ISSN 1727–6209

TERNOPIL NATIONAL ECONOMIC UNIVERSITY
RESEARCH INSTITUTE OF INTELLIGENT COMPUTER SYSTEMS
IN COOPERATION WITH V.M. GLUSHKOV INSTITUTE FOR
CYBERNETICS, NATIONAL ACADEMY OF SCIENCES, UKRAINE

International Journal of

Computing

Published since 2002

March 2013, Volume 12, Issue 1

Ternopil
TNEU
2013

International Journal of Computing

March 2013, Vol. 12, Issue 1

**Research Institute of Intelligent Computer Systems,
Ternopil National Economic University**

**Registered by the Ministry of Justice of Ukraine.
Registration certificate of printed mass media – KB #17050-5820PR, 15.07.2010.**

It's published under resolution of the TNEU Scientific Council, protocol # 5, February 27, 2013

Editorial Board

EDITOR-IN-CHIEF

Anatoly Sachenko
Ternopil National Economic University, Ukraine

DEPUTY EDITOR-IN-CHIEF

Volodymyr Turchenko
University of Tennessee, USA

ASSOCIATE EDITORS

Svetlana Antoshchuk
Odessa National Polytechnic University, Ukraine

Plamenka Borovska
Technical University of Sofia, Bulgaria

Kent A. Chamberlin
University of New Hampshire, USA

Dominique Dallet
University of Bordeaux, France

Pasquale Daponte
University of Sannio, Italy

Mykola Dyvak
Ternopil National Economic University, Ukraine

Richard J. Duro
University of La Coruña, Spain

Vladimir Golovko
Brest State Polytechnical University, Belarus

Sergei Gorlatch
University of Muenster, Germany

Lucio Grandinetti
University of Calabria, Italy

Domenico Grimaldi
University of Calabria, Italy

Uwe Großmann
Dortmund University of Applied Sciences and Arts, Germany

Halit Eren
Curtin University of Technology, Australia

Vladimir Haasz
Czech Technical University, Czech Republic

Robert Hiromoto
University of Idaho, USA

Orest Ivakhiv
Lviv Polytechnic National University, Ukraine

Zdravko Karakehayov
Technical University of Sofia, Bulgaria

Mykola Karpinskyy
University of Bielsko-Biała, Poland

Volodymyr Kochan
Ternopil National Economic University, Ukraine

Yury Kolokolov
UGRA State University, Russia

Gennady Krivoula
Kharkiv State Technical University of Radioelectronics, Ukraine

Theodore Laopoulos
Thessaloniki Aristotle Univerity, Greece

Fernando López Peña
University of La Coruña, Spain

Kurosh Madani
Paris XII University, France

George Markowsky
University of Maine, USA

Richard Messner
University of New Hampshire, USA

Yaroslav Nykolaiychuk
Ternopil National Economic University, Ukraine

Vladimir Oleshchuk
University of Agder, Norway

Oleksandr Palahin
V.M.Glushkov Institute of Cybernetics, Ukraine

José Miguel Costa Dias Pereira
Polytechnic Institute of Setúbal, Portugal

Dana Petcu
Western University of Timisoara, Romania

Vincenzo Piuri
University of Milan, Italy

Oksana Pomorova
Khmelnytsky National University, Ukraine

Peter Reusch
University of Applied Sciences, Germany

Sergey Rippa
National University of State Tax Service of Ukraine, Ukraine

Volodymyr Romanov
V.M. Glushkov Institute of Cybernetics, Kiev, Ukraine

Andrzej Rucinski
University of New Hampshire, USA

Bohdan Rusyn
Physical and Mechanical Institute of Ukrainian NASU, Ukraine

Rauf Sadykhov
Byelorussian State University of Informatics and Radioelectronics, Belarus

Jürgen Sieck
HTW – University of Applied Sciences Berlin, Germany

Axel Sikora
University of Applied Sciences Offenburg, Germany

Rimvydas Simutis
Kaunas University of Technology, Lithuania

Tarek M. Sobh
University of Bridgeport, USA

Volodymyr Tarasenko
National Technical University of Ukraine "Kyiv Polytechnics Institute", Ukraine

Wiesław Winiecki
Warsaw University of Technology, Poland

Janusz Zalewski
Florida Gulf Coast University, USA

Address of the Editorial Board

Research Institute of Intelligent Computer Systems
Ternopil National Economic University
3, Peremoga Square
Ternopil, 46020, Ukraine

Phone: +380 (352) 47-5050 ext. 12234
Fax: +380 (352) 47-5053 (24 hours)
computing@computingonline.net
www.computingonline.net

© International Journal of Computing, 2013

ISSN 1727–6209

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
НАУКОВО-ДОСЛІДНИЙ ІНСТИТУТ ІНТЕЛЕКТУАЛЬНИХ
КОМП'ЮТЕРНИХ СИСТЕМ
У СПІВПРАЦІ З ІНСТИТУТОМ КІБЕРНЕТИКИ ІМ. В.М. ГЛУШКОВА,
НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ

Міжнародний науково-технічний журнал

Комп'ютинг

Видається з 2002 року

Березень 2013, Том 12, Випуск 1

Постановою президії ВАК України № 1-05/3 від 14 квітня 2010 року науково-технічний журнал "Комп'ютинг" віднесено до переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата технічних наук

Тернопіль
ТНЕУ
2013

Міжнародний журнал “Комп’ютинг”

Березень 2013, том 12, випуск 1

Науково-дослідний інститут інтелектуальних комп’ютерних систем
Тернопільський національний економічний університет

Зареєстрований Міністерством юстиції України. Свідоцтво про державну реєстрацію друкованого засобу масової інформації – серія КВ №17050-5820ПР від 15.07.2010.

