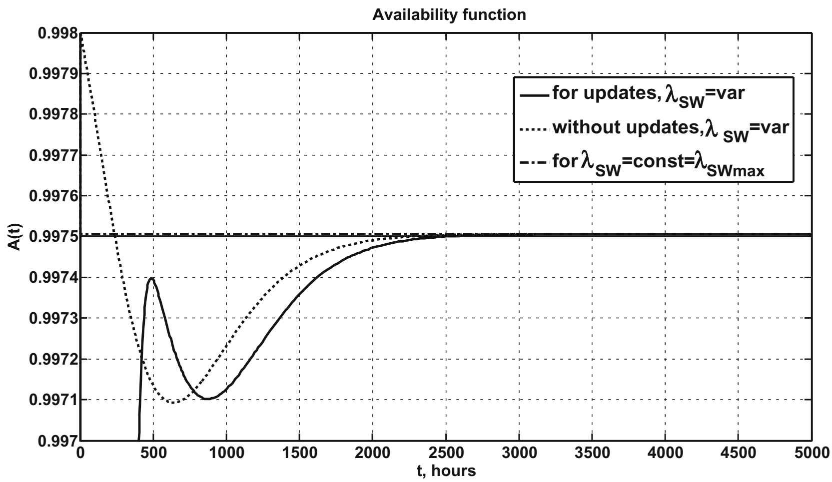


Fig. 13. Availability function of spacecraft ICS with eliminating of collected errors

same time such model is a more adequate one in the case of real conditions of ICS functioning. The model with updates has even more decrease of the trend at the beginning which caused by considering of the additional non-functional states of system updates.

Considering obtained results we can conclude that software updating without eliminating faults allows getting an advantage in 500 h. After that the effectiveness of system updating is lowering, availability rate leads to the limit specified by $\lambda_{\mathrm{SW\,max}} = 1\mathrm{e}-2$.

6 Conclusions

The paper analyzes the spacecraft ICS software as an object of verification, the regulatory framework in this area and the adequacy of existing verification methods for software specifics of space systems.

The concept of online corrective verification is introduced, which can be carried out under different scenarios for the functions of different levels of criticality. This concept, in our view, is important in terms of methodology and is considered as a component of the updating of partially serviced systems, which, in turn, relate to the evolved real-time systems [24].

The proposed set-theoretic description of the functions by the criticality levels, objectives and operations of OCV, sets of scenarios, indicators and models establish the basis for the development of procedures for software online verification and assessment of availability models of spacecraft ICS.

Further research would be aimed at the development of the procedures for the choosing of verification strategies, and at the architecting of the systems with OCV Further development of the models may be connected to the embedding of programs which calculate the software failure rates and recovery rates using different models and different initial data obtained from the testing and maintenance results of ICS.

Besides, it is interesting to research such systems considering other dependability attributes, first of all, integrity and confidentiality [20]. In this case the proposed models should be enhanced taking into account results of ICS software component vulnerability analysis and different types of attack rate.

References

1. Gorbenko, A., Kharchenko, V., Tarasyuk, O.: Risk analysis of rocket space system emergencies 2000 years. In: Kharchenko, V. (ed.) Safety Critical Infrastructures. Mathematical and Engineering Methods of Assessment and Ensuring, pp. 551–573. National Aerospace University "KhAI", Kharkiv (2011)
2. Collofello, J.: Introduction to Software Verification and Validation. SEI Curriculum Module SEI-CM-13-1.1. Software Engineering Institute (2004)
3. ECSS-Q-40B-2002. Space product assurance. Safety Requirements & Standards Division, Noordwijk, The Netherlands (2002)
4. ECSS-E-ST-40C6-2009. Space Engineering. Software Requirements & Standards Division, Noordwijk, The Netherlands (2009)
5. SOU-N NKAU 0078:2014. The Verification of Software-Hardware Complexes for Critical Application. Guidelines of the National Space Agency of Ukraine, Kiev (2014)
6. Min, X., Kim-Leng, P., Yuan-Shun, D.: Computing System Reliability: Models and Analysis. Kluwer Academic/Plenum Publishers, New York (2004)
7. Rotaru, T., Dalheimer, M., Pfreundt, F.-J.: Service-oriented middleware for financial monte carlo simulations on the cell broadband engine. Concurrency Comput. Pract. Exp. **22**(5), 643–657 (2010). John Wiley & Sons, New York
8. Gashi, I., Popov, P., Stankovic, V.: Uncertainty explicit assessment of off-the-shelf software: a bayesian approach. Elsevier J. Inf. Softw. Technol. **51**(2), 497–511 (2009)
9. Trivedi, K.S., Kim, D.S., Roy, A., Medhi, D.: Dependability and security models. In: Proceedings of 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), Washington, DC, pp. 11–20 (2009)
10. Trivedi, K.S., Vasireddy, R., Trindade, D., Nathan, S., Castro, R.: Modeling high availability systems. In: Proceedings of 12th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC '06), pp. 154–164. IEEE Press, New York (2006)
11. Kharchenko, V., Zamyrets, M., Zasuha, S., Ponochovniy, Y.: Elements of methodology correcting operational verification software information and control systems of spacecraft. Aerosp. Eng. Technol. **83**(6), 81–95 (2011)
12. Trivedi, K.S., Ciardo, G., Dasarathy, B., Grottke, M., Rindos, A., Varshaw, B.: Achieving and assuring high availability. In: Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008), Miami, Florida, USA, pp. 1–7 (2008)
13. Kharchenko, V., Odarushchenko, O., Odarushchenko, V., Popov, P.: Selecting mathematical software for dependability assessment of computer systems described by stiff markov chains. In: Ermolayev, V., Mayr, H.C., Nikitchenko, M., Spivakovsky, A., Zholtkevych, G. (eds.) ICTERI-2013. CCIS, vol. 1000, pp. 146–162. Springer, Heidelberg (2013)
14. Yoo, J., Jee, E., Cha, S.: Formal modeling and verification of safety-critical software. IEEE Softw. **26**(3), 42–49 (2009)

15. SOU-N NKAU 0058:2009. The Requirements to Functional Safety of Software-Hardware Complexes for Critical Application. Guidelines of the National Space Agency of Ukraine, Kiev (2009)
16. Abrial, J.-R.: The B-Book: Assigning Programs to Meanings. Cambridge University Press, New York (1996)
17. Ameur, Y.A., Boniol, F., Wiels, V.: Toward a wider use of formal methods for aerospace systems design and verification. Int. J. Softw. Tools Technol. Transf. **12**(1), 1–7 (2010)
18. Kulba, V., Mykryn, E., Pavlov, B.: Design of Information Control Systems for Orbital Stations. Nauka, Moscow (2002)
19. Guimaraes, A.P., Oliveira, H., Barros, R., Maciel, P.: Availability analysis of redundant computer networks: a strategy based on reliability importance. In: Proceedings of 3rd IEEE International Conference on Communication Software and Networks (ICCSN 2011), pp. 328–332. IEEE Press, Xi'an (2011)
20. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. **1**, 11–33 (2004)
21. Zasuha, S.: Investigation of influence of temporal parameters of software updates on the willingness of dual-channel information management system for spacecraft. Collected Works of Kharkiv Air Force University **29**(3), 131–135 (2011)
22. Grtheory - Graph Theory Toolbox by Sergii Iglin. http://www.mathworks.com/matlabcentral/fileexchange/4266-grtheory-graph-theory-toolbox/content/grPlot.m
23. Solve Stiff Differential Equations and DAEs; Variable Order Method - MATLAB ode15s. http://www.mathworks.com/help/matlab/ref/ode15s.html
24. Gorbenko, A., Kharchenko, V., Abdul-Hadi, A.M.: Evaluation of dependability concept: moving toward resilient and self-evolvable IT systems. In: Proceedings of 1st International Workshop on Critical Infrastructure Safety and Security (CrISS-DESSERT), Kirovograd, Ukraine, vol. 2, pp. 344–349 (2011)

Evolutionary Game of N Competing AIMD Connections

Oleksii Ignatenko(✉) and Oleksandr Synetskyi

Institute of Software Systems of NAS of Ukraine, Ukraine, Ukraine
oignat@isofts.kiev.ua

Abstract. This paper deals with modeling of network's dynamic using evolutionary games approach. Today there are many different protocols for data transmission through Internet, providing users with better or worse service. The process of choosing better protocol could be considered as a dynamic game with players (users), trying to maximize their payoffs (e.g. throughput). In this work we presented the model of network's dynamic using differential equations with discontinuous right side and proved existence and uniqueness of solution, formulated payoff matrix for a network game and found conditions of equilibrium existence depending of loss sensitivity parameter. The results are illustrated by simulations.

Keywords: Computer networks · Evolutionary games · Game theory · Congestion control

1 Introduction

This work deals with analytical model of competition between data flows in a network. Each flow belongs to a selfish non-cooperative user. Selfishness means that a user wants to utilize as much network resources as possible and the term non-cooperative does not imply that the users do not cooperate, but it rather means that any cooperation must be based on individual information with no communication or coordination among the players.

This competitive situation leads to obvious conflict when summary users' demand is bigger than network supply. If this happens network drops users' data, so generally it is not an event good for users. From the other side, network underloading (when demand is smaller than network capacity) is also undesirable because it leads to losses in efficiency of resource using.

From the beginning the Internet has been regulated by protocol (algorithms controlling user's behavior and implicitly forming network behavior as well) called TCP (the Transmission Control Protocol), introduced in the 1970s to provide reliable data transfer. However, that version was proved to be unreliable because it caused phenomenon known as *congestion collapse*. Later Van Jacobson improved TCP by developing a congestion control mechanism.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 325–342, 2014.
DOI: 10.1007/978-3-319-13206-8_16

The main idea is rather simple rule for a user's behavior, depending on information about network state:

if network is underloaded – increase your rate,
if network is overloaded – decrease your rate.

Delivering information about network state to the end user is a challenging problem and a crucial part of any feedback based protocol. As a rule a user has knowledge about successful delivery of his data (in other words he knows that network is probably underloaded) and about overload event (if he doesn't receive successful ACK – acknowledgement packet) with some delay. This type of information is called binary feedback. The natural rate control based on this information called AIMD [1] (additive increase, multiplicative decrease) scheme. There are another possibilities, but it was proved that AIMD algorithm will oscillate near the point of effective (all bottlenecks will be loaded) and fair (in some sense) allocation of network resources.

Nowadays, TCP isn't one protocol but a big family (number keeps increasing) of algorithms implementing different implementation of the origin idea. Protocol development went through the competitive evolution between different protocols, abandonment of some protocols and appearance of new ones. The possibility to deploy new versions of protocols gives users control to improve performance of his connection by choosing suitable algorithm. When many users are trying to achieve better performance it is difficult to predict consequences of such a competition. There is a problem how to ensure stable, fair and effective network behavior in the situation of dynamic and antagonistic interaction of selfish users. We address this problem with evolutionary game approach.

First we make short introduction of using game and control theory frameworks related to network problems. The work of F. Kelly et al. [2] was the first example of considering of resource allocation as an optimization problem. Later many authors see for example [3–5] have developed generalizations and variations of this framework. There are many approaches of investigation of complex networks from different directions (static, dynamic, deterministic etc.). The evolutionary games concept is a part of game theory that focuses on studying interactions between populations rather than individual players. One of the earliest publications about the use of evolutionary games in networking is [6] that study through simulations some aspects of competition between TCP users. The evolutionary games based on the concept of the ESS (Evolutionary Stable Strategy), defined in 1972 by the biologist Maynard Smith [7]. The ESS concept has been used in [8] in the context of ALOHA with power control. Fundamental survey of applications of game theory to networks is [9]. In this paper we develop the line of research presented in [6] by Altman et al. They considered a model of users which are using two different TCP connections – peaceful and aggressive. For this model it was shown that dynamic of this process described by difference equation has a stable solution and users payoffs are forming a structure of evolutionary game known as Hawk-Dove game. Also there were identified conditions under which equilibrium is evolutionary stable. However, there are limitations of the proposed approach.

First, the proposed method could not be generalized on the case of three or more protocols. Second, the network considered in this paper has very simple topology (a single node) and no extensions in this direction were proposed.

In current work we propose more general approach to evolutionary game of N different AIMD connections competing for resource. We found the solution using fixed point theorem, which makes possible generalization on N connection case and complex network topology. We formulated a game for player population, with strategy of choosing the best protocol. We found conditions for equilibrium existence and described it properties. Also we illustrate conditions by simulations of system dynamic.

2 Model

Consider a network of M processing nodes connected in some topology. Every node has at least one service link with limited overall capacity (or processing rate) p_i, $i = 1, \ldots, M$. Let I, K be a set of nodes indexes $\{1, \ldots, M\}$ and service links indexes $\{1, \ldots, L\}$ respectfully. There are N users, connected to this network. Let $x_j(t)$ be the transmission rate of j users, where $j \in J = \{1, \ldots, N\}$. There is natural assumption about vector of rates $\bar{x} = (x_1, \ldots, x_N)$: $x \in R_+^N$. If sum of transfer rates of data flows using the node's links is equal or bigger than the node capacity then overload event occurs (overload here is a synonym of packet loss). This scheme is an idealized model of widely deployed Droptail scheme. We will assume that routing is deterministic and uncontrolled and information about overload delivers to users momentarily. Let us fix the following notation (used for example in [10]) throughout this paper.

Denote $u_k(t)$, $k \in K$ as the service rate of k's link. The constituency matrix is the $M \times L$ matrix C whose c_{ij} element is equal to 1 if i's link belongs to j's node and otherwise is 0. Using this matrix we define a set U as $\left\{u \in R_+^K \mid Cu \leq 1\right\}$. The set U contains all possible service rates for the system. Let P be $diag\{p_1, \ldots, p_M\}$ - diagonal matrix.

The routing matrix R is the $M \times M$ matrix defined for $i, j \in P$. Element r_{ij} is equal to 1 if the output of i's link is the input of j's link and otherwise is 0. The input matrix A is the $L \times N$ matrix defined for $i \in K,\ j \in J$. Element a_{ij} is equal to 1 if j's user uses i's link and otherwise is 0.

2.1 Overload Conditions

When the system produces overload and how one can analytically predict it? This is an important problem of network modeling.

Proposition 2.1. (Stability condition) If user's vector of rates $x(t)$, $t \in [t_0, t_1]$ satisfy condition $\Xi\,\bar{x}(t) < 1$, where matrix $\Xi = CP^{-1} \sum_{k=0}^{M-1} \left(R^T\right)^k A$ then the system doesn't produce any overload events.

Proof. Let $\bar{x}(t)$ be the users' rates vector. In order to serve this data flow the system allocates the vector of service rates $\bar{u}(t)$ such that $Ax(t) - (I - R^T)P\bar{u}(t) = 0$. It is always possible if $\bar{u}(t) = P^{-1}(I - R^T)^{-1}A\bar{x} \in U$. The inverse matrix exists as a power series $[I - R]^{-1} = \sum_{k=0}^{M-1} R^k$. Stability condition is a formal expression of following inclusion: $P^{-1}(I - R^T)^{-1}A\bar{x} \in \mathrm{int}U$, where boundary of U was excluded to prevent overload event. Matrix Ξ describes controllability of the system, defined by matrixes C, P, R, A.

To clarify this notation and stability condition we will now consider some classical examples.

2.2 Examples

Single server. This is the simplest possible network. This model fits well for investigation interaction between users rather than network dynamic. Consider single node with capacity p, $M = L = 1$. There are N users with rates $\bar{x}(t) = (x_1(t), \ldots, x_N(t))^T$. Matrix $A = [1 \quad \ldots \quad 1]$, and set $U = [0, p] \subset R_+$. Here $\Xi = \left[\frac{1}{p}\right]$ and stability condition is $x_1(t) + \ldots + x_N(t) < p$.