Друкується за постановою вченої ради ТНЕУ, протокол № 5 від 27 лютого 2013 року

Редакційна колегія

ГОЛОВНИЙ РЕДАКТОР

Анатолій Саченко
Тернопільський національний економічний університет, Україна

ЗАСТУПНИК РЕДАКТОРА

Volodymyr Turchenko
University of Tennessee, USA

ЧЛЕНИ РЕДКОЛЕГІЇ

Світлана Антощук
Одеський національний політехнічний університет, Україна

Микола Дивак
Тернопільський національний економічний університет, Україна

Орест Івахів
Національний університет “Львівська політехніка”, Україна

Володимир Кочан
Тернопільський національний економічний університет, Україна

Геннадій Кривуля
Харківський державний технічний університет радіоелектроніки, Україна

Ярослав Николайчук
Тернопільський національний економічний університет, Україна

Олександр Палагін
Інститут кібернетики ім. В.М. Глушкова НАНУ, Україна

Оксана Поморова
Хмельницький національний університет, Україна

Сергій Ріппа
Національний університет державної податкової служби України, Україна

Володимир Романов
Інститут кібернетики ім. В.М.Глушкова НАНУ, Україна

Богдан Русин
Фізико-механічний інститут НАНУ, Україна

Володимир Тарасенко
Національний технічний університет України “Київський політехнічний інститут”, Україна

Plamenka Borovska
Technical University of Sofia, Bulgaria

Kent A. Chamberlin
University of New Hampshire, USA

Dominique Dallet
University of Bordeaux, France

Pasquale Daponte
University of Sannio, Italy

Richard J. Duro
University of La Coruña, Spain

Vladimir Golovko
Brest State Polytechnical University, Belarus

Sergei Gorlatch
University of Muenster, Germany

Lucio Grandinetti
University of Calabria, Italy

Domenico Grimaldi
University of Calabria, Italy

Uwe Großmann
Dortmund University of Applied Sciences and Arts, Germany

Halit Eren
Curtin University of Technology, Australia

Vladimir Haasz
Czech Technical University, Czech Republic

Robert Hiromoto
University of Idaho, USA

Zdravko Karakehayov
Technical University of Sofia, Bulgaria

Mykola Karpinskyy
University of Bielsko-Biała, Poland

Yury Kolokolov
UGRA State University, Russia

Theodore Laopoulos
Thessaloniki Aristotle Univerity, Greece

Fernando López Peña
University of La Coruña, Spain

Kurosh Madani
Paris XII University, France

George Markowsky
University of Maine, USA

Richard Messner
University of New Hampshire, USA

Vladimir Oleshchuk
University of Agder, Norway

José Miguel Costa Dias Pereira
Polytechnic Institute of Setúbal, Portugal

Dana Petcu
Western University of Timisoara, Romania

Vincenzo Piuri
University of Milan, Italy

Peter Reusch
University of Applied Sciences, Germany

Andrzej Rucinski
University of New Hampshire, USA

Rauf Sadykhov
Byelorussian State University of Informatics and Radioelectronics, Belarus

Jürgen Sieck
HTW – University of Applied Sciences Berlin, Germany

Axel Sikora
University of Applied Sciences Offenburg, Germany

Rimvydas Simutis
Kaunas University of Technology, Lithuania

Tarek M. Sobh
University of Bridgeport, USA

Wiesław Winiecki
Warsaw University of Technology, Poland

Janusz Zalewski
Florida Gulf Coast University, USA

Адреса редакції :

НДІ інтелектуальних комп’ютерних систем
Тернопільський національний економічний університет
площа Перемоги, 3
Тернопіль, 46020, Україна

Тел.: 0 (352) 47-50-50 внутр. 12234
Факс: 0 (352) 47-50-53
computing@computingonline.net
www.computingonline.net

© Міжнародний науково-технічний журнал “Комп’ютинг”, 2013

CONTENTS

N. Anantalapochai, A. Sikora
Integration of BACNET OPC UA-Devices Using a JAVA OPC UA SDK
Server with BACNET Open Source Library Implementation 7

A.J. Kornecki, S.T. Wierzchon, J. Zalewski
Reasoning Under Uncertainty with Bayesian Belief Networks Enhanced with Rough Sets 15