Klimov model. Consider a model shown in Fig. 1. There are N users with vector of rates $\bar{x}(t) = (x_1(t), \ldots, x_N(t))^T$ and N links, where their data flows are processed. Each link has maximum capacity p_j, $j = 1, .., N$ and $u_j(t)$ is percentage of use of total amount of resource (e.g. CPU time, network bandwidth). Taking into account that $C = [1 \quad \ldots \quad 1]$, $P = \begin{bmatrix} p_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & p_N \end{bmatrix}$ we have

$$U = \left\{u \in R_+^K \mid CP^{-1}u \leq 1\right\} = \left\{u \in R_+^K \mid \frac{u_1}{p_1} + \ldots + \frac{u_N}{p_N} \leq 1\right\}.$$

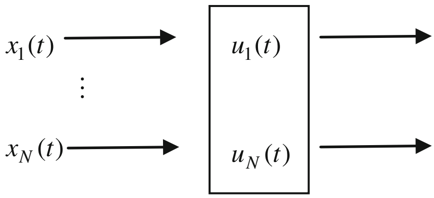


Fig. 1. Klimov model

Matrix $A = I$, so $\Xi = \left[\frac{1}{p_1} \quad \ldots \quad \frac{1}{p_N}\right]$ and stability condition is given by the following formula $\frac{x_1(t)}{p_1} + \ldots + \frac{x_N(t)}{p_N} < 1$.

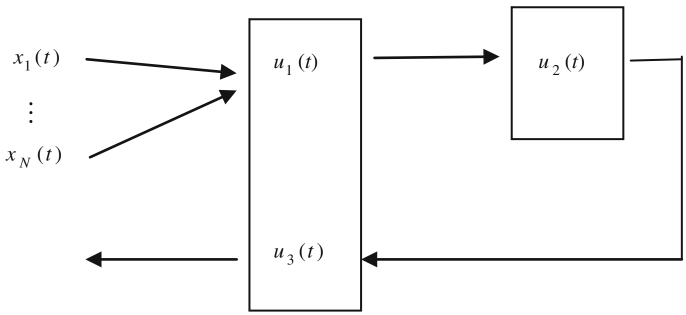


Fig. 2. Re-entrant line

Simple re-entrant line. Consider a more complex system shown in Fig. 2.

There are N users with vector of rates $\bar{x}(t) = (x_1(t), \ldots, x_N(t))^T$ sending data to first link of first node with p_1 maximum capacity and $u_1(t) \in [0,1]$ control parameter. After that packets go to second node with the single link with p_2 maximum capacity and $u_2(t) \in [0,1]$ control parameter. Finally, packets return to second link of first node (p_3 maximum capacity and $u_3(t) \in [0,1]$ control parameter) and left the system. Links 1 and 2 are situated on the same node, so their summary capacity is limited.

Let us define all matrixes for this network:

$$P = \begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & \ldots & 1 \\ 0 & \ldots & 0 \\ 0 & \ldots & 0 \end{bmatrix},$$

$$U = \left\{ u \in R_+^3 \mid \frac{u_1}{p_1} + \frac{u_3}{p_3} \leq 1,\ u_2 \leq p_2 \right\}$$

Routing matrix $R^T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, and $(I - R^T)^{-1} = I + R^T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$.

Finally, $\Xi = \begin{bmatrix} \frac{1}{p_1} + \frac{1}{p_3} & \ldots & \frac{1}{p_1} + \frac{1}{p_3} \\ \frac{1}{p_2} & \ldots & \frac{1}{p_2} \end{bmatrix}$ and stability condition consists of two inequalities:

$$\left(\frac{1}{p_1} + \frac{1}{p_3}\right)(x_1(t) + \ldots x_N(t)) < 1$$
$$(x_1(t) + \ldots x_N(t)) < p_2$$

If rates x satisfy stability condition then network will be lossless. But from practical point of view, there are many problems with applicability of this condition. First, in real network each user doesn't have information about system's current state and about rates of other users so he cannot calculate a proper rate. Second, the user cannot choose any rate he wants (at least in TCP scheme). Instead he chooses protocol, controlling his rate. The dynamic of protocol is investigated in next section.

2.3 System Dynamics

There are different instances of TCP class. The mostly used one is New-Reno. The behavior of New Reno is close to pure AIMD scheme. It adapts to the available capacity by increasing the window size in a linear way by α packets every round trip time and when it detects congestion it decreases the window size to β times its value. The constants α and β are 1 and 1/2, respectively, in New-Reno.

The original TCP specification doesn't forbid using any user defined congestion control mechanism. Even for AIMD like control one have freedom in changing values α and β. Obviously, if you set α and β bigger than standard (1, 1/2), you receive advantage against flow with lower values. This will cause unfair allocation of network resources and thus is undesirable.

This interaction of first protocol on second one is called unfriendly and it is said that the first protocol is more "aggressive" and latter is more "peaceful". Aggression here is an opportunity to grab more network resources than would be fair. The question of protocols interaction is quite complex. Building analytic model for predicting network behavior for different protocols is a challenging problem.

In last years, more aggressive TCP versions have appeared, such as HSTCP (High Speed TCP) and Scalable TCP. HSTCP can be modeled by an AIMD behavior where α and β are not constant anymore: α and β have minimum values of 1 and of 1/2, resp. and both increase as the window size increases. Scalable TCP is an MIMD (Multiplicative Increase Multiplicative Decrease) protocol, where the window size increases exponentially instead of linearly and is thus more aggressive. Versions of TCP which are less aggressive than the New-Reno also exist, such as Vegas.

In this section we build a dynamical model of AIMD connection using differential equation with discontinuous right-hand side.

Let $\bar{x}_0$ be an initial vector of rates and $\bar{\alpha}$, $\bar{\beta}$ vectors of parameters. According to original AIMD scheme [1] user rates are increasing between overloads with rate $\bar{\alpha}$. When overload occurs the rate drops to $\bar{\beta}x$. Now we will put into formal definitions. Define t_i, $i \geq 1$ as the first moment of time $t_i > t_{i-1}$, such that there exists $j \in J: \ [\Xi x(t_i)]_j = 1$. We will assume that the RTT (round trip times) are the same for all connections and losses are synchronized: when the combined rates attain capacity, all connections suffer from a loss.

Consider the following equation

$$\dot{x}(t) = \bar{\alpha} - \sum_{i=1}^{N_t} (I - B)x(t_i)\delta(t - t_i)\,, \tag{1}$$

where δ is delta-function, $B = diag\{\beta_1, \ldots, \beta_N\}$, $N_t = \max\{n : t_n \leq t\}$. Equation (1) is well-defined Caratheodory equation with discontinuous right-hand side, differential equations with impulses have been examined in many papers, which cannot all be referenced here (see [11] for references). It is known [11] that there is an almost continuous solution (continuous in all points except a set of measure zero)

$$x(t) = \alpha t - \sum_{i=1}^{N_t} (I - B)x(t_i)\eta(t - t_i), \tag{2}$$

where η is the Heaviside step function. Explicit formula (2) is not very practical but gives us important information about solution existence and its continuity in almost all points.

Denote V as a set $\left\{v \in R_+^N \mid \min[\Xi\, v]_j = 1, j \in J\right\}$ and W as a set $\left\{w \in R_+^N \mid \Xi\right.$ $\left. w \leq 1\right\}$. It is clear that V is compact set, W is convex compact set and $V \subseteq \partial W$.

Condition 2.1. For any $x \in V$ it is true that $Bx \in \mathrm{int}W$.

Let us explain Condition 2.1 informally. W is the vector set of possible user rates. W is convex compact set and $x(t) \in W$ for $t \geq t_0$. As mentioned $x(t)$ is an almost continuous function, and drops happened when $x(t) \in V$. After drop event users rates equal to $Bx(t)$. The Condition 2.1 means that after applying decreasing operator B user rate still will be in the admissible set W.

We use Brouwer's theorem to prove existence of limit solution of (1). Let us remind one version of it [12]. First we define homeomorphism and simplexes.

A set X is homeomorphic to the set Y if there is a bijective continuous function $h : X \to Y$ such that h^{-1} is also continuous.

A set $\{x^0, \ldots, x^n\} \subset R^m$ is affinely independent if $\sum_{i=0}^{n} \lambda_i x^i = 0$ and $\sum_{i=0}^{n} \lambda_i = 0$ imply that $\lambda_0 = \ldots = \lambda_n = 0$. An n-simplex is the set of all positive convex combinations of $n + 1$ element affinely independent set. Let Δ_n denote the closure of the standard n-simplex $\left\{y \in R^{n+1} \mid y_i > 0, i = 0, \ldots, n; \sum_{i=0}^{n} y_i = 1\right\}$.

Theorem. (Brouwer) [12] Let $X \subset R^n$ be homeomorphic to simplex Δ_{n-1} and let $f : X \to X$ be continuous, then f has a fixed point.

2.4 Solution of Dynamic System

Now we can formulate the main result of this section – existence and uniqueness of the limit solution.

Proposition 2.2. Let us consider admissible pair $\bar{\alpha}$, $\bar{\beta}$. If Condition 2.1 holds then for any $\bar{x}_0 \in W$ solution of (1) exists and is converging to unique periodical solution $\hat{x}(t)$.

Proof. Consider a map $f : V \to V$ defined as $f(v) = \{y \in V \mid \exists t > 0 : Bv + \alpha t = y\}$. Condition 2.1 holds, this means that $f(\cdot)$ is the well-defined function. It is clear that $f(\cdot)$ is continuous. Consider simplex Δ_{N-1} defined as $co\{e_1, \ldots, e_N\}$, where e_i are vectors from standard basis. For any $x \in V$ there exists unique $\psi \in \Delta_{N-1}$ such that $x = a\psi$, for some $a \in R$. This means that V is homeomorphic to Δ_{N-1}. Then Brouwer's theorem is applicable and there is a fixed point. Let $\bar{x}^* \in V$ be the fixed point of the map $f(\cdot)$. Denote $\bar{\gamma}$ as a vector with components $\gamma_i = \frac{\alpha_i}{1-\beta_i}$. Then the following is true: $B\bar{x}^* + \bar{\alpha}T = \bar{x}^*$, where $T = \min\{t : \bar{x}^* + \bar{\alpha}t \in V\}$. Using this property, we can calculate $\bar{x}^*$ directly

$$\bar{x}^* = (I - B)^{-1}\bar{\alpha}T = \gamma T.$$

Condition $\bar{x}^* + \bar{\alpha}T \in V$ could be re-written as $\min_i(\Xi\,\bar{x}^*)_i = 1$ or $\min_i(\Xi\,\gamma T)_i = 1$. So we can conclude that T is unique, so fixed point also is unique. Now let us solve (1) with the initial point $B\bar{x}^*$. It is clear that $\hat{x}(t) = \alpha t + B\bar{x}^*$ for $t \in [0, T)$ and $\hat{x}(T) = \hat{x}(0)$. So, solution $\hat{x}(t)$ is periodical with period T. Consider arbitrary solution with the initial point $x(0) \in W$. Let $x(t_1)$ be the first moment when $x(t_1) \in V$, then define $z_n = f(z_{n-1})$, $z_0 = f(x(t_1))$. All elements of $\{z_n\}$, $n = 0, \ldots, \infty$ belong to compact set V so there is a limit $\tilde{x} \in V$, $\tilde{x} = f(\tilde{x})$. The only possible solution, associated with this limit point is $\hat{x}(t)$. So we proved convergence of any solution to this periodical solution.

In next section we solve dynamic system (1) for different examples using Proposition 2.2. This notation is as follows: T - oscillation period, x^* fixed point from set V.

2.5 Applications

Single server. Note that $\Xi = \begin{bmatrix} \frac{1}{p} & \cdots & \frac{1}{p} \end{bmatrix}$, so condition $x \in V$ could be re-written as $\frac{x_1}{p} + \ldots + \frac{x_N}{p} = 1$. Let $x^* \in V$ be fixed point, then $x^* = \begin{bmatrix} \frac{\alpha_1}{1-\beta_1} & \cdots & \frac{\alpha_N}{1-\beta_N} \end{bmatrix}^T T$, and $\Xi\,A(Bx^* + \alpha T) = 1$. Solving this system of equations we obtain:

$$T = \frac{p}{\frac{\alpha_1}{1-\beta_1} + \ldots + \frac{\alpha_N}{1-\beta_N}}$$

Note, if all $\alpha_i = \alpha$, $\beta_i = \beta$, then $T = \frac{p(1-\beta)}{N\alpha}$, $x_i^* = \frac{p}{N}$.

This result was found by solving difference equation directly for two protocols system considered in [Altman].

Klimov model. Consider a model shown in Fig. 1. Matrix $\Xi = \begin{bmatrix} \frac{1}{p_1} & \cdots & \frac{1}{p_N} \end{bmatrix}$, so condition $x \in V$ could be re-written as $\frac{x_1}{p_1} + \ldots + \frac{x_N}{p_N} = 1$. Fixed point is the solution of following system of equations:

$$\Xi(Bx+\alpha T)=\frac{\beta_1 x_1}{p_1}+\frac{\alpha_1}{p_1}T+\ldots\frac{\beta_N x_N}{p_N}+\frac{\alpha_N}{p_N}T=1,$$
$$x=\begin{bmatrix}\frac{\alpha_1}{1-\beta_1} & \ldots & \frac{\alpha_N}{1-\beta_N}\end{bmatrix}^T T.$$

Finally,

$$T=\frac{1}{\frac{\alpha_1}{(1-\beta_1)p_1}+\ldots+\frac{\alpha_N}{(1-\beta_N)p_N}},\ x_i^*=\frac{\alpha_i}{(1-\beta_i)\left(\frac{\alpha_1}{(1-\beta_1)p_1}+\ldots+\frac{\alpha_N}{(1-\beta_N)p_N}\right)}.$$

Re-entrant line. $\Xi=\begin{bmatrix}\frac{1}{p_1}+\frac{1}{p_3} & \ldots & \frac{1}{p_1}+\frac{1}{p_3}\\ \frac{1}{p_2} & \ldots & \frac{1}{p_2}\end{bmatrix}$. Fixed point is the solution of following system of equations:

$$\Xi(Bx+\alpha T)=\begin{bmatrix}\left(\frac{1}{p_1}+\frac{1}{p_3}\right)(\beta_1 x_1+\alpha_1 T+\ldots+\beta_N x_N+\alpha_N T)\\ \frac{1}{p_2}(\beta_1 x_1+\alpha_1 T+\ldots+\beta_N x_N+\alpha_N T)\end{bmatrix}=1,$$
$$x=\begin{bmatrix}\frac{\alpha_1}{1-\beta_1} & \ldots & \frac{\alpha_N}{1-\beta_N}\end{bmatrix}^T T$$

The result is

$$T=\frac{1}{\sum_i \frac{\alpha_i}{(1-\beta_i)}\max\left\{\frac{1}{p_2},\frac{1}{p_1}+\frac{1}{p_3}\right\}},\ x_i^*=\frac{\alpha_i}{(1-\beta_i)}T.$$

3 AIMD Game Formulation

Now we consider a competition between users which use AIMD version of TCP with different parameters. Their connections are sharing a common network. We will assume that users send their packets exactly the same way, so we can reduce network topology to the single link type with capacity c.

In order to formulate a game in strategic form we must specify the players, their strategies, and their potential payoffs. Player here is a user. We assume that there are N AIMD strategies s_i with control parameters (α_i,β_i), $i=1,\ldots,N$. Denote S as a set of all possible strategies.

We consider payoff of the form $J_i(s) = Thp_i(s) - \lambda R(s)$, where $\bar{s} = (s_1, \ldots, s_N)$ - vector of strategies; $Thp_i(s) = 0.5(1+\beta_i)x_i^*$ - average throughput of i's player; λ - tradeoff parameter (sensitivity to losses); $R(s) = \frac{1}{T(s)}$ - loss rate.

Example. Let us calculate the payoffs for two strategies:

$$\begin{aligned}
J_1(s_i, s_i) =& J_2(s_i, s_i) = \frac{(1+\beta_i)}{4}c - \lambda \frac{2\alpha_i}{c\bar{\beta}_i},\\
J_1(s_1, s_2) =& \frac{(1+\beta_1)\alpha_1 c\bar{\beta}_2}{2(\alpha_1\bar{\beta}_2 + \alpha_2\bar{\beta}_1)} - \frac{\lambda}{c}\left(\frac{\alpha_1}{\bar{\beta}_1} + \frac{\alpha_2}{\bar{\beta}_2}\right),\\
J_1(s_2, s_1) =& \frac{(1+\beta_2)\alpha_2 c\bar{\beta}_1}{2(\alpha_1\bar{\beta}_2 + \alpha_2\bar{\beta}_1)} - \frac{\lambda}{c}\left(\frac{\alpha_1}{\bar{\beta}_1} + \frac{\alpha_2}{\bar{\beta}_2}\right),\\
& J_2(s_1, s_2) = J_1(s_2, s_1),\\
& J_2(s_2, s_1) = J_1(s_1, s_2)
\end{aligned}$$

3.1 Equilibrium in *N* Protocols Game

Consider a game with N AIMD strategies. We assume that all s_i are ordered lexicographically, $s_1 \geq s_2 \geq \ldots \geq s_N$, where $s_i \geq s_j$ means that $\alpha_i \geq \alpha_j$ and $\beta_i \geq \beta_j$. In other words protocols are sorted by aggressiveness ordering.

Proposition 3.1. If λ is sufficiently small than the most aggressive protocol is dominant strategy.

Proof. Suppose $\alpha_1 \geq \alpha_i$, $\beta_1 \geq \beta_i$ for all $i = 2, \ldots, N$. Consider payoffs for the first player $J_1(s_1, s_{-1})$ and $J_1(s_j, s_{-1})$. Let us find the period for both strategy profiles:

$T(s_1, s_{-1}) = \frac{c}{\frac{\alpha_1}{(1-\beta_1)} + A}$, where $A = \sum_k \frac{\alpha_k}{1-\beta_k}$ is sum, defined by strategy set s_{-1}, $T(s_j, s_{-1}) = \frac{c}{\frac{\alpha_j}{(1-\beta_j)} + A}$. Note, that $T(s_1, s_{-1}) < T(s_j, s_{-1})$.

Calculate throughputs:

$$Thp_1(s_1, s_{-1}) = \frac{(1+\beta_1)x_1^*}{2} = \frac{(1+\beta_1)\alpha_1 c}{2(1-\beta_1)\left(\frac{\alpha_1}{(1-\beta_1)} + A\right)} = \frac{(1+\beta_1)c}{2\left(1 + \frac{(1-\beta_1)}{\alpha_1}A\right)},$$

$$Thp_1(s_j, s_{-1}) = \frac{(1+\beta_j)x_j^*}{2} = \frac{(1+\beta_j)c}{2\left(1 + \frac{(1-\beta_j)}{\alpha_j}A\right)}.$$

Calculate payoffs:

$$J_1(s_1, s_{-1}) = Thp_1(s_1, s_{-1}) - \frac{\lambda}{c}\left(\frac{\alpha_1}{(1-\beta_1)} + A\right),$$
$$J_1(s_j, s_{-1}) = Thp_1(s_j, s_{-1}) - \frac{\lambda}{c}\left(\frac{\alpha_j}{(1-\beta_j)} + A\right).$$

$$Thp_1(s_1, s_{-1}) - \frac{\lambda}{c}\left(\frac{\alpha_1}{(1-\beta_1)} + A\right) > Thp_1(s_j, s_{-1}) - \frac{\lambda}{c}\left(\frac{\alpha_j}{(1-\beta_j)} + A\right),$$
$$\frac{(1+\beta_1)c}{2\left(1+\frac{(1-\beta_1)}{\alpha_1}A\right)} - \frac{(1+\beta_j)c}{2\left(1+\frac{(1-\beta_j)}{\alpha_j}A\right)} > \frac{\lambda}{c}\frac{\alpha_1}{(1-\beta_1)},$$
$$\lambda < \frac{c^2(1-\beta_1)}{\alpha_1}\left[\frac{(1+\beta_1)c}{2\left(1+\frac{(1-\beta_1)}{\alpha_1}A\right)} - \frac{(1+\beta_j)c}{2\left(1+\frac{(1-\beta_j)}{\alpha_j}A\right)}\right]$$

And since expression in right side is positive we obtain the result.

3.2 Nash Mixed and Pure in Two Protocols Game

Here we investigate the game for two protocols and find conditions for Nash equilibrium. From definition it is clear that $J_i(s_k, s_k) = J_j(s_k, s_k)$ - we will write just

$$J(s_k, s_k),\ J_i(s_k, s_p) = J_j(s_p, s_k),\ j \in \{1, 2\} \backslash i.$$

The matrix of this game is shown in Table 1.

Table 1. The payoff matrix

Player 2

Player 1

Strategy	s_1	s_2
s_1	$J(s_1, s_1),\ J(s_1, s_1)$	$J(s_1, s_2),\ J(s_2, s_1)$
s_2	$J(s_2, s_1),\ J(s_1, s_2)$	$J(s_2, s_2),\ J(s_2, s_2)$

Using standard techniques for calculating Nash we obtain:

$$J_1(s_1) = pJ(s_1, s_1) + (1-p)J(s_1, s_2)$$
$$J_1(s_2) = pJ(s_2, s_1) + (1-p)J(s_2, s_2)$$

assuming the probability of player 2 using the first strategy is p. In Nash equilibrium the payoff can't be further increased, so these two values should be indistinguishable, which leads to the following equation

$$pJ(s_1,s_1) + (1-p)J(s_1,s_2) = pJ(s_2,s_1) + (1-p)J(s_2,s_2)$$

Or, after solving it for p:

$$p = \frac{J(s_1,s_2) - J(s_2,s_2)}{(J(s_1,s_2) - J(s_2,s_2)) + (J(s_2,s_1) - J(s_1,s_1))}$$

Taking into account that p is a probability, we impose a natural restrictions on it: $0 \le p \le 1$, where cases with $p = 1$ or $p = 0$ result in game having a pure-strategy equilibrium (with dominant strategy s_1 and s_2, respectively), and $0 < p < 1$ corresponds to the case of mixed-strategy Nash equilibrium.

Should we investigate the conditions for the former, we get

$$\begin{aligned} J(s_1,s_2) - J(s_2,s_2) = & \frac{-\lambda(\alpha_2(1-\beta_1) + \alpha_1(1-\beta_2))}{c(1-\beta_2)(1-\beta_1)} + \frac{c\alpha_1(1+\beta_1)(1-\beta_2)}{2(\alpha_2(1-\beta_1)+\alpha_1(1-\beta_2))} \\ & + \frac{2\lambda\alpha_2}{c(1-\beta_2)} + \frac{1}{4}C(1+\beta_2). \end{aligned}$$

$$\begin{aligned} J(s_2,s_1) - J(s_1,s_1) = & \frac{-\lambda(\alpha_2(1-\beta_1) + \alpha_1(1-\beta_2))}{c(1-\beta_2)(1-\beta_1)} + \frac{c\alpha_2(1+\beta_2)(1-\beta_1)}{2(\alpha_2(1-\beta_1)+\alpha_1(1-\beta_2))} \\ & + \frac{2\lambda\alpha_1}{c(1-\beta_1)} + \frac{1}{4}C(1+\beta_1) \end{aligned}$$

Consequently,

$$p = 1 - \frac{2\alpha_2(1-\beta_1)}{\alpha_2(1-\beta_1) - \alpha_1(1-\beta_2)} - \frac{4\lambda(\alpha_1+\alpha_2) + c^2(\beta_1^2 - 1)}{c^2(1-\beta_1)(\beta_1-\beta_2)} + \frac{4\lambda\alpha_2}{c^2(1-\beta_1)(1-\beta_2)}.$$

Considering the case where game has pure-strategy equilibrium, we get two possible conditions: $p = 1$ or $p = 0$.

Solving the equations, we find the values of λ that correspond to the case of dominant strategy:

$$\lambda = \frac{c\beta_1\beta_2(\alpha_2\beta_1(1-\beta_1+2\beta_2) - \alpha_1(1-\beta_2)(1+\beta_1))}{4(\alpha_2(1-\beta_1) - \alpha_1(1-\beta_2))(\alpha_2(1-\beta_1)+\alpha_1(1-\beta_2))}$$

$$\lambda = \frac{c\beta_1\beta_2(\alpha_2\beta_1(1+\beta_2) + \alpha_1(1-\beta_2)(1+2\beta_1-\beta_2))}{4(\alpha_2(1-\beta_1) - \alpha_1(1-\beta_2))(\alpha_2(1-\beta_1)+\alpha_1(1-\beta_2))} \tag{3}$$

Now, for the game to have mixed-strategy equilibrium the following system of inequalities must hold:

$p < 1$ and $p > 0$.

After solving this system for λ we get

$$\frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+\beta_1)+\alpha_2\bar{\beta}_1(\beta_1-2\beta_2-1))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)} < \lambda < \frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+2\beta_1-\beta_2)-\alpha_2\bar{\beta}_1(1+\beta_2))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)} \quad (4)$$

We have proved:

Proposition 3.2. If λ satisfies (4) then there is Nash equilibrium in mixed strategies. If λ satisfies (3) then there is Nash equilibrium in pure strategies.

3.3 Extension for Protocols Parameters

The game settings in previous sections were limited by aggressive ordering of protocols. In this section we weaken this condition to cover protocol parameters relation that falls beyond the "aggressive-peaceful" scheme, namely situation when $\alpha_1 \le \alpha_2$ and $\beta_1 \ge \beta_2$.

Applying the same considerations as above, we get the same results for pure-strategy Nash equilibria, but for mixed-strategy equilibrium an additional constraint emerges.

Since we're looking for cases with $0 < p < 1$, we get the following conditions for $p > 0$:

$$\begin{cases} J(s_1,s_2) - J(s_2,s_2) > 0 \\ (J(s_1,s_2) - J(s_2,s_2)) + (J(s_2,s_1) - J(s_1,s_1)) > 0 \end{cases}$$

Similarly, $p < 1$ holds when

$$(J(s_1,s_2) - J(s_2,s_2)) + (J(s_2,s_1) - J(s_1,s_1)) > J(s_1,s_2) - J(s_2,s_2)$$

(It can be shown that other case with $J(s_1,s_2) - J(s_2,s_2) < 0$ results $\lambda < 0$, which has no physical sense, recalling that λ is an error weight).

So, in the end we have the following system of inequalities:

$$\begin{cases} J(s_1,s_2) - J(s_2,s_2) > 0 \\ J(s_2,s_1) - J(s_1,s_1) > 0 \end{cases}$$

Or, after replacement of J and transformations

$$\begin{cases} \lambda < \frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+2\beta_1-\beta_2)-\alpha_2\bar{\beta}_1(1+\beta_2))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)} \\ \lambda > \frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+\beta_1)+\alpha_2\bar{\beta}_1(\beta_1-2\beta_2-1))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)} \end{cases}$$

Replacing

$$\mu_1 = \frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+2\beta_1-\beta_2)-\alpha_2\bar{\beta}_1(1+\beta_2))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)}$$
$$\mu_2 = \frac{C^2\bar{\beta}_1\bar{\beta}_2(\alpha_1\bar{\beta}_2(1+\beta_1)+\alpha_2\bar{\beta}_1(\beta_1-2\beta_2-1))}{4(\alpha_1^2\bar{\beta}_2^2-\alpha_2^2\bar{\beta}_1^2)}$$

We get two possible solutions to the system above:

$$\begin{cases} \alpha_2(1-\beta_1)-\alpha_1(1-\beta_2)>0 \\ \mu_2<\lambda<\mu_1 \end{cases}$$

Or

$$\begin{cases} \alpha_2(1-\beta_1)-\alpha_1(1-\beta_2)<0 \\ \mu_1<\lambda<\mu_2 \end{cases}$$

Since $\alpha_1 \leq \alpha_2$ and $\beta_1 \geq \beta_2$ then $\mu_2 > \mu_1$, the actual solution is

$$\begin{cases} \alpha_2(1-\beta_1)-\alpha_1(1-\beta_2)<0 \\ \mu_1<\lambda<\mu_2 \end{cases}$$

Proposition 3.3. If $\alpha_1 \leq \alpha_2$ and $\beta_1 \geq \beta_2$ and$\alpha_1 < \alpha_2 < \frac{\alpha_1(1-\beta_2)}{1-\beta_1}$, $\mu_1 < \lambda < \mu_2$, then there is evolutionary stable equilibrium in mixed strategies.

Formulated conditions are consistent with the previous result with regards to protocol parameters specifics.

In next section we illustrate theoretical results with numeric simulations.

4 Simulation

We study in this section numerically dynamic system (1) and equilibriums of defined game with replicator dynamics. The practical value of these results could be divided on two parts. Firstly, this is analytical tool for predicting shares of network resources for given set of AIMD protocols. Existence and uniqueness of this point of resource allocation proved in Proposition 2.2.

Secondly, we can model users behavior (taking into account usual game theory assumption about rationality, common knowledge etc.) using replicator dynamic equation. This equation is rather quality solution tool that show a dynamic and shares of network resources for each users group.

4.1 Solution for Dynamic System

Numerical simulations were made using Wolfram Mathematica environment. On the picture below we show convergence of AIMD scheme for 2 and 3 dimensions (Fig. 3).

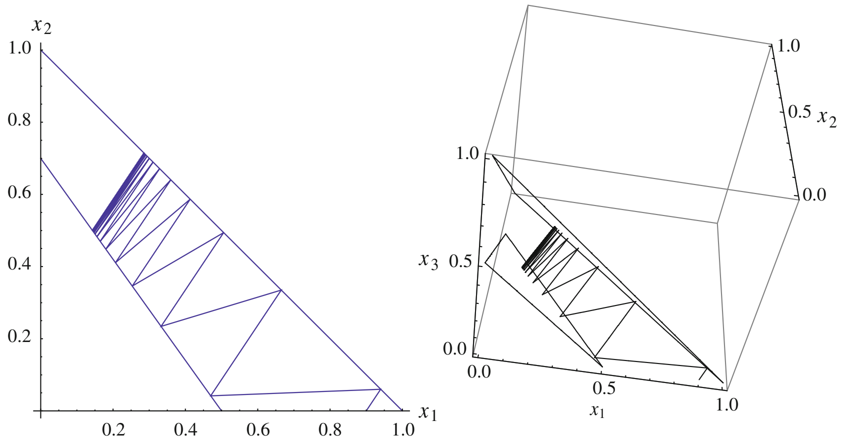


Fig. 3. Simulations results for 2-d and 3-d systems

4.2 Replicator Dynamics

We introduce here the replicator dynamics which describes the evolution in the population of the various strategies. In the replicator dynamics, the share of a strategy in the population grows at a rate equal to the difference between the payoff of that strategy and the average payoff of the population. More precisely, consider N strategies. Let x be the N-dimensional vector whose i element x_i is the population share of strategy i (i.e. the fraction of the population that uses strategy i). Thus, we have $\sum_i x_i(t) = 1$, $x_i(t) \geq 0$. Then the replicator dynamics is defined as

$$\dot{x}_i(t) = x_i(t)K\left(\sum_{j\neq i} J(i,j)x_j - \sum_j x_j(t)\sum_{k\neq j} J(j,k)x_k\right)$$

We investigate a case with $N = 3$ distinct strategies and pairwise payoff comparison:

$$J(s_1, X(t-\tau)) = x_1(t-\tau)J(s_1,\, s_1) + x_2(t-\tau)J(s_1,\, s_2) + x_3(t-\tau)J(s_1,\, s_3)$$
$$\dot{x}_i(t) = x_i(t)K(J(i,\, X(t-\tau) - (x_1(t-\tau)J(s_1, X(t-\tau)) + x_2(t-\tau)J(s_1, X(t-\tau)) + x_2(t-\tau)J(s_1, X(t-\tau)))$$

We provide simulation results for the 3 sets of parameters (Figs. 4, 5, and 6):