V. Deibuk, I. Grytsku
Optimal Synthesis of Reversible Quantum Summators Using Genetic Algorithm 31

V.G. Red'ko
Interaction Between Learning and Evolution in Populations of Autonomous Agents 41

V. Chernega
Performance of a Transport Level of WLANS IEEE 802.11g
Functioning in Infrastructural Mode 47

V. Haasz, D. Slepicka, P. Suchanek
Post-Correction of ADC Non-Linearity Using Integral Non-Linearity Curve 55

D.V. Komashinskiy, I.V. Kotenko
Intelligent Data Analysis for Malware Detection 62

M. Knemeyer, M. Nsaif, F. Glinka, A. Ploss, S. Gorlatch
Towards Data Persistency in Real-Time Online Interactive Applications 74

I.V. Kotenko, P.G. Nesteruk, A.V. Shorov
Conception of a Hybrid Adaptive Protection of Information Systems 85

Abstracts 98

Information for Papers Submission to Journal 104

CONTENTS / ЗМІСТ / СОДЕРЖАНИЕ

N. Anantalapochai, A. Sikora
Integration of BACNET OPC UA-Devices Using a JAVA OPC UA SDK
Server with BACNET Open Source Library Implementation 7

A.J. Kornecki, S.T. Wierzchon, J. Zalewski
Reasoning Under Uncertainty with Bayesian Belief Networks Enhanced with Rough Sets 15

В. Дейбук, І. Грицку
Оптимальний синтез зворотних квантових суматорів з допомогою генетичних алгоритмів 31

V.G. Red'ko
Interaction Between Learning and Evolution in Populations of Autonomous Agents 41

В. Чернега
Пропускна Спроможність Транспортного Рівня Безпровідних Локальних Мереж
IEEE 802.11g, що функціонують в інфраструктурному режимі 47

V. Haasz, D. Slepicka, P. Suchanek
Post-Correction of ADC Non-Linearity Using Integral Non-Lineatity Curve 55

Д. Комашинский, И. Котенко
Интеллектуальный анализ данных для выявления вредоносных программ 62

M. Knemeyer, M. Nsaif, F. Glinka, A. Ploss, S. Gorlatch
Towards Data Persistency in Real-Time Online Interactive Applications 74

И. Котенко, Ф. Нестерук, А. Шоров
Концепция гибридной адаптивной защиты информационных систем 85

Abstracts / Резюме 98

Information for Papers Submission to Journal / Інформація для оформлення
статей до журналу / Информация для оформления статей в журнал 104

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

INTEGRATION OF BACNET OPC UA-DEVICES USING A JAVA OPC UA SDK SERVER WITH BACNET OPEN SOURCE LIBRARY IMPLEMENTATION

Naksit Anantalapochai, Axel Sikora

Hochschule Offenburg, Badstrasse 24, D77652 Offenburg, Germany
nanantal@stud.hs-offenburg.de, axel.sikora@hs-offenburg.de

Abstract: *The variety of technologies used in modern Building Automation Systems (BAS) calls for methods to support interoperability of the devices from different technologies and vendors. OLE for Process Control Unified Architecture (OPC UA) provides the possibility to enable secure interoperability of devices with platform independence and efficient information model features. However, OPC has not found broad space in the world of building automation, yet.*

In this paper, results and experiences from a project are presented, where BACnet devices were implemented with OPC UA standard models. The values and controls are presented by the OPC UA server running on an embedded device. In the method, we map the BACnet information models into the corresponding OPC UA information models. The information model (in OPC UA form) of the BACnet devices can be accessed by connecting the OPC UA Clients to the OPC UA Server. This objective should be pursued by using as many available open-source projects as possible.

Keywords: *BACnet, OPC Unified Architecture, Building Automation Systems, OPC UA Server*

1. INTRODUCTION

One of the major challenges for developers and integrators of modern Building Automation Systems (BAS) is the integration of very different technologies and of devices from numerous vendors. Interoperability of these devices is required to integrate all devices and all information into the same interworking model and into one server for improved controllability and observability, but for the reduction of cost, installation efforts, reliability, and system complexity. OLE for Process Control Unified Architecture (OPC UA) can be used to present a generic view to the monitoring clients the need access to the entire Building Automation System. Using Java as a programming language as a SDK for the OPC UA Server is a promising stepping stone for the platform independence.

The creation of OPC information model for the BACnet types and objects is generally the most significant task in this process, and thus also of this project. This paper describes one of the basic approaches of BACnet-to-OPC UA integration and sample implementation. Furthermore, as the server will be running constantly (always on), energy consumption of the control system must also be taken into account. Therefore, the presented project uses a programmable embedded device for the OPC server, which connects to the BACnet devices and the client for monitoring in the same local area network.

This paper is structured as follows: After the description of some parallel activities in ch. 2, a brief overview about the OPC UA Standards and the BACnet Protocol is given in chapters 3 and 4. The following ch. 5 describes the method to setup and prepare the devices and working environment. Ch. 6 elaborates the use of the “BACnet4J” Java library to access BACnet services, before ch. 7 describes the setup of Java OPC UA Server and Client, as well as the monitoring the values of the devices. Finally, ch. 8 reports on the implementation and its results.