α_1	α_2	α_3	β_1	β_2	β_3	λ	K	c	τ
1.5	1.25	1	0.75	0.5	0.25	168	0.2	50	0.25

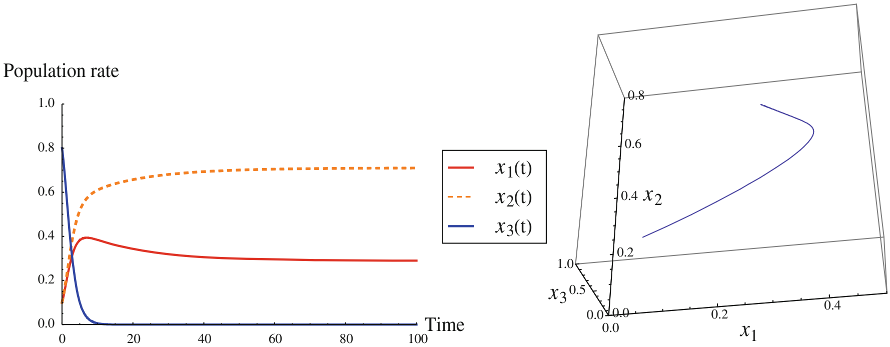


Fig. 4. $x_1(t)$, $x_2(t)$, $x_3(t)$ - shares of population

α_1	α_2	α_3	β_1	β_2	β_3	λ	K	c	τ
1.5	1.25	1	0.75	0.5	0.25	140	0.2	50	0.25

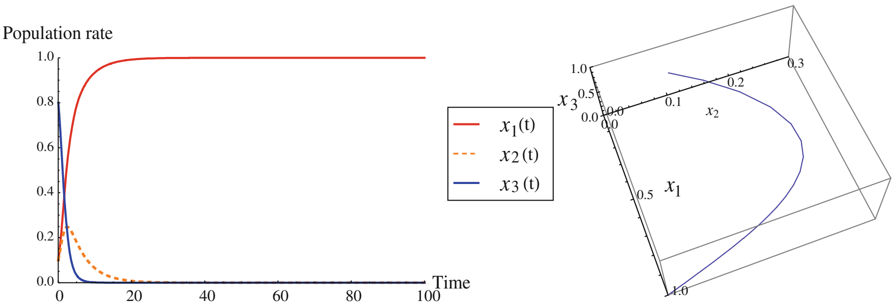


Fig. 5. $x_1(t)$, $x_2(t)$, $x_3(t)$

Expand the replicator dynamics differential equation to include a case with $N = 3$ different strategies.

$$J(s_1, X(t-\tau)) = \sum_p \sum_q x_p(t-\tau) x_q(t-\tau) J(A, p, q)$$

We provide a simulation results for 2 sets of parameters (Figs. 7 and 8):

α_1	α_2	α_3	β_1	β_2	β_3	λ	K	c	τ
1.5	1.25	1	0.75	0.5	0.25	168	0.2	50	15

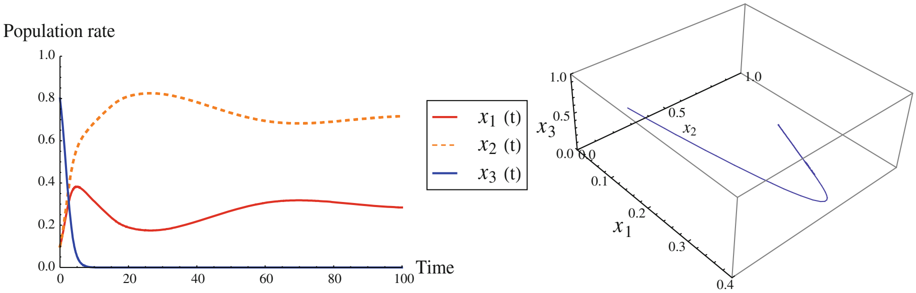


Fig. 6. $x_1(t)$, $x_2(t)$, $x_3(t)$

α_1	α_2	α_3	β_1	β_2	β_3	λ	K	c	τ
1.3	1.5	1.1	0.25	0.4	0.85	140	0.25	50	5

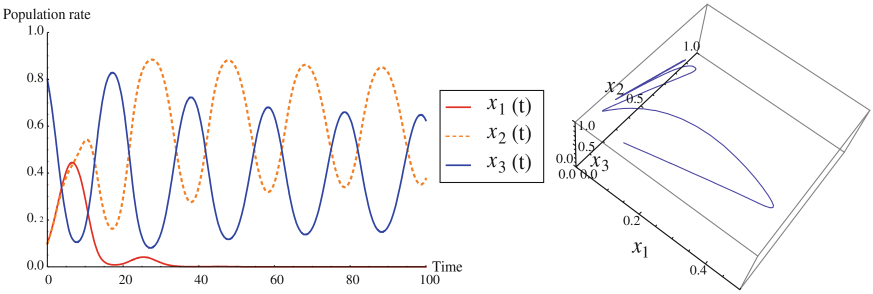


Fig. 7. $x_1(t)$, $x_2(t)$, $x_3(t)$

α_1	α_2	α_3	β_1	β_2	β_3	λ	K	c	τ
1.5	1.25	1	0.75	0.5	0.25	140	0.25	50	5

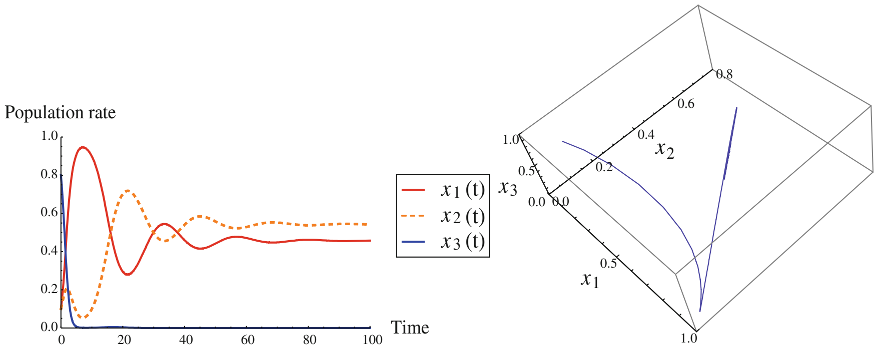


Fig. 8. $x_1(t)$, $x_2(t)$, $x_3(t)$

5 Conclusions

This paper deals with modeling of network's dynamic using evolutionary games approach. It is presented the model of network's dynamic using differential equations with discontinuous right side and proved existence and uniqueness of solution, formulated payoff matrix for a network AIMD connections game and found conditions of equilibrium existence depending of loss sensitivity parameter. The results are illustrated by simulations using Wolfram Mathematica.

References

1. Chiu, D.M., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Comput. Netw. ISDN Syst. **17**(1), 1–14 (1989)
2. 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**, 237–252 (1998)
3. Mo, J., Walrand, J.: Fair end-to-end window-based congestion control. IEEE/ACM Trans. Netw. (ToN) **8**(5), 556–567 (2000)
4. Paganini, F., Doyle, J., Low, S.: Scalable laws for stable network congestion control. In: Proceedings of the 40th IEEE Conference on Decision and Control, vol. 1, pp. 185–190. IEEE (2001)
5. Low, S.H., Srikant, R.: A mathematical framework for designing a low-loss. Low-Delay Internet. Netw. Spat. Econ. **4**(1), 75–101 (2004)
6. Altman, E., El-Azouzi, R., Hayel, Y., Tembine, H.: The evolution of transport protocols: an evolutionary game perspective. Comput. Netw. **53**(10), 1751–1759 (2009)
7. Smith, J.M.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
8. Altman, E., Bonneau, N., Debbah, M., Caire, G.: An evolutionary game perspective to ALOHA with power control. In: Proceedings of the 19th International Teletraffic Congress, Beijing, China (2005)
9. Han, Z.: Game Theory in Wireless and Communication Networks: Theory, Models, and Applications. Cambridge University Press, Cambridge (2012)
10. Meyn, S.P.: Control Techniques for Complex Networks. Cambridge University Press, Cambridge (2008)
11. Filippov, A.F., Arscott, F.M. (eds.): Differential Equations with Discontinuous Righthand Sides. Mathematics and its Applications, vol. 18. Springer, Heidelberg (1988)
12. Border, K.C.: Fixed Point Theorems with Applications to Economics and Game Theory. Cambridge Books, Cambridge (1990)

Forming an Evolutionarily Stable Firm Strategy Under Cournot Competition Using Social Preferences

Vitaliy Kobets(✉) and Maksim Poltoratskiy

Kherson State University, 27, 40 Rokiv Zhovtnya St, Kherson 73000 Ukraine
vkobets@kse.org.ua, maksimpoltoratskii@mail.ru

Abstract. The paper investigates what kinds of firms' social preferences (egoists, altruists or punishers) prove to be evolutionarily stable strategies under Cournot competition in the market of homogeneous product. Combining the traditional and evolutionary approaches to competition we define the conditions that forecast the attitude and profit dynamics of competitors under Cournot competition and adaptive behavior of rivals.

Keywords: Evolutionarily stable strategies · Institutions · Mechanisms · Social preferences · Experiment

1 Introduction

People make decisions under the influence of institutions. The mechanism of interaction between people under existing institutions brings a result that stimulates participants to revise their strategies.

Consider the basic components of environment interaction, such as people, institutions, mechanism and result (Fig. 1).

People have different incentives for actions, as characterized by different attitudes to their rivals. These relations consist of social preferences, which include selfishness, altruism, punishment, cooperation and so on.

Classical economics perceives an individual as an "economic man" (selfish person), who seeks to achieve his optimization choice exclusively to his advantage in an environment without institutions (fixed norms of behavior for the vast majority of participants). Selfishness generates:

- positive effects: invisible hand of market that means compatibility of individual and social interests;
- negative effects: tragedy of commons that implies the contradistinction between individual and social interests, which harms everyone.

Individualism generates selfishness. The individual cares only about his own welfare and ignores the rest. The more goods he receives, the more utility he benefits [6]. The benefit of one agent does not mean anything to the other [3].

It is interesting why seemingly irrational behavior of agent (altruism or retaliation) survives under economic interaction among firms. Traditional approach in economics

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 343–361, 2014.
DOI: 10.1007/978-3-319-13206-8_17

can not explain such firm's behavior under classical assumptions. But such behavior exists in real life. We can consider market equilibrium under dynamic interaction between firms using social preferences instead of one shot interaction to explain why altruism or retaliation can be evolutionarily stable firm strategies.

2 Related Works

Economic experiments and real life observations confirm that people often do not go for the deterioration of their own welfare for increasing the others' benefits (altruist), but for the sake of punishing those persons, who violate social norms (punisher). When the joint actions bring more benefits than the total benefit of the individual actions, then there are the prerequisites for participants' interaction. These social benefits are listed in Table 1.

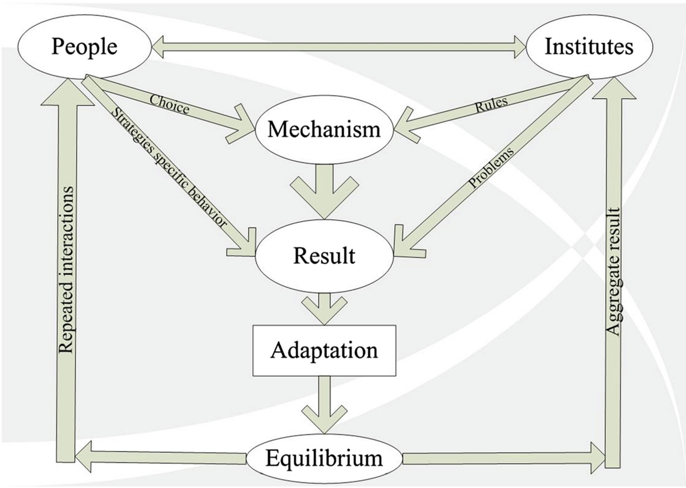


Fig. 1. Interaction of people and institutions under a competitive mechanism

Different social preferences of people explain the expansion of altruistic and punishing behavior in the environment, even when selfish behavior would bring more personal material gain. Altruistic and punishing behaviors are not considered in traditional economic models of market competition.

People tend to imitate others and be guided by social norms for cooperation and altruism. This behavior does not contradict rationality, combining individual selfishness with social norms [5]. Mutual gains from cooperation induce altruistic incentives to honest interaction, supported by social norms. The firm values the benefit of other participants as its own.

Table 1. Classification of social preferences

Cost/Benefit	Private costs	Private benefits
Outsider benefits	Altruists	Co-operators
Outsider costs	Punishers	Selfish persons

Observations and experiment give examples of such social benefits:

- voluntary behavior;
- Paying taxes without offshore optimization;
- compliance with norms and laws when they can not be verified.

The behavior of people is not uniform, as some are concerned only about their own benefits, while others - about social interests; so different groups of participants will be presented at the beginning of their interaction.

Besides, the type of individual behavior can vary or even change, as some individuals pursue their policy, while others adapt to a more favorable one (conformists). This explains the transformation of egoist behavior into cooperation for common benefit, and vice versa.

Under limited numbers of participants their actions are interrelated. This creates strategic behavior of individuals. This behavior is focused on the future results and takes into account the expected behavior of others.

Changing the individual's attitudes to competitors can permanently change his behavior if it increases his benefits.

Everyone has the choice to maximize his profit function taking into account his relation to rivals.

After interaction and updating information about the environment, the behavior and attitude of competitors (their type) will change the strategy and attitude of each individual under finite or infinite repeated interaction (feedback of outcome and individual behavior in Fig. 1).

Schaffer proposed a concept of evolutionarily stable strategy (ESS) for small population [11] as a generalization of the standard ESS concept for large population by M. Smith. He demonstrated that a finite population ESS is not generally a Nash equilibria strategy. Schaffer applied this concept to economic game and showed that the strategy that survives in economic natural selection is the relative payoff maximizing strategy [10].

Vega-Redondo studied a stochastically stable state in a symmetric oligopoly with homogeneous goods and showed that perfect competitive behavior is a stochastically stable state, where firm spends most of the time in long-run period [15]. It is proved that dynamics of differentiated oligopoly is not a one-shot Nash equilibrium [8].

When all firms choose same value of strategic variable (price or quantity), but one firm (experimenter, entrant) chooses another one and his profit will increase then other firms strive to follow the new strategy after the previous strategy becomes unstable and the new strategy is a globally surviving strategy GSS [13]. Unique GSS output is between the Nash-Cournot equilibrium and the competitive output for quantity-setting oligopoly [13].

ESS is defined as the Nash equilibrium strategy in an evolutionary game, which is stable in the sense that (a) it is the best response against itself in the population and (b) once it is fixed, no alternative strategy can invade the population successfully [7].

ESS coincides with the consistent conjectural variations of the static duopoly game. Only those players who maximize their payoff functions taking into consideration the right conjectures about the rival firm's behavior will be more successful in the market, and these conjectures are the ones that remain in an evolutionary game too.

The market output and price induced by ESS can be socially optimal only if the firms produce homogeneous products. Moreover, if firms produce heterogeneous goods, the market output under ESS is less than socially optimal ("underproduction"), while the price exceeds the socially optimal price.

With the spite effect, learning through imitation and mutation diverts the attention of firms from their own profits and prevents the emergence of dynamics based on best replies that could assure the evolutionary stability of the Cournot equilibrium. The Walrasian equilibrium becomes the only ESS equilibrium in this case [14].

Recently game-theorists have established the connection between conventions, bounded rationality, learning and evolutionary models. The idea has been popularized in the social sciences by Axelrod. The central notion is very simple. Firms follow 'strategies', or rules which tell them what to do: different agents try out different strategies. Some strategies are more successful than others. Over time, successful strategies will become more common, either through a form of propagation, or imitation. Hence strategies that lead to firms being more profitable will tend to predominate over time. We can then explain the strategies of firms as being the result of such a process of social evolution. There are strong evolutionary forces driving behavior towards cooperation. Collusion is robust against invasion of competitive strategy [2].