2. STATE OF THE ART

To the best knowledge of the authors, the most up-to-date procedure to create an OPC information model for BACnet information was presented in [1] and [2], which cover the most important BACnet information (object and property) types, i.e. the mandatory types.

With this initial description it gives an inspiration to research for the optimized way for industrial operation and commissioning. As stated in the introduction section that a programmable embedded device is used for the operation to satisfy these requirements. Due to the restrictions in memory and processing power, this project implements the

integration in a simplified way, using selected concepts and ideas from [1] and [2].

3. OPC UA

3.1 OVERVIEW

Object Linking and Embedding (OLE) for Process Control or OLE for Process Control (OPC) was developed in 1996 by the *OPC Foundation*, an association of worldwide industrial automation suppliers working on cooperation with Microsoft. Now, OPC is an open standard specification that describes the communication of real-time plant data between control devices from different manufacturers.

The origin of OPC was based on OLE, Component Object Model (COM) and Distributed Component Object Model (DCOM) technologies developed by Microsoft for Windows operating systems. The very first standard from *OPC Foundation* was for Data Access Specification or OPC Data Access (OPC DA). Soon after OPC DA was launched, it was realized that communicating other types of data could benefit from standardization. Thus, the later standards for Alarm and Events (A&E), Historical Data Access (HDA), Data Exchange and Batch Data were published.

The fact that legacy OPC standards are based on Microsoft's COM/DCOM paradigm later turned out to be a limitation to several operations especially for interoperability. Besides, with the insufficiency of object oriented concept it was impractical to model complex data structures.

Because of these restrictions, a new standard was needed. Eventually, OPC Unified Architecture (OPC UA) was released, which extends the OPC communication protocol and enables data acquisition, information modeling, reliable and secure communication between the plant floor and the enterprise resource planning level. It was intended to be a full replacement of the classic OPC specifications. The key features and benefits of OPC UA are:

- platform independence to run on any operating system including embedded devices
- single set of services to expose OPC data models (DA, A&E, HDA, Batch, Data eXchange, etc.)
- reliable, secure and efficient way to transport higher level structured data
- broader scope of connectivity
- extensible for future applications and modifications
- backward compatibility

The overall OPC UA specifications consist of 13 specification parts shown in Figure 1 [3].

It is important to understand how to model the address spaces and information, which is described in Part 3 "Address Space Model" and Part 5 "Information Model".

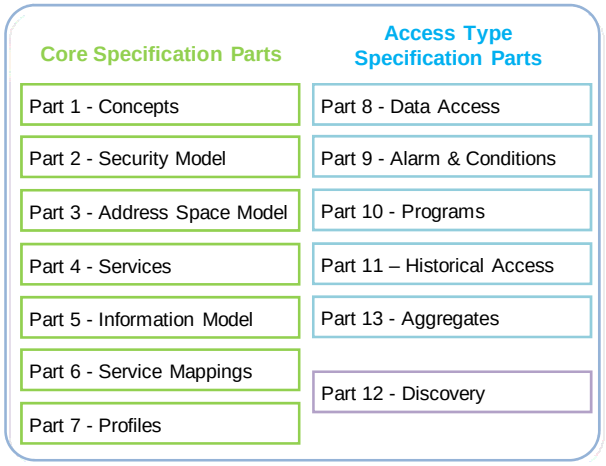


Fig. 1 – Parts of the OPC UA Specifications [3]

These two specifications are the key documents for the design and development of OPC UA server to integrate with other standards and protocols. Before describing the information and address space modeling in OPC UA, some basic infrastructures (described in specifications Part 1 "Concept") should be introduced first.

Common OPC UA Client and Server according the specifications both consist of the two parts UA application and Communication Stack (cf. Fig. 2).

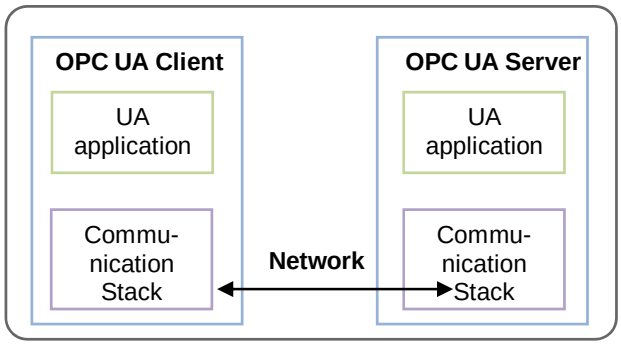


Fig. 2 – Common OPC UA Client and Server [3]

In the OPC UA Server, the application part contains the OPC UA address space representing the UA complex information model as a node. Figure 3 illustrates the complete architecture of OPC UA Server.

3.2 INFORMATION AND ADDRESS SPACE MODELING

As the nodes represent the UA information, it is necessary to understand the basic concept of information modeling in OPC UA. To enable interoperability between devices from different

vendors, a uniform representation of data is required, which is called information model in OPC UA. The devices from any vendor can use or even inherit (with regards to the concept of object orientation) the model for their own usage. However before having an information model, first a place to store this information model must be instantiated.