The paper has the following structure: the first part is devoted to literature review; the second one determines problem statement; the third part constructs the general model of social preferences under Cournot competition; the fourth part models evolutionarily stable strategy for firms' social preferences under Cournot competition by the means of developed software; the fifth part concludes.

3 Problem Statement

Institutions are laws, informal rules and agreements that provide long-run foundation for social interactions between population members [1].

Institutions are supported by rules, which are protected by:

- *laws*– centralized compulsion;
- *informal rules* - social sanctions;
- *agreements* - mutual expectations (e.g., from a previous experience).

Institutes accumulate experience of individuals' interactions. The best experience is such that gives the greatest benefit to all participants of society. This experience is summarized in the form of norms, i.e. rules of interaction that apply to all or the vast majority of society participants. Rules become effective institutions, where participants are convinced that failure to comply with these rules will penalize them.

The rules can be formal (government laws) or informal (customs and traditions).

The task for institutions is to maintain or to extend the strategy of the most successful participants of interactions. Repeated interaction (cut and try method) via empirical data about common result provides information about the effectiveness of applied social norms [12].

The norms that increase the benefit of the majority of participants will expand, and those reducing the benefit will disappear (feedback of result and institutions on Fig. 1). Thus, institutions are well modeled as games.

Institutions may change as equilibrium of certain games. When institutions are long-run, the games have a stable equilibrium. When defined set of mutual best responses becomes universal for all participants and is saved for a long time, it becomes a constituent element of the institute. The best institutions are created not as agreements, but as random foundations or historical experience that proved to be effective.

Institutions create conditions for mechanism of transforming social preferences [4]. Society is interested in creating institutions that are self-regulated, that can work satisfactorily without a leader.

Coordination of results and social goals requires the corresponding mechanism of participants' interaction. The mechanism transforms the individual decisions into social outcome according to rules.

Result depends on the level of interaction between participants (efforts coordination or conflict of interest) defined by the methods of game theory. Multiple repeated interactions adapt strategies of individuals and society norms to equilibrium state. Market agents are adaptive, since they have the intelligence ability to take into account relevant information that can change their behavior.

Adaptation process is reproduced by means of computer simulation, which does not allow obtaining general conclusions, but may indicate the presence of unique conclusions [9].

The evolution of the society (social preferences) during adaptation of people's strategies determines the dynamics of society behavior. Evolution forms the retrospective type of behavior that proved to be effective in the past. Changing the direction or speed of evolution can be caused by: accidents, replication (reproduction of successful strategies), initial conditions (history) system. The evolution of institutions is an example of non-cooperative interaction with a common interest.

In the course of evolution process the system reaches single or multiple equilibriums. If equilibrium is multiple, the system can drift from one equilibrium to another, forming punctuated equilibriums, under the influence of exogenous factors (intentional or unintentionally suboptimal responses, chance, history, innovation). With multiple equilibrium, evolutionarily stable strategies are those in which the equilibrium in system is maintained a longer period of time.

The distinguishing features of the evolutionary approach are:

1. *modeling accidents*: behavioral innovations that are not the best response, the difference between the actual and expected results.

A minor unlikely event often has a significant impact on long-run outcome through positive feedback. Random events influence the direction, not only on the rate of evolution change.

2. *differential replication* (selection) sets the direction of the evolutionary process. Replication is a copying and dissemination of behavioral characteristics of individuals and society, while competing rules, beliefs and preferences disappear (Table 2).

Table 2. Types of differential replication

Replicator	Selection level	
	Individual	Group
Studied behavior	Social learning: conformism, the best response, consolidating training	Copying behavior of successful groups, absorption of unsuccessful groups

Payments measure reproductive success of individual types (adjustment). Differential adjustment is a rate at which individuals change their behavior for the new one. The reasons of differential copying: (i) conformity - the behavior adopted in this environment; (ii) consolidating training - the behavior that yielded greater gains in the past; (iii) updating the best reaction - the maximization of expected individual's gain in relation to other participants. The process of changing individual's type to a different one is called cultural sign that is acquired rather than succeeded.

The subject of replication can be the types of firms participating in market interaction. This approach is new, because such types used to be considered for biological competition, rather than economic competition.

3. *nonequilibrium state* describes the parameters of the system when a system leaves or fails to reach the steady state. It shows the coexistence of different types of participants.

Mostly altruist and punisher models are used in the theory of consumption. It is interesting to investigate how the model of altruist and punisher affects the behavior of firms in the market.

The paper **goal** is to investigate which social preferences of market participants are evolutionarily stable strategies under Cournot competition on homogeneous market product.

4 General Model of Firms' Social Preferences under Cournot Competition

Consider n firms that produce homogeneous product and compete in quantity in the market (Cournot model). Each of the firms may have the following attitude to the rivals:

1. selfish (α = 0): $\pi_i = P \cdot q_i - v_i \cdot q_i \xrightarrow{q_i \geq 0} \max$;
2. altruistic (α > 0) $\pi_i = P \cdot q_i - v_i \cdot q_i + \alpha_i \cdot \sum_{j \neq i} \pi_j \xrightarrow{q_i \geq 0} \max$;

3. punishing (α < 0) $\pi_i = P \cdot q_i - v_i \cdot q_i + \alpha_i \cdot \sum_{j \neq i} \pi_j \xrightarrow{q_i \geq 0} \max.$

Selfish acts only in his own interests, e.g. a firm only maximizes its own profits, *Altruist* values his own profits and the profits of his competitors; *punisher* punishes the rest of the firms that make profit, despite the fact that he can suffer losses (in particular, he can only punish egoists).

Initially there is a distribution between these three types of attitudes to competitors. Each of the firms at the initial time has a type: the level of cost per unit or average cost v_i (unchanged throughout the interaction) and attitude to rivals α_i (random distributed, which may change during the interaction).

Model task is to investigate which social preferences of firms are evolutionarily stable strategies in a Cournot competitive market.

So the question is which attitude to the competitors will be resistant to the changes in the course of repeated interaction (repeated games) between participants in this model.

Model assumptions:

- n firms in the market participate in Cournot competition;
- market demand and the cost of competitors is *common* knowledge, e.g. information about these figures symmetrically distributed among all firms;
- inverse demand function $P = b - c \cdot Q$ is linear on product;
- the goal of each firm is to maximize its own objective function subject to its attitude to competitors (egoistic, altruistic, punitive);
- the product of firms is homogeneous;
- return to scale is constant (average cost and marginal cost of firm are equal).

In general, the profit of each firm in the market can be presented as follows: $\pi_i = P \cdot q_i - v_i \cdot q_i + \alpha_i \cdot \Pi_{-i}$, where π_i - profit of firm i; P - product price; q_i - product quantity of firm i; Q - industry output, which includes n firms; b - maximal market price under zero industry output; c - marginal change of price after quantity unit change; v_i - marginal (average) cost or cost per unit for firm i; Π_{-i} - total profit of all firms except i, α_i - attitude of firm i to competitors, $\alpha_i \in [-1; 1]$, where $\alpha_i = 1$ means *absolute altruist* (profits of other firms regarded as one's own), $\alpha_i = -1$ shows the *absolute punisher* (fully prepared to sacrifice his own profit, in order to leave the rest of the firms without profit).

The objective functions of n participants are represented as the following equation system:

$$\begin{cases} \pi_1 = P \cdot q_1 - v_1 \cdot q_1 + \alpha_1 \cdot \sum_{j \neq 1} \pi_j, \\ \pi_2 = P \cdot q_2 - v_2 \cdot q_2 + \alpha_2 \cdot \sum_{j \neq 2} \pi_j, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \pi_{n-1} = P \cdot q_{n-1} - v_{n-1} \cdot q_{n-1} + \alpha_{n-1} \cdot \sum_{j \neq n-1} \pi_j, \\ \pi_n = P \cdot q_n - v_n \cdot q_n + \alpha_n \cdot \sum_{j \neq n} \pi_j. \end{cases} \tag{1}$$

Shifting the profit functions of all participants to the left side of each system Eq. (1), we get:

$$\begin{cases} \pi_1 - \alpha_1 \cdot \pi_2 - \alpha_1 \cdot \pi_3 - \ldots - \alpha_1 \cdot \pi_n = (P - v_1) \cdot q_1, \\ -\alpha_2 \cdot \pi_1 + \pi_2 - \alpha_2 \cdot \pi_3 - \ldots - \alpha_2 \cdot \pi_n = (P - v_2) \cdot q_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ -\alpha_{n-1} \cdot \pi_1 - \alpha_{n-1} \cdot \pi_2 - \ldots + \pi_{n-1} - \alpha_{n-1} \cdot \pi_n = (P - v_{n-1}) \cdot q_{n-1}, \\ -\alpha_n \cdot \pi_1 - \alpha_n \cdot \pi_2 - \ldots - \alpha_n \cdot \pi_{n-1} + \pi_n = (P - v_n) \cdot q_n. \end{cases} \tag{2}$$

Represent system (2) in matrix form:

$$\left(\begin{array}{cccccc|c} 1 & -\alpha_1 & -\alpha_1 & \ldots & -\alpha_1 & -\alpha_1 & (P - v_1) \cdot q_1 \\ -\alpha_2 & 1 & -\alpha_2 & \ldots & -\alpha_2 & -\alpha_2 & (P - v_2) \cdot q_2 \\ -\alpha_3 & -\alpha_3 & 1 & \ldots & -\alpha_3 & -\alpha_3 & (P - v_3) \cdot q_3 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -\alpha_{n-1} & -\alpha_{n-1} & -\alpha_{n-1} & \ldots & 1 & -\alpha_{n-1} & (P - v_{n-1}) \cdot q_{n-1} \\ -\alpha_n & -\alpha_n & -\alpha_n & \ldots & -\alpha_n & 1 & (P - v_n) \cdot q_n \end{array}\right) \tag{3}$$

Expressing the profit of each firm as a function of the *n* firm's profit by identical transformations of system (3), we obtain:

$$\begin{cases} \left(1 + \frac{1}{\alpha_1}\right)\pi_1 - \left(1 + \frac{1}{\alpha_n}\right) \cdot \pi_n = \frac{1}{\alpha_1}(P - v_1) \cdot q_1 - \frac{1}{\alpha_n}(P - v_n) \cdot q_n, \\ \left(1 + \frac{1}{\alpha_2}\right)\pi_2 - \left(1 + \frac{1}{\alpha_n}\right) \cdot \pi_n = \frac{1}{\alpha_2}(P - v_2) \cdot q_2 - \frac{1}{\alpha_n}(P - v_n) \cdot q_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \left(1 + \frac{1}{\alpha_{n-1}}\right)\pi_{n-1} - \left(1 + \frac{1}{\alpha_n}\right) \cdot \pi_n = \frac{1}{\alpha_{n-1}}(P - v_{n-1}) \cdot q_{n-1} - \frac{1}{\alpha_n}(P - v_n) \cdot q_n, \\ \frac{1}{\alpha_n} \cdot \pi_n - \Pi_{-n} = \frac{1}{\alpha_n}(P - v_n) \cdot q_n. \end{cases} \tag{4}$$

The dependence between the profits of the first *n−1* firms and profit of firm *n* is explicitly represented by the following equations system:

$$\begin{cases} \pi_1 = \frac{\alpha_1 \cdot (\alpha_n + 1)}{\alpha_n \cdot (\alpha_1 + 1)} \cdot \pi_n + \frac{1}{\alpha_1 + 1}(P - v_1) \cdot q_1 - \frac{\alpha_1}{\alpha_n \cdot (\alpha_1 + 1)}(P - v_n) \cdot q_n, \\ \pi_2 = \frac{\alpha_2 \cdot (\alpha_n + 1)}{\alpha_n \cdot (\alpha_2 + 1)} \cdot \pi_n + \frac{1}{\alpha_2 + 1}(P - v_2) \cdot q_2 - \frac{\alpha_2}{\alpha_n \cdot (\alpha_2 + 1)}(P - v_n) \cdot q_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \pi_{n-1} = \frac{\alpha_{n-1} \cdot (\alpha_n + 1)}{\alpha_n \cdot (\alpha_{n-1} + 1)} \cdot \pi_n + \frac{1}{\alpha_{n-1} + 1}(P - v_{n-1}) \cdot q_{n-1} - \frac{\alpha_{n-1}}{\alpha_n \cdot (\alpha_{n-1} + 1)}(P - v_n) \cdot q_n. \end{cases} \tag{5}$$

From the last equations system (4) profit function of firm n is: $\pi_n = \alpha_n \cdot \Pi_{-n} + (P - v_n) \cdot q_n$. To find the total profit of all firms except n Π_{-n} by addition of the respective left and right sides of equations (5), we have:

$$\Pi_{-n} = \frac{\alpha_n + 1}{\alpha_n} \cdot \pi_n \cdot \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1} + \sum_{j \neq n} \frac{1}{\alpha_j + 1} \cdot (P - v_j) \cdot q_j - \frac{1}{\alpha_n} \cdot (P - v_n) \cdot q_n \cdot \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}. \tag{6}$$

Then profit function of firm n will look as:

$$\pi_n = (\alpha_n + 1) \cdot \pi_n \cdot \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1} + \alpha_n \cdot \sum_{j \neq n} \frac{1}{\alpha_j + 1} \cdot (P - v_j) \cdot q_j - (P - v_n) \cdot q_n \cdot \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1} + (P - v_n) \cdot q_n. \tag{7}$$

Or after identical transformations of firm's n profit function will be:

$$\pi_n = \frac{\alpha_n \cdot \sum_{j \neq n} \frac{1}{\alpha_j + 1} \cdot (P - v_j) \cdot q_j + (1 - \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}) \cdot (P - v_n) \cdot q_n}{1 - (\alpha_n + 1) \cdot \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}}. \tag{8}$$

First order condition of profit maximization (7): $\frac{\partial \pi_n}{\partial q_n} = 0$ is equivalent to the following equation:

$$q_n = \frac{b - v_n}{2c} - \frac{1 + \frac{\alpha_n}{\alpha_1 + 1} - \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}}{2 \cdot \left[1 - \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}\right]} \cdot q_1 - \ldots - \frac{1 + \frac{\alpha_n}{\alpha_{n-1} + 1} - \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}}{2 \cdot \left[1 - \sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1}\right]} \cdot q_{n-1} \tag{9}$$

Condition (9) defines the reaction function of firm n to product quantities of all its rivals in which firm n achieves maximum of its objective function under given attitude to competitors.

Second order condition of profit maximization (9): $\frac{\partial^2 \pi_n}{\partial q_n^2} < 0$ is identical to the following condition: $\sum_{j \neq n} \frac{\alpha_j}{\alpha_j + 1} < 1$.

Each firm determines its optimal output according to its function response.

Then the firm compares its own profit with the average across the industry. If its profit is higher than the industry average, the firm does not change its type, but changes the level of its attitude to the rivals: for example, level of altruism or level of retaliation (e.g., 0.05).

If its profit is lower than the industry average, the firm compares its profit with the average across each group of other participants (egoists, altruists or punishers). The firm will change its type for the type of such group, where the average profit is the highest.