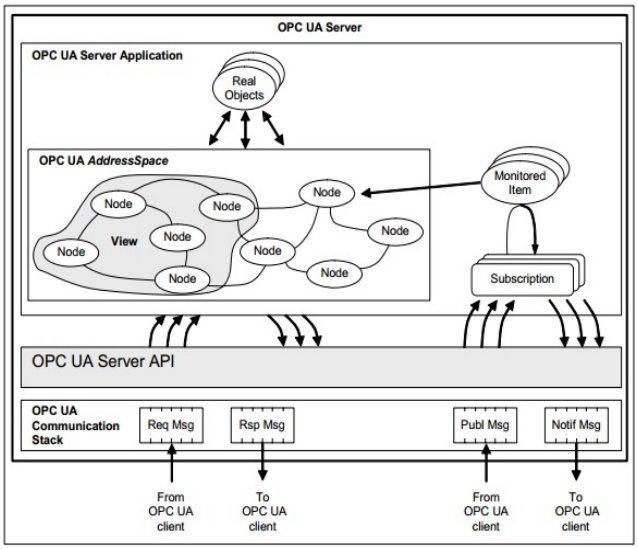


Fig. 3 – OPC UA Server Architecture [3]

Every information model is represented in the address space as a set of nodes and interconnected by references in the hierarchical form (branches of a tree), where the highest node is called 'Root' node. Figure 4 describes the node model and the details of the address space node model.

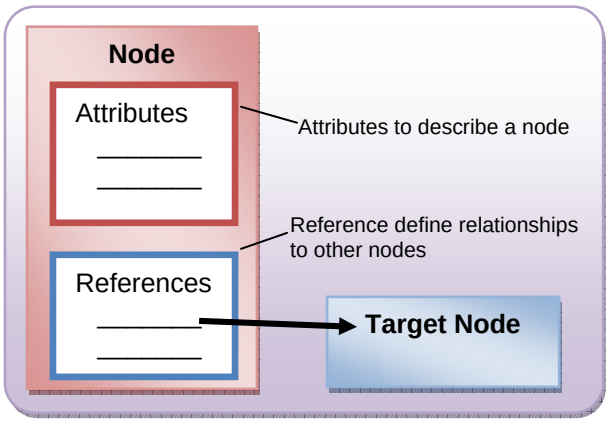


Fig. 4 – OPC UA Address Space Node Model [3]

When a node is instantiated in the address space, it is defined by 'Node Class'. These node classes are referred to collectively as the metadata for the address space. The node classes can be classified to types and instances as stated in Table 1.

Table 1. OPC UA built-in definition node classes [3]

OPC UA built-in type definition node classes	
Data Type Node Class	Describes the structure of the Value Attribute of Variables and their Variable Types
Variable Type Node Class	Provides type definitions for Variables
Object Type Node Class	Provides type definition for Objects
Reference Type Node Class	Specifies References
OPC UA built-in instance definition node classes	
Object Node Class	Represents systems, system components, real-world objects and software objects. Defined by Object Type
Variable Node Class	Represent Values which may be simple or complex. Defined by Variable Type
Method Node Class	Defines callable functions. Invoked using Call Service defined in Part 4 of OPC UA Specification
View Node Class	Defines a subset of the Nodes in the Address Space in order to limit the node visibility for some specific purpose. All Nodes contained in the View shall be accessible from the View node when browsing

The OPC UA built-in '*DataTypes*' are all inherited from the highest parent node of *BaseDataType* and can be used directly for this work i.e. *Boolean*, *Integer(Number)* and *Float(Number)*. The same holds true for '*VariableTypes*' and '*ObjectTypes*', they are inherited from *BaseVariableType* and *BaseObjectTypes*, respectively. For the 'ReferenceTypes', there are two highest parent nodes *HierarchicalReferences* and *NonHierarchicalReferences*.

4. BACNET

4.1 OVERVIEW

In June of 1987, BACnet was introduced as a communication protocol for building automation and control networks by American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE, http://www.ashrae.org). BACnet was designed to allow communication of building automation and control systems for applications i.e. heating system control, air control, lighting control, access control, fire detection and any other control systems related to building automation. The BACnet protocol provides services for programmable/electronic building automation devices to exchange data.

The BACnet protocol has been frequently updated every year. The last update has been made in October 2012. Therefore, it is recommended to regularly check for updates from the ASHRAE SSPC 135 BACnet official website [4].

The BACnet layered architecture based on reduced ISO-OSI Reference Model [6] is shown in Figure 5 [5].

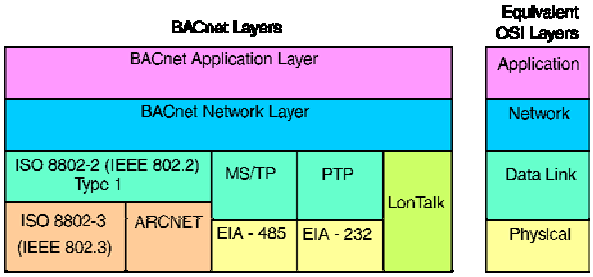


Fig. 5 – BACnet OSI-based Reduced Architecture [5]

The Application Layer is the most relevant layer to this project, this layer defines the BACnet information model and the BACnet services which will be used to firstly poll the data from the BACnet device and then to be processed in its information model before the integration to OPC UA.

4.2 BACNET: OBJECTS AND SERVICES IN APPLICATION LAYER

The application Layer contains information data often refers to objects and their properties in term of BACnet protocol) and its services used to exchange information through the below layers.

BACnet object has its type to distinguish the kind of the object. Up to date, there are already 50 object types defined within the standard. Within this project, five different object types are implemented, i.e. Analog Input, Analog Output, Binary Input, Binary Output and Device. An example of the Analog Input object, which is commonly used, is shown in Fig. 6.

Object

Object_Type:	ANALOG_INPUT
Object_Name:	Temperature_1
Object_Identifier:	ANALOG_INPUT1
Present_Value:	23.1
Status_Flags:	In_Alarm
Event_State:	Normal, Out-of-Service
Units:	Degrees-Celsius
Out_Of_Service:	False

Fig. 6 – Example of BACnet Analog Input Object

This example of BACnet Analog Input object shows the object characterized by its mandatory properties. Different object types will have different mandatory properties together with the possibility to add optional properties.

From [7], in order to exchange the information, BACnet services are classified into four different categories: Alarm&Event, File Access, Object Access and Remote Device Management. With regards to the project, only services from Object Access and Remote Device Management services are utilized.

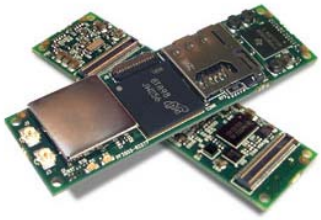
Remote Device Management services are used in the step of discovering BACnet devices in the network. Generally, it is to broadcast a “Who-Is” request to the network and wait for an “I-Am” reply from the BACnet devices if they are accessible in the network.

Object Access services are used to access the objects of the device and to read or write the values of its properties. Examples of Object Access services are ‘ReadProperty’, ‘WriteProperty’, ‘ReadPropertyMultiple’ and ‘WritePropertyMultiple’ services.

5. DEVICES AND SYSTEMS PREPARATION

5.1 OPC UA SERVER

As stated in the introduction section, an embedded device is used for the OPC server. This project takes Overo Air Computer-on-Module (COM) with Tobi expansion board from Gumstix into operation. A brief overview of the device and its expansion can be found in the Table 2 below. The specification sheet of the devices can be accessed from the Gumstix official website in the product section for more details [8] [9].



a) Gumstix Overo Air COM

b) Gumstix Tobi Expansion Board

Fig. 7 – Devices for running OPC UA Server [8] [9]

Table 2. Device Overview

Gumstix Overo Air Computer-on-Module	
Architecture	ARM Cortex-A8
Processor	Texas Instruments OMAP3503 Applications Processor
Address	10, Times New Roman, Regular
CPU Speed	600 MHz
RAM	512 MB
NAND Flash	512 MB
Performance	up to 1,200 Dhrystone MIPS
Power Mgmt	Texas Instruments TPS65950
Gumstix Tobi Expansion Board	
Network Socket	10/100BaseT Ethernet
DVI-D (HDMI)	HDMI
USB Slots	USB Host USB On-the-Go (OTG) USB Console
Power	3.5V – 5V

The Gumstix Overo Air COM is mounted on Gumstix Tobi Expansion Board for the primary purpose to connect it to the Ethernet network and to the power supply. The second purpose is to be able to locally control and monitor the functionalities of the OPC UA Server (e.g. mouse/keyboard connection, display on monitor screen).

Ubuntu 10.04 Lucid Desktop-Lite is installed as operating system due to its capability to have additional programs and easy configuration in the future. Nevertheless, other operating systems supported by Gumstix, like Angstrom, Android, Debian, or Windows CE would also be feasible. Since the server is Java based, a Java Runtime Environment (JRE) is required in the system.

5.2 OPC UA CLIENT

Although the OPC UA Client is not the main focus of the project, it is indispensable for monitoring the result of the approach and in the real operation.

A simple PC or mobile device which can connect to the OPC Server as a Client in the network is the only requirement. It is also possible to have an OPC UA Client in the embedded device which runs the OPC UA Server itself and to communicate over local loopback.