5 Modeling Evolutionarily Stable Firm Strategy Using Social Preferences Under Cournot Competition

Consider the case of interaction between two firms $n = 2$. From Eqs. (8) and (9) we obtain the following results:

$$q_i = \frac{(1-\alpha_i)\cdot b - 2\cdot v_i + (1+\alpha_j)\cdot v_j}{c\cdot\left[4-(\alpha_i+1)\cdot(\alpha_j+1)\right]},\ i = 1,2,\ j = 2,1,\ i \neq j \tag{10}$$

$$Q = \sum_{i=1}^{2} q_i = \frac{(1-\alpha_1)\cdot(b-v_2) + (1-\alpha_2)\cdot(b-v_1)}{c\cdot[4-(\alpha_1+1)\cdot(\alpha_2+1)]} \tag{11}$$

$$P = \frac{(1-\alpha_1\cdot\alpha_2)\cdot b + (1-\alpha_1)\cdot v_2 + (1-\alpha_2)\cdot v_1}{4-(\alpha_1+1)\cdot(\alpha_2+1)} \tag{12}$$

$$\pi_i = \frac{\alpha_i\cdot(P-v_j)\cdot q_j + (P-v_i)\cdot q_i}{1-\alpha_i\cdot\alpha_j},\ i = 1,2,\ j = 2,1,\ i \neq j \tag{13}$$

Let us take $b = 100$, $c = 1$, $v_1 = 2$, $v_2 = 3$ as initial parameters. We can interpret b as maximal price on market (100 %) under zero industry output and c as price reaction on 1 % quantity change (unit price elasticity allows to consider more generalized case of price change between elastic and inelastic market demand). We assume that firms use highly productive technologies. Because its average costs v_1 and v_2 are substantially lower than maximal market price. In this interaction the model will hold two basic assumptions:

- Firms act rationally under quantitative competition, following their response functions (9).
- Firms adapt to such social interaction that gives more profits. This social interaction can be represented by the following algorithm:

$$\begin{aligned}
&\text{If } \pi_1 \geq \pi_2,\\
&\text{then If } \alpha_1 \geq \alpha_2,\\
&\text{then } \alpha_1 := \alpha_1 + 0.05 \text{ else } \alpha_1 := \alpha_1 - 0.05\\
&\text{else If } \alpha_1 < \alpha_2,\\
&\text{then } \alpha_1 := \alpha_1 + 0.05 \text{ else } \alpha_1 := \alpha_1 - 0.05.
\end{aligned} \tag{14}$$

This algorithm shows that if the first firm's profit is higher than the second one's, and the attitude of the first firm to its rival is better, then the attitude of the 1st firm to the 2nd will further be improved, and vice versa. If the attitude of larger profit firm to its rival gets worse, it will further deteriorate, and vice versa.

Each firm may be an egoist, altruist or punisher.

For modeling of firms' interaction was developed program module 'Altruist-egoist-punisher' (AEP). Given software is developed by the means of computer language C# specially for modeling of firms' evolutionarily stable strategies.

Software development based on the following technologies:

- Development environment Microsoft Visual Studio 2012;
- Computer language C#;
- Extensible Markup Language XML;
- Data manager SQL SERVER 2008.

XML is applied for saving of output computation, which can be used for publications these results on web-resources. Data manager SQL SERVER 2008 is used for saving of model estimated values.

The menu of software modules "AEP" provide saving calculations in the DOC-format, which can be used by the user as a report on the results of firms' interactions modeling for determined parameters.

Developed software AEP based on standard libraries of C#, including: System; System.Collections.Generic; System.ComponentModel; System.Data; System.Drawing; System.Linq; System.Text; System.Windows.Forms; System.IO; Word = Microsoft.Office.Interop.Word.

Proposed model of firms' behaviors is described by the following algorithm by the means of C#:

```
if (pi2 >= pi1)
 {
  if (alf1 >= alf2)
    {
     arrALFA2.Add(alf2);
     richTextBox2.Text += alf2.ToString() + "\n";
     while (alf2 > -1)
     {
      alf2 = alf2 - h;
      if (alf2 < -1) break;
      richTextBox2.Text += alf2.ToString() + "\n";
      arrALFA2.Add(alf2);
     }
    }
  else
    {
    arrALFA2.Add(alf2);
   richTextBox2.Text += alf2.ToString() + "\n";
   while (alf2 < 1)
    {
    alf2 = alf2 + h;
    if (alf2 > 1) break;
    richTextBox2.Text += alf2.ToString() + "\n";
    arrALFA2.Add(alf2);
    }
   }
 }
```

```
        else
        {
          if (alf2 >= alf1)
            {
              arrALFA2.Add(alf2);
              richTextBox2.Text += alf2.ToString() + "\n";
              while (alf2 > -1)
                {
                  alf2 = alf2 - h;
                  if (alf2 < -1) break;
                  richTextBox2.Text += alf2.ToString() + "\n";
                  arrALFA2.Add(alf2);
                }
              }
              else
              {
                arrALFA2.Add(alf2);
                richTextBox2.Text += alf2.ToString() + "\n";
                while (alf2 < 1)
                {
                  alf2 = alf2 + h;
                  if (alf2 > 1 || alf2 == 1)
                    break;
                  else
                    richTextBox2.Text += alf2.ToString() + "\n";
                  arrALFA2.Add(alf2);
                }
              }
}
```

Case 1. Consider the situation where both participants are egoists: $\alpha_1 = \alpha_2 = 0$, and their behavior is described by algorithm (14). In this case, the dynamics of their social preferences and profits are shown in Figs. 2 and 3, respectively. We obtain that under

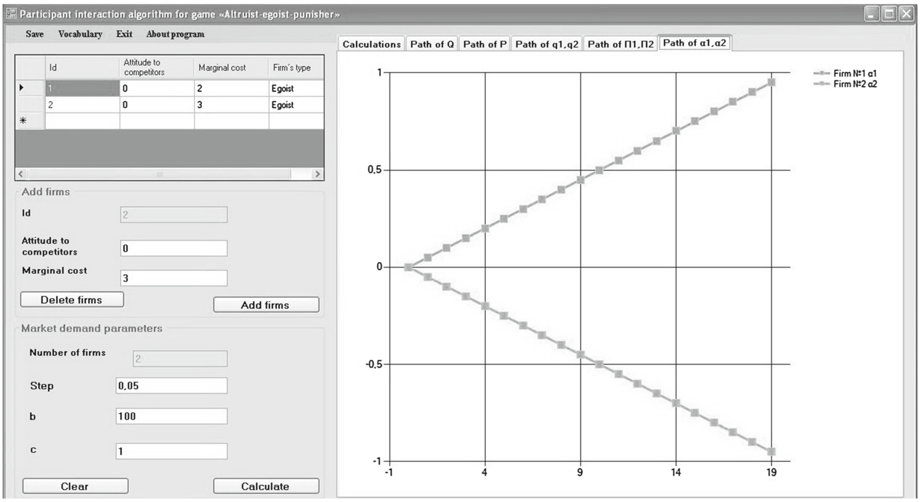


Fig. 2. Decreasing dynamics of social preferences of firms-egoists: $\alpha_1 = \alpha_2 = 0$

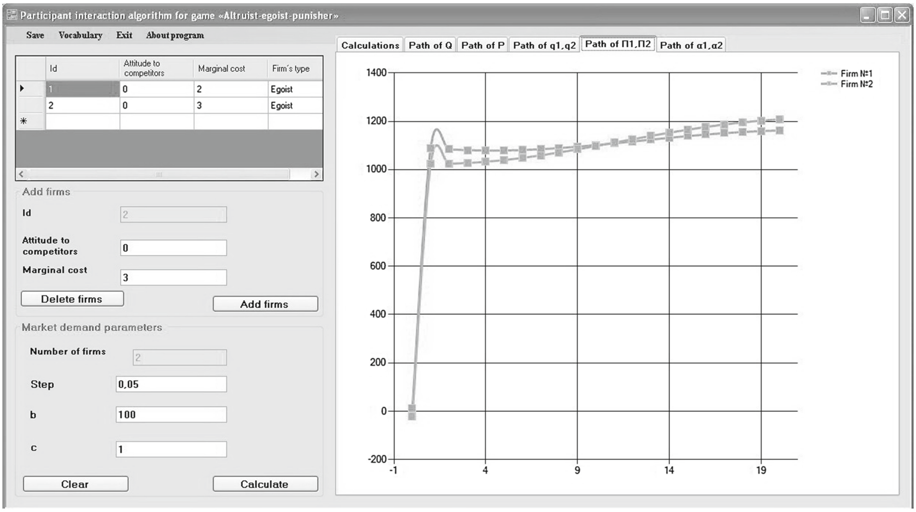


Fig. 3. Decreasing dynamics of firms-egoists' profits due to appropriate changes in their social preferences

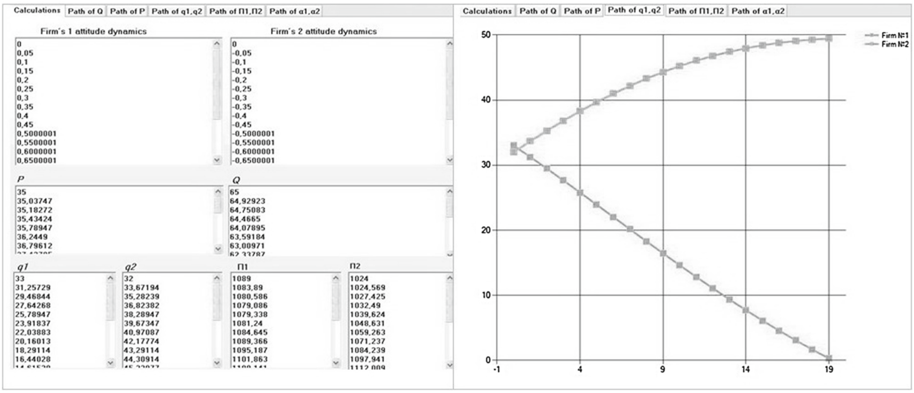


Fig. 4. Quantity dynamics for original firms-egoists under appropriate changes in their social preferences

quantitative competition low cost egoist becomes punisher and high cost egoist transforms in altruist. Their profits will increase simultaneously.

Firm 1 (altruist) will decrease his output q_1, whereas firm 2 (punisher) will increase his output q_2 (Fig. 4).

Under quantity competition and social preferences adaptation price will increase and total quantity will decrease (Fig. 5).

Case 2a. Consider the situation where both participants are altruists and the firm with lower costs is more altruistic: $\alpha_1 = 0,2$, $\alpha_2 = 0,1$; their behavior is described by the same algorithm (14). In this case, the dynamics of their social preferences and profits

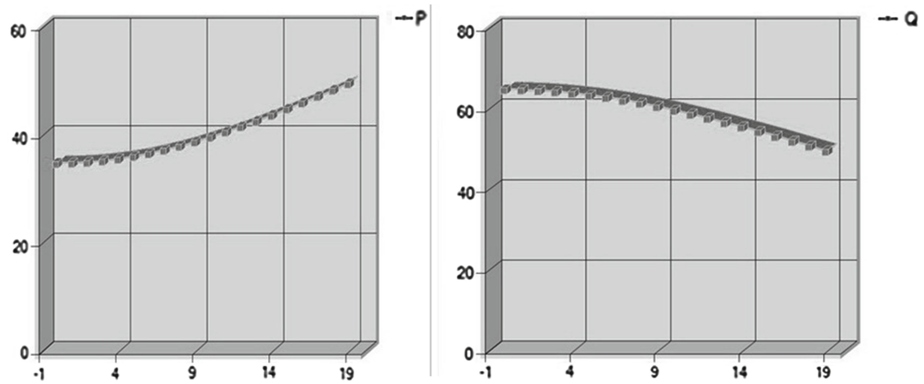


Fig. 5. Price and total quantity dynamics for original firms-egoists under appropriate changes in their social preferences

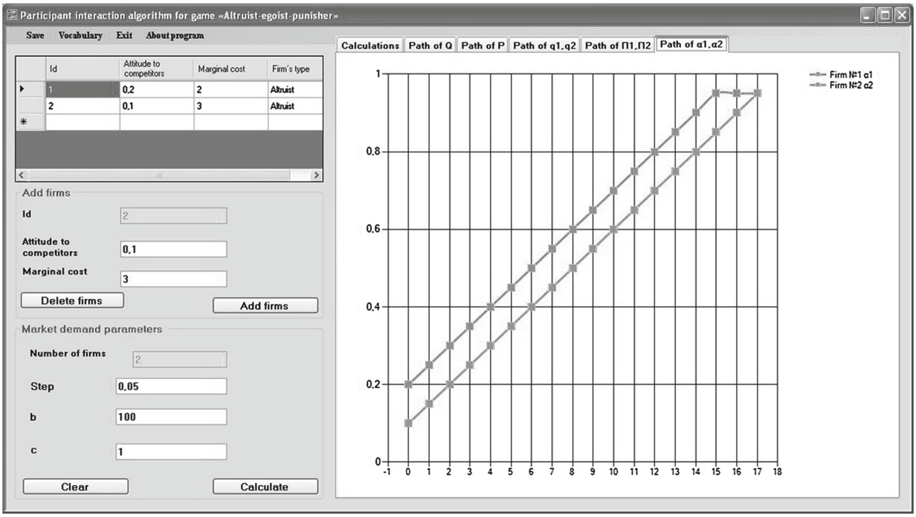


Fig. 6. Increasing dynamics of social preferences of firms-altruists (if the firm with lower costs is more altruistic: $\alpha_1 = 0,2$, $\alpha_2 = 0,1$)

are shown in Figs. 6 and 7, respectively. We derive that altruists under quantitative competition remain altruistic, which leads to a simultaneous increasing of their profits. The firm with lower average costs (AC) will have greater profit.

High cost altruist (firm 2) and low cost altruist (firm 1) have decreasing dynamics of their individual outputs (Fig. 8).

Case 2b. Consider the situation where both participants are altruists and the firm with lower costs is less altruistic: $\alpha_1 = 0,01$, $\alpha_2 = 0,1$, their behavior is described by the same algorithm (14). In this case the dynamics of their social preferences and profits are

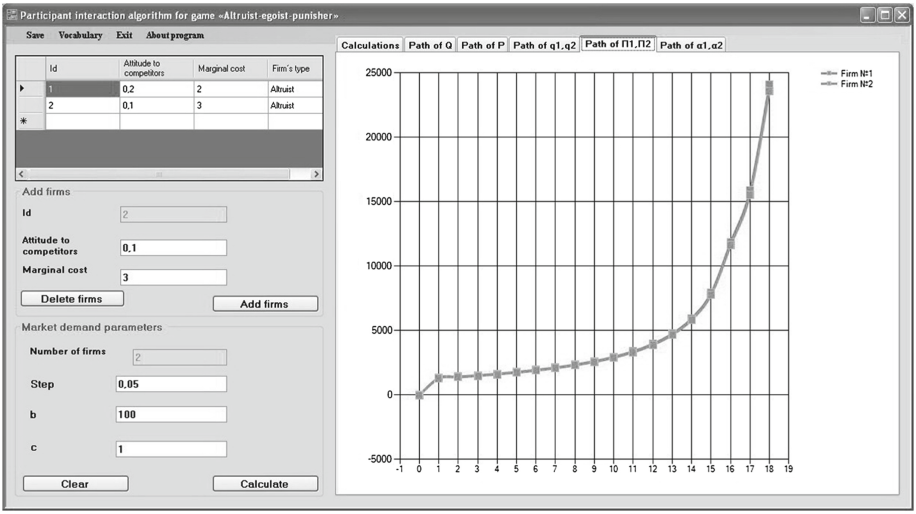


Fig. 7. Increasing dynamics of firms-altruists' profits (if the firm with lower costs is more altruistic: $\alpha_1 = 0,2$, $\alpha_2 = 0,1$)

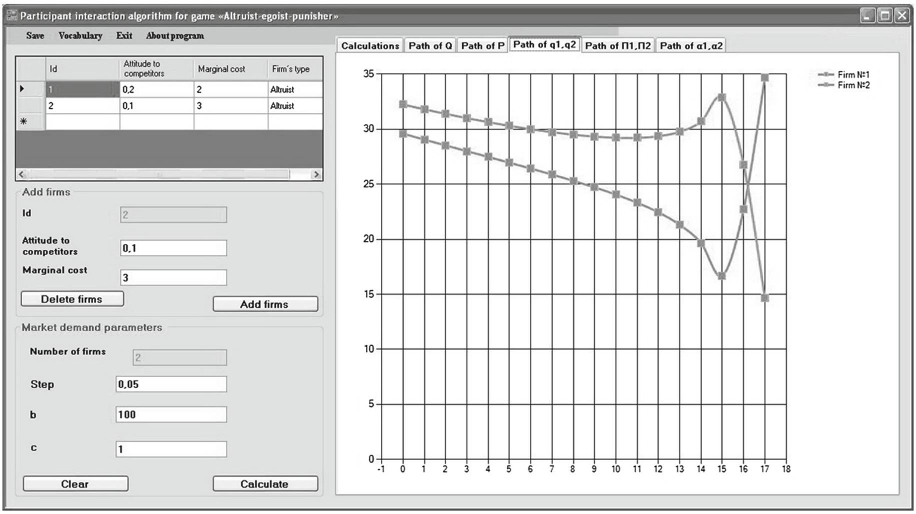


Fig. 8. Quantity dynamics for firms-altruist under appropriate changes in their social preferences

shown in Figs. 9 and 10, respectively. In such case altruists under quantitative competition become punishers, which leads to a drop in their profits. The firm with higher average costs will have lower profit.

After transformation of social preferences of firms from altruist to punishers, the output of each firm will increase (Fig. 11).

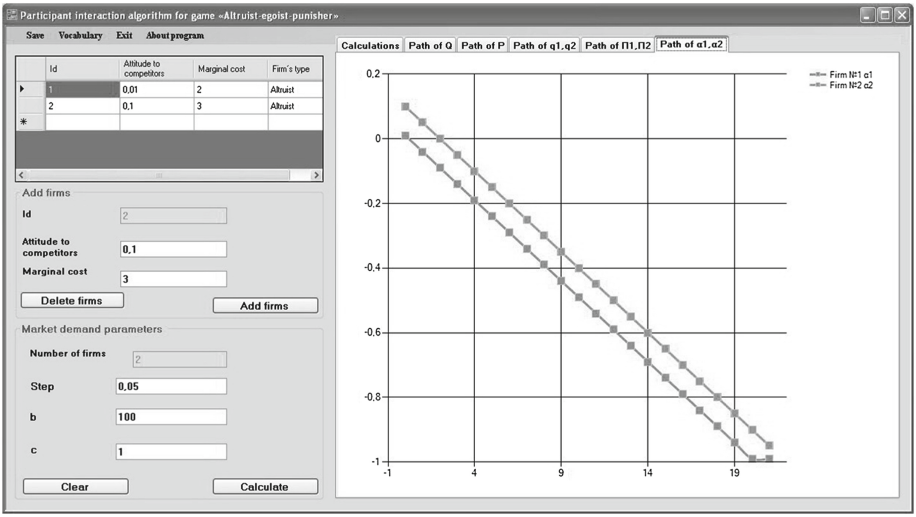


Fig. 9 Decreasing dynamics of social preferences of firms-altruists (if the firm with lower costs is less altruistic: $\alpha_1 = 0,01,\ \alpha_2 = 0,1$)

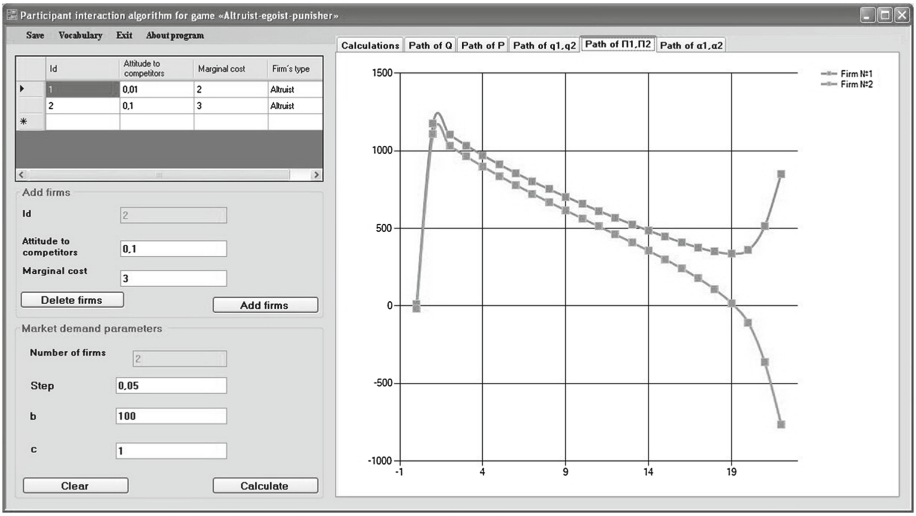


Fig. 10. Decreasing dynamics of firms-altruists' profits (if the firm with lower costs is less altruistic: $\alpha_1 = 0,01,\ \alpha_2 = 0,1$)

Similarly, by means of experiment we determine the dynamics of attitudes for firms-punishers and other types of firms. The results of the experiments are presented in Table 3.

50 % competitors under such adaptive behavior will become altruists and 50 % will become punishers. So successful rules (norms) for participants here are:

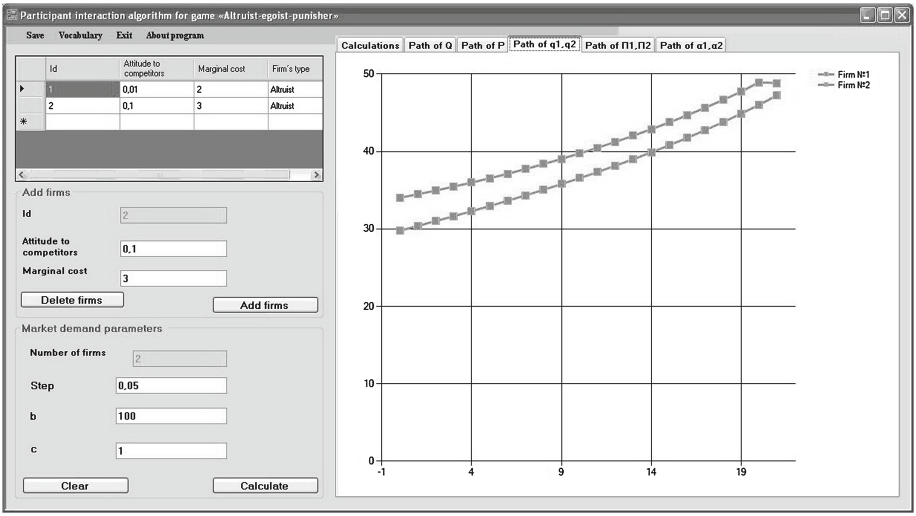


Fig. 11. Quantity dynamics for original firms-altruist which become punishers

Table 3. The change of attitudes between competitors under Cournot competition

Attitudes (social preferences)	Egoist	Altruist	Punisher
Egoist	**Punisher or altruist** (profits increasing)	If AC of egoists is higher, then ***<u>Altruists</u>*** (profits increasing) and vice versa	If AC of egoists is higher, then ***Punishers*** (profits decreasing) and vice versa
Altruist	If AC of altruist is higher, then ***<u>Punishers</u>*** (profits decreasing)and vice versa	If firm with lower AC is higher altruist, then **Altruists** (profits increasing) If firm with lower AC is lower altruist, then **Punishers** (profits decreasing)	If AC of altruist is higher, then ***<u>Punishers</u>*** (profits decreasing) and vice versa
Punisher	If AC of punisher is higher, then ***Altruists*** (profits increasing) and vice versa	If AC of punisher is higher, then ***<u>Altruists</u>*** (profits increasing) and vice versa	If firm with lower AC is lower punisher, then **Altruists** (profits increasing) If firm with lower AC is higher punisher, then **Punishers** (profits decreasing)

- The firm will expand his own attitude if it has lower average cost than its rival;
- The firm has to follow another attitude if it has higher average cost than its rival.

6 Conclusions

So we conclude that "selfishness" attitude is not an evolutionarily stable strategy (ESS). Firms-egoists under competition with other social preferences firms will be converted either to punishers, or to altruists, depending on what type of firm has more productive efficiency (e.g., lower average cost). If only selfish firms compete with each other, they become punisher or altruist.

The attitude of "altruism" is ESS if altruist have higher productive efficiency (e.g., lower average costs) under competition with selfish firm. If a firm with lower costs is more altruistic, the competition between firms leaves them altruists, and their attitude to each other is improving. But if a firm with lower average cost is less altruistic, then the two are eventually transformed into retaliation and "altruism" is not ESS in such case.

The attitude of "retaliation" is also ESS if punisher has higher productive efficiency (e.g., lower average costs) under competition with altruist. If a firm with lower costs is higher punisher, then the competition between punishers does not change their types and their attitudes to each other will worse. But if a firm with smaller average cost is lower punisher, the two firms become altruists and "punisher" is not ESS in this case.

The approach of evolutionary game theory shows under what conditions and in what direction the firms' attitudes will change. This approach considerably enhances the analysis and prediction of social preferences between firms in the market in comparison with traditional approach of Nash equilibrium.

References

1. Bowles, S.: Microeconomics: Behavior, Institutions, and Evolution. Princeton University Press, Princeton and Oxford (2004)
2. Dixon, H.D., Wallis, S., Moss, S.: Axelrod meets Cournot: Oligopoly and the evolutionary metaphor. Comput. Econ. **20**, 139–156 (2002)
3. Filatov, I.V.: Individualism and Holism in Social Perception. Modern West Philosophy (1998) (Dictionary, TON)
4. Kobets, V.: Mechanism design for foreign producers of unique homogeneity product. In: Ermolayev, V., et al. (eds.) Proceedings of 9-th International Conference ICTERI 2013, Kherson, Ukraine, 19–22 June 2013, CEUR-WS.org/Vol-1000, pp. 116–125. CEUR-WS.org/Vol-1000/ICTERI-2013-CEUR-WS-Volume.pdf. ISSN 1613-0073
5. McFadden, D.: The human side of mechanism design: a Tribute to Leo Hurwicz and Jean-Jacque Laffont. Rev. Econ. Des. **13**, 77–100 (2009)
6. Myerson, R.: Optimal coordination mechanisms in generalized principal agent problems. J. Math. Econ. **10**, 67–81 (1982)
7. Rachapalli, S.R., Kulshreshtha, P.: Evolutionarily stable conjectures and social optimality in oligopolies. Theor. Econ. Lett. **3**, 12–18 (2013)
8. Rhode, P., Stegeman, M.: Non-nash equilibria of darwinian dynamics with applications to duopoly. Int. J. Ind. Organ. **19**, 415–453 (2001)

9. Bowles, S., Choi, J.-K., Hopfensitz, A.: The co-evolution of individual behaviors and social institutions. J. Theor. Biol. **223**, 135–147 (2003)
10. Schaffer, M.E.: Are profit-maximizers the best survivors? A darwinian model of economic natural selection. J. Econ. Behav. Organ. **12**, 29–45 (1989)
11. Schaffer, M.E.: Evolutionarily stable strategy for a finite population and a variable contest size. J. Theor. Biol. **132**, 469–478 (1988)
12. Smith, V.L.: Game Theory and Experimental Economics: Beginnings and Early Influences. Duke University Press, Durham (1992)
13. Tanaka, Y.: Stochastically stable states in an oligopoly with differentiated goods: Equivalence of price and quantity strategies. J. Math. Econ. **34**, 235–253 (2000)
14. Vallée, T., Yildizoglu, M.: Convergence in the finite Cournot oligopoly with social and individual learning. J. Econ. Behav. Organ. **2**, 1–27 (2009)
15. Vega-Redondo, F.: The evolution of walrasian behavior. Econometrica **65**, 375–384 (1997)

Plagiarism Detection Tools for Scientific e-Journals Publishing

Liliia Luparenko(✉)

Institute of Information Technologies and Learning Tools,
9, M. Berlynskogo St., Kiev 04060, Ukraine
lisoln@rambler.ru

Abstract. The paper deals with the concept of plagiarism, self-plagiarism and their basic types. Open access to scientific global experience in the Internet and the benefits of electronic format led to the appearance of a numerous array of works that duplicate each other. With the exponential growth in scientific production, plagiarism has become a huge problem for editors of scientific e-journals because it is too difficult to detect immediately its presence in the manuscript. The article's object is the plagiarism automatic detection tools for text documents. The purpose of the presented research is to consider this popular software and online resources and their possibility to check text similarity, carry out their detailed analysis test and evaluate them according to defined software specifications, productivity of functioning and obtained results during the checking. As a result, there are presented recommendations of tools that meet the needs of scholars and can be used by the editors of scientific journals.

Keywords: Plagiarism · Self-plagiarism · Plagiarism detection software comparative analysis · Scientific e-journals · Open Journal Systems

1 Introduction

Over the last fifty years there were revealed two revolutionary technological changes in scientific publishing:

a. *computerization* has significantly decreased the publishing process costs and led to the appearance a huge number of scientific journals and books in a paper format;
b. *transition* the full publishing cycle (submission-review-publication) *to the Internet* has extremely minimized the costs and accelerated the process of scientific communication by production of e-journals.

Ability to manage the editorial and publishing process, submit, review and read articles online has revolutionized many aspects of scientific publishing. Online publishing significantly reduces production costs for printing texts and graphics, as well as provides to readers, authors and publishers many new features such us:

a. inclusion of audio and video components to the research papers;
b. ability to interact through discussions and comments directly at the e-journals webpages.

V. Ermolayev et al. (Eds.): ICTERI 2014, CCIS 469, pp. 362–370, 2014.
DOI: 10.1007/978-3-319-13206-8_18

Moving the publishing process to the online environment should be achieved by careful selection of the service that will best meet with the needs of individual researches, research institution and journals editors.

Nowadays, Ukrainian scientific institutions more often use an open source solution to managing and publishing scholarly journals online (e-publishing systems) such as Open Journal Systems (OJS). An example of using such a kind of systems in Ukraine is the lead scientific electronic edition "Information Technologies and Learning Tools" (http://journal.iitta.gov.ua), that contains developments in sphere "ICT in education" and operates on the basis of OJS.

With the transition to electronic publishing more and more full text articles are instantly available to readers for free. In recent years, researchers have increasingly preferred to publish online via open access journals to disseminate their research papers to potential readers.

Open access to scientific global experience in the Internet and the benefits of electronic format, that allow easy to apply the principle "Copy & Paste", led to appearance a numerous array of works that duplicate each other. This situation is typical for students, inexperienced authors, teachers and scientists, who have additional administrative burden (e.g., institutes and research agencies chiefs, university departments' chiefs, etc.). The number of their research output and ratings directly affect the prospects for further career growth and opportunity of getting qualification degree. It reduces the quality of manuscripts and led to more frequent occurrence of *plagiarism* in scientific publications.

Naturally, each particular scientific research can't be isolated - it is surrounded with the array of literature on specific scientific issues. Borrowing of short statements or small text parts is acceptable, common and even compulsory phenomenon: the process of generating new knowledge is based on the previous works ideas. Author can describe the overall concept, propose a new way of problem solving or argue the research results basing on the works of predecessors, but do not copy them!

With the exponential growth in scientific production, plagiarism has become a huge *problem for editors of scientific e-journals* because it is too difficult to detect immediately its presence in the manuscript. According to the statements of Committee on Publication Ethics (COPE), editors are responsible for all published content, that's why one of their main duties is to check the manuscripts for plagiarism and violations of ethical standards of research, because it significantly affects not only particular edition or author reputation, but also on the development of the field of science in general.

Classes of specific software and online resources designed for automatic textual plagiarism detection have been created with the development of information technology. In the scientific literature, research in automatic plagiarism detection focuses on three main areas:

a. analysis of existing algorithms of textual matches detection [5, 12] as well as development and technical description of new methods [14, 17, 21, 22, 26, 27, 28, 29];
b. classification of plagiarism detection tools [23, 24];
c. comparative analysis of existing software [3, 6, 10, 13, 16, 19, 20] as well as the design and development of new ones [1, 8, 13].

In a number of the above papers, as well as at numerous websites and personal blogs attempts were repeatedly made to analyze the existing plagiarism detection tools, but haven't been justified by what criteria and for what reason certain resources were selected. Still full-scale studies haven't been made for such a kind of tools by the single set of features and a clear division on software and online resources.

So the *article's goal* is the analysis of plagiarism detection software solutions to determine the most optimal resource that can be used in the process of scientific e-journals publishing.