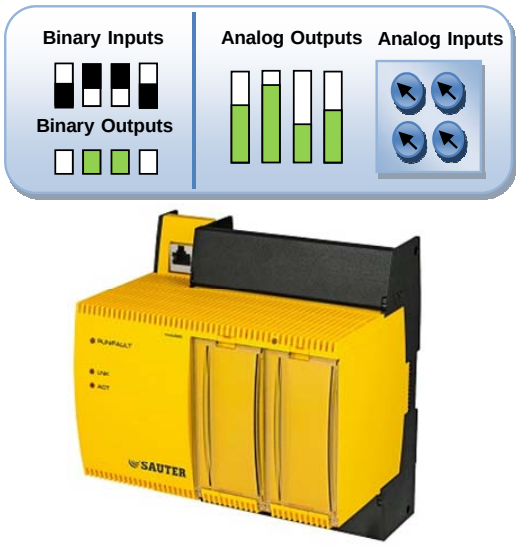


Fig. 8 – Basic Concept of BACnet Automation Station and Sauter EY-modulo 5 device for Demonstration [10]

5.3 BACNET DEVICE

At least one BACnet device is required in the network in order to provide the OPC UA server the information with regards to BACnet protocol. The project focuses on the 'Present_Values' information of Analog Input/Analog Output/Binary Input/Binary Output due to common usage in industry.

A BACnet automation station simulator kit which consists of all Analog Input/Analog Output/Binary Input/Binary Output is used in the project. Figure 8 shows the basic concept of BACnet automation station which is taken into operation.

6. BACNET4J LIBRARY

6.1 ABOUT THE LIBRARY

BACnet4J stands for BACnet/IP for Java. It is an open source library of the BACnet protocol. As described by the project team of this library from the BACnet4J developer website, "it is a high performance implementation of the BACnet/IP protocol written in Java (minimum version 1.5) by Serotonin Software, supports all BACnet Services and full message segregation and can be used for field devices for control platforms" [11].

6.2 FUNCTIONALITIES

The BACnet4J library enables the user to program an application communicating via the BACnet protocol using BACnet services. Most of the BACnet objects and properties (for example Analog Input, Analog Output, Binary Input, Binary Output and Device in this project) can be simply defined in Java object. The BACnet services can also be created by a Java object, which represents all services and calls the methods from the library to

execute the services.

With an additional programming on OPC UA server side (in Java) the server is able to retrieve BACnet information from the BACnet devices.

7. JAVA BASED OPC UA SERVER SDK

Java is the preferred language for the development of OPC UA server applications due to its platform independence especially on embedded devices. In addition, an open source library to communicate in BACnet (BACnet4J) already exists. Of course, other programming languages could also be used.

Prosys PMS Ltd. [12] is a contributor to the OPC Foundation UA Java stack and developed an OPC UA Java SDK. This comes with both OPC UA server SDK and OPC UA client SDK. It is now available in full release to purchase. Nevertheless, an evaluation is available. This project used the evaluation versions of this OPC UA server SDK.

The Prosys OPC UA server SDK supports the creation of an OPC UA address space and information model in the server which is mainly required for the integration of the BACnet information (objects and properties) to the OPC UA nodes and variables.

8. BACNET-to-OPC UA INTEGRATION

The first step in this integration process is to have all of the devices and systems in the same network domain, where all are accessible at least up to the network layer. For demonstration purposes, a local private network has been created and the devices are connected as illustrated in Figure 9.

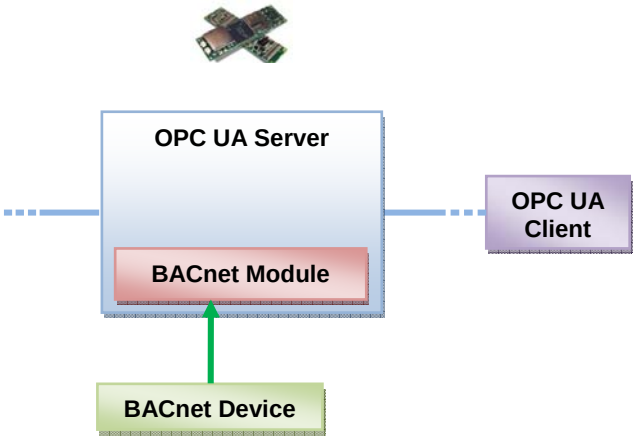


Fig. 9 – Connections of the devices

The BACnet Module inside the OPC UA server is the Java program that perform BACnet services (Remote Device Management and Object Access services) and stores the BACnet information in a Java object (BACnet4J functionality).

Based on the work of [1] and [2], OPC UA node types in the list must be prepared for the BACnet information:

- DataType
- ReferenceType
- ObjectType
- VariableType

In our project, we focused on the 'Present_Values' (cf. ch. 5.3) plus the concept of making the OPC UA information model for BACnet information in the simplified way. Consequently, the procedures in this project are:

Starting with the 'DataType', OPC UA built-in data types can be used for BACnet 'Present_Values' directly without any problem. The data type definition of *Boolean* inherited from *BaseDataType* can be used for the BACnet 'Present_Values' of Binary Input and Binary Output and the data type definition of *Float* which is inherited from *BaseDataType»Number»Float* can be used for the BACnet 'Present_Values' of Analog Input and Analog Output.

For all references of the nodes for BACnet information, the reference type definition will be the OPC UA built-in *HasComponent* inherited *HierarchicalReferences »Aggregates»HasComponent.*

Some nodes of object type definition must be defined for BACnet objects as illustrated in Figure 10.