2 Theoretical Background of the Plagiarism Phenomenon

In the legislative and scientific literature, the concept of plagiarism is determined as follows:

- "promulgation (publication), fully or partially, of another's work under the name of a person who is not the author of this work" [11];
- "attribution of authorship to someone other's work of science, literature, art, invention or an innovative proposal, as well as using another's work without reference to the author" [15];
- "kind of copyright violation and illegally using someone else's work (research, literature, music, invention or innovative proposals), fully or partially, under own name without specifying the source of borrowing …" [4].

Also, plagiarism is an illegal action to *enforce for co-authorship* [9], and can be seen as a type of *electronic crimes*, such us hacking, spreading viruses, spamming, and fishing in an electronic environment.

Nowadays, the most widespread tendency in the Ukrainian scientific community is phenomenon of *self-plagiarism* – re-publishing of large text parts from own scientific papers by the author without stating the fact of their prior or simultaneous publication.

Self-plagiarism has two main forms: a) re-publication of the previous article with a minor title and content changes ("double" or "multiple publication"), as well as b) reuse of textual fragments of previously published research papers without specifying links to them ("salami slicing") [6].

From the law point self-plagiarism is an act of fraud, but not an offense against intellectual property.

The following actions clear characterize the process of plagiarism [18]:

- turning in someone else's work as your own;
- copying another person's words or ideas without reference to its work;
- intentional omission the quote from the reference list;
- providing incorrect source data (such as "broken" links);
- changing words order, while preserving the overall structure of a sentence;
- copying large parts of text or ideas that makes up the majority of new article.

Plagiarism is classified in the following categories [16]:

a. the exact verbatim copying (Copy & Paste) without a proper bibliographic reference to the borrowed fragments;
b. copying with modifications in language, vocabulary and technological interpretation (the words switching, replacing letters, numbers, etc.);
c. style plagiarism;
d. translation from another language;
e. idea plagiarism.

The above categories describe *intentional* kind of plagiarism. Additionally distinguish two types: (a) *unintentional* (accidental similarity, convergence of ideas or discoveries made by various authors, independently from one another, etc.) and (b) *covert* (presenting the well known facts like own ideas without specifying the quote to source material)

According to the nature of research activity plagiarism is divided to two kinds [9]:

– *professional* – assignment of other person intellectual, creative and professional achievements for increase own authority, earnings, receipt of awards, recognition, etc.;
– *educational* – appropriation of another's intellectual property solely in the process of getting a scientific degree or educational qualification.

There are three types of plagiarism in education and research:

1. *Textual plagiarism* − full or partial copying of text fragments (modified or not) in the articles, theses, reports, books, manuscripts, theses, and so on.
2. *Source code plagiarism* – full or partial copying of source code written by another person (e.g. students) and submitted as their own development.
3. *Non-textual plagiarism* – illegal copying of data into spreadsheets, charts, scientific experiments datasets, photos and media resources.

There are two possible ways of plagiarism detecting:

– *manual detection* that are carried out directly by teachers, scholars, editors, readers, etc. The presence in the scientific journals editorial board conscientious, honest, conscientious reviewers who are experts in their field and thoroughly analyze manuscripts can greatly minimize the problem, but not eliminate it entirely.
– *automatic detection* using software tools.

However, professionals who are engaged in research activities (teachers, scholars, editors of scientific e-journals and organizers of scientific conferences) require only qualitative tools for in-depth manuscripts analysis on the text fragments convergence. Some of the above plagiarism types is easily detected by using wide range of proprietary and open source software and on-line solution.

3 Research Methods

During the research process we have analyzed the scientific literature, as well as a wide range of Internet sources, defined and selected a number of popular online resources and software tools designed to identify illegal "borrowing" in the text documents (Table 1).

Table 1. Textual plagiarism detection tools

Type	Tool's name	
Software	• Anti-Plagiarism • eTXT Antiplagiat • Advego Plagiatus • Double Content Finder	• Praide Unique Content Analyzer • Viper • Plagiata.NET (http://www.mywebs.ru/plagiatanet.html)
On-line resources	• Dupli Checker • PaperRater • Plagiarisma.net • PlagiarismChecker • Plagium, • PlagTracker • SeeSources • PlagScan • Plagiarism Detector	• Zashhita unikal'nosti kontenta (http://www.content-watch.ru/text) • FindCopy (Miratools) • Grammarly • Docol©c • Text.ru • Antiplagiat (antiplagiat.ru)

Selected resources were analyzed according to the following groups of characteristics:

I. Original data: the developer's name, web-site address, type of resource (proprietary/open source), availability of free trial access, the need for mandatory registration at web-site and contract with the developer, the number of research institution's users that have access to the program.

II. Functional and software specifications:
   a. the number of supported languages;
   b. file format, which operates the resource;
   c. database type (Internet/institutional repository/local system database) and the integration with the search engines;
   d. the types of checked items (manually entered text fragments/text files/web pages);
   e. limitations by entered text size;
   f. limitations by the number of checking;
   g. the algorithms of comparing documents used by the system.

III. Productivity:
   a. the checking time (from text entering till getting the summary report);
   b. the form of final result representation (statistical report);
   c. information about search results;

d. the features of use or install;
e. the interface features (intuitive clarity, availability of instructional and reference material).

In the process of comparing, the level of functionality and productivity of these tools as well as it's suitability to specifics of the research activities and features of Ukrainian language were estimated.

As a text testing sample in Ukrainian was selected scientific article of Bykov V.Yu. [2], published in electronic edition "Information Technologies and Learning Tools" – № 1 (2006). According to Google Scholar this paper has been most cited (50 times) for the entire period of the release of the e-journal that potentially have to provide a large number of matches in search results. Moreover, we found that this scientific work has been repeatedly copied by unscrupulous sites and spread like a student essay.

As a text testing sample in English was selected scientific paper of Dave E. Marcial, Mitzi S. Fortich, Jeambe B. Rendal [25], published in the latest issue – № 1 (2014) – of the same e-journal and duplicated only at the author's personal website.

Validation was carried out in the following ways: by introducing an arbitrary part of text in a special box on the site, loading the text file and checking an articles by it's URL-address in the journal.

4 Findings

4.1 Analysis of Textual Plagiarism Detection Software

After testing of selected 7 textual plagiarisms detection software, the following results were obtained.

1. *Double Content Finder (DCFinder), Praide Unique Content Analyser II* – not recommended for use because of the restrictions on use;
2. *Advego Plagiatus, Viper* and *Plagiata.NET* can be used by personal individual scholars for validation personal manuscripts. Given the rather long time required to check a single document these software tools are not useful for the preparation of scientific journals where necessary to carry out the verification of large array of manuscripts;
3. *Anti-Plagiarism* – recommended for use under conditions of subscription;
4. *eTXT Antiplagiat* – recommended for use by editors of scientific e-journals.

The details of comparative analysis are presented in Tables 1–3 of article appendix "Plagiarism Detection Tools Comparative Tables", that is available to download at: http://lib.iitta.gov.ua/4066/.

4.2 Analysis of On-line Resources for Textual Plagiarism Detection

After testing of selected textual plagiarism detection, 15 online resources were obtained the following results:

1. *PlagiarismChecker, Plagium, PlagTracker* – test failed due to an error in the resource;
2. *Plagiarisma.Net* and *Antiplagiat (antiplagiat.ru)* cannot be recommended because of the restrictions on use;
3. *Duplichecker, PaperRater, SeeSources, Plagiarism Detector* can be used for rapid nondeep analysis of text documents;
4. *PlagScan, Docol©c, Grammarly* – recommended for use under conditions of subscription;
5. *FindCopy(MiraTools), Zashhita unikal'nosti kontenta* (http://www.content-watch.ru/text) and *Text.ru* – recommended for use by editors of scientific e-journals.

The details of comparative analysis are presented in Tables 1–3 of article appendix "Plagiarism Detection Tools Comparative Tables", that is available to download at: http://lib.iitta.gov.ua/4066/.

The samples of *resulting reports* of testing the plagiarism detection tools are viewable at: http://lib.iitta.gov.ua/4064/.

5 Conclusion

As the result of our research can be concluded: the best tools that meet the needs of scholars and can be used by the scientific journals' editors are following:

– *eTXT Antiplagiat, FindCopy(MiraTools), Zashhita unikal'nosti kontenta* (http://www.content-watch.ru/text) and *Text.ru*;
– *Anti-Plagiarism, PlagScan, Docol©c, Grammarly* – recommended for use under conditions of subscription;
– *Advego Plagiatus, Plagiata.NET* – recommended for use under conditions of absence of lack of time to perform inspections and *Viper* – only for English texts.

These tools only detect the texts similarity and find a source of borrowing, but do not determine whether a particular piece of text is a primary source or a valid quote. Sometimes the quote is rated as duplication, and thus at the resulting report user get only the list of *potentially* plagiarized fragments. The final decision on the fact of plagiarism is left to the user. So the need of manual testing and human evaluation (peer reviewing) of the manuscripts still exists.

As each plagiarism detection tool has its own advantages and disadvantages in functional and software specifications, so the user cannot expect absolutely accurate results from electronic machines. But, at least, the implementation of this class of program solutions will have a number of positive consequences for Ukrainian science such as:

– maximum brake tendency to rapid spread of plagiarism;
– encourage researchers to move from a quantitative to a qualitative approach in their professional activities,
– a scientific journals editorial teams to review an editorial and publishing process standards and establish more stringent requirements for published materials.

References

1. Mokin, V.B., Voitko, V.V., Bevz, S.V.: Automated system of checking text for plagiarism. Visnyk Vinnytskoho Politekhnichnoho Instytutu **5**, 12–17 (2010) (in Ukrainian)
2. Bykov, V.Yu.: The theoretical and metodological grounds for modeling of educational environment of modern pedagogic systems. Information technologies and learning tools 1 (2006). http://journal.iitta.gov.ua/index.php/itlt/article/view/285/271 (in Ukrainian)
3. Bolilyi, V.O.: Check the uniqueness of the text in the assessment of student work creative or exploratory. Naukovi Zapysky NDU im. M. Hoholia **34**, 134–145 (2011). (in Ukrainian)
4. Prohorov, A.M. (ed.): The Big Soviet Encyclopedia, vol. 19. Sovetskaja Enciklopedija, Moscow (1975) (in Russian)
5. Kvashina, Ju.A.: Methods of finding duplicate texts arranged scientific style. Tehnologicheskij Audit i Rezervy Proizvodstva. **3**, 16–20 (2013). http://journals.uran.ua/tarp/article/viewFile/14893/12698 (in Russian)
6. Kotljarov, I.D.: Plagiarism in Scientific Publications: Nauchnaja Periodika. Problemy i Reshenija **4**, 6–12 (2011). (in Russian)
7. Linnik, I.: software to detect plagiarism: practical aspects. Naukovyi blog NaU ≪Ostrozka Akademiia≫ (2013). http://naub.oa.edu.ua/2013/prohramne-zabezpechennya-dlya-vyyavlennya-plahiatu-praktychnyj-aspekt/ (in Ukrainian)
8. Mykhailovskyi, YuB: Anti-plagiarism system as a tool for plagiarism preventing in educational and research activities. Visnyk Khmelnytskoho Natsionalnoho Universytetu. Tekhnichni Nauky **3**, 162–168 (2013). (in Ukrainian)
9. Petrenko, V.S.: The concept and types of plagiarism. Chasopys Tsyvilistyky **14**, 128–131 (2013). (in Ukrainian)
10. Popovskyi, O.I.: Review of Applications for the Comparative Analysis of Matches. In: Zbirnyk Tez Dopovidei 2ho Kirovohradskoho Sotsialno-Ekonomichnoho Forumu «Informatsiine Suspilstvo Vlada», Kirovohrad, pp. 99–100 (2013) (in Ukrainian)
11. On Copyright and Related Rights: Law of Ukraine on 23.12.1993 № 3792-XII. Vidomosti Verkhovnoi Rady Ukrainy vid 29.03.1994, 13 (1993) (in Ukrainian)
12. Chyzhova, A.A.: Algorithms of Finding Plagiarism. Skhidno-Yevropeiskyi zhurnal peredovykh tekhnolohii **46**, 13–16 (2010). http://journals.uran.ua/eejet/article/view/2921 (in Ukrainian)
13. Sharapova, E.V.: Universal system of text checking for plagiarism «Avtor.net». Informatika i Ejo Primenenija **6**, 52–58 (2012). (in Russian)
14. Shynkarenko, V.I.: The System for Control of Plagiarism in Student Papers. Skhidno-Yevropeiskyi zhurnal peredovykh tekhnolohii **4**, 32–36 (2012). http://journals.uran.ua/eejet/article/view/4214 (in Ukrainian)
15. Brokgauz, F.A., Efron, I.A.: Collegiate Dictionary, vol. 23, book 48, St. Petersburg (1898) (in Russian)
16. El Tahir, A.A.M.: Overview and comparison of plagiarism detection tools. In: CEUR Workshop Proceedings, vol. 706, pp. 161–172 (2011). http://ceur-ws.org/Vol-706/poster22.pdf
17. Anzelmi, D., Carlone, D., Rizzello, F., Thomsen, R., Akbar Hussain, D.M.: Plagiarism detection based on SCAM algorithm. In: Proceedings of the International MultiConference on Engineers and Computer Scientists 2011, vol. 1, pp. 272–277 (2011). http://www.ijsce.org/attachments/File/v2i5/E0984092512.pdf
18. Berlinck, R.G.S.: The academic plagiarism and its punishments - a review. Braz. J. Pharmacogn. **3**, 365–372 (2011)

19. Bull, J., Coughlin, E., Collins, C., Sharp, D.: Technical review of plagiarism detection software report, Luton (2000). http://www.jisc.ac.uk/uploaded_documents/luton.pdf
20. Hunes, C., Stiffler, J., Malsed, M.: Examining anti-plagiarism software: choosing the right tool, Clermont (2003). http://pdf.brand2.net/examining-anti-plagiarism-software-choosing-the-right-tool-w2556/
21. Hariharan, S.: Automatic plagiarism detection using similarity analysis. Int. Arab. J. Inf. Technol. **9**, 322–326 (2012). http://www.ccis2k.org/iajit/PDF/vol.9,no.4/2796-4.pdf
22. Kharat, R.: Semantically detecting plagiarism for research papers. Int. J. Eng. Res. Appl. **3**, 077–080 (2013). http://www.ijera.com/papers/Vol3_issue3/P33077080.pdf
23. Lancaster, T.: Effective and efficient plagiarism detection. Ph.D. thesis, London (2003)
24. Lancaster, T.: Classifications of Plagiarism Detection Engines. Innovation in Teaching and Learning in Information and Computer Sciences 5 (2005). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.184.2406&rep=rep1&type=pdf
25. Marcial, D.E., Fortich, M.S., Rendal, J.B.: ICT skills enhancement training in teacher education: the case in Central Visayas, Philippines. Inf. Technol. Learn. Tools **39**, 230–240 (2014). http://journal.iitta.gov.ua/index.php/itlt/article/view/964/749
26. Meyer zu Eissen, S.: intrinsic plagiarism detection: advances in information retrieve. In: Proceedings of 28th European Conference on IR Research (ECIR), pp. 565–569 (2006). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.5366&rep=rep1&type=pdf
27. Shenoy, M., Shet, K.C., Acharya, U.D.: Automatic plagiarism detection using similarity analysis: advanced computing. Int. J. **3**, 59–62 (2012). http://airccse.org/journal/acij/papers/0512acij06.pdf
28. Singh, R., Dutta, C.: Duplicity detection system for digital documents. Int. J. Soft Comput. Eng. **2**, 24–28 (2012). http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp272-277.pdf
29. Tschuggnall, M., Specht, G.: Detecting plagiarism in text documents through grammar-analysis of authors. In: 15th GI-Symposium Database Systems for Business, Technology and Web, pp. 241–259 (2013). http://www.btw-2013.de/proceedings/Detecting%20Plagiarism%20in%20Text%20Documents%20through%20GrammarAnalysis%20of%20Authors.pdf

Author Index