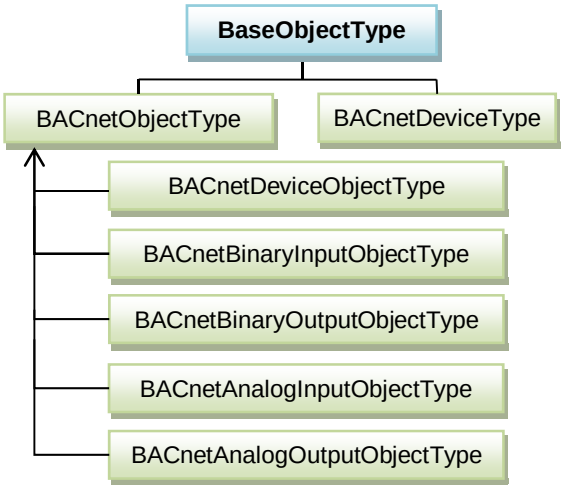


Fig. 10 – OPC UA Object Type Nodes for BACnet Objects

Every node of object type definition for BACnet objects (in green) inherits from the original OPC UA built-in '*BaseObjectType*' (in blue) like other OPC UA nodes of object type definition.

As well as ObjectType, some nodes of variable type definition must be defined for the BACnet properties. Figure 11 shows the node hierarchies for BACnet properties.

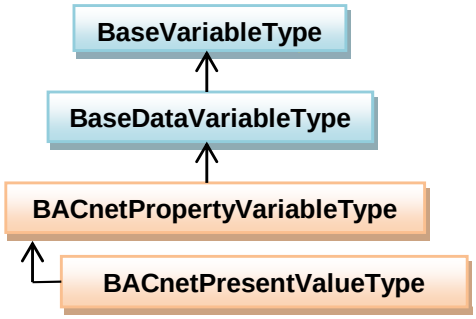


Fig. 11 – OPC UA Property Type Nodes for BACnet Properties

Every node of property type definition for BACnet properties (in orange) inherits from the original OPC UA built-in "*BaseDataVariableType*" and "*BaseVariableType*" (in blue). Other BACnet properties like "*Present_Value*" can be defined under the "*BACnetPropertyVariableType*", as well.

9. RESULTS

9.1 BACNET BROWSING RESULT

The results of browsing (discovery and reading properties) the BACnet demo device with the use of BACnet4J library and displayed in console is shown in Figure 12.

```
Initializing Local Device...
Discovering on port 47808 (BACx0)

Send Broadcast WhoIsRequest()

IAm received: RemoteDevice(instanceNumber=250
macAddress=[c0,a8,1,c,ba,c0]), network=null)

1 Remote Device(s) found

Start read properties
Properties read done in 49 ms
        - Device 2500
                Object name = DemoAS525
        - File 204
        - File 300
        - File 301
        - File 1000
        - File 1001
        - File 1002
        - File 1003
        - File 1004
        - Vendor Specific (384) 0
        - Vendor Specific (384) 1
        - Vendor Specific (384) 2
        - Vendor Specific (384) 3
        - Vendor Specific (384) 4
        - Vendor Specific (384) 5
        - Vendor Specific (384) 6
        - Vendor Specific (384) 7
        - Vendor Specific (384) 8
        - Program 1
        - Analog Output 0
                Present value = 1.0
        - Analog Output 1
                Present value = 1.0
        - Analog Input 16
                Present value = 41.70447
        - Analog Input 17
                Present value = 29.036024
        - Binary Input 4
                Present value = 0
        - Binary Input 14
                Present value = 0
        - Binary Input 15
                Present value = 0
        - Binary Output 20
                Present value = 0
        - Binary Output 21
                Present value = 1
        - Analog Output 2
                Present value = 0.0
        - Analog Output 3
                Present value = 0.0
        - Analog Input 18
                Present value = 56.686707
        - Binary Input 5
                Present value = 1
        - Binary Input 6
                Present value = 1
        - Binary Output 22
                Present value = 1
```

Fig. 12 – Results of Browsing BACnet Device

To monitor the BACnet service messages between the BACnet module and the device, Figure 13 displays the packet of BACnet services in Wireshark network sniffer.

192.168.1.10	192.168.1.255	BACnet-	50 Unconfirmed-REQ	who-Is
192.168.1.12	192.168.1.255	BACnet-	62 Unconfirmed-REQ	i-Am device,2500
192.168.1.10	192.168.1.255	BACnet-	62 Unconfirmed-REQ	i-Am device,1234
192.168.1.10	192.168.1.12	BACnet-	59 Confirmed-REQ	readProperty[0]
192.168.1.12	192.168.1.10	BACnet-	275 Complex-ACK	readProperty[0]
192.168.1.10	192.168.1.12	BACnet-	59 Confirmed-REQ	readProperty[0]
192.168.1.12	192.168.1.10	BACnet-	275 Complex-ACK	readProperty[0]
192.168.1.10	192.168.1.12	BACnet-	59 Confirmed-REQ	readProperty[0]
192.168.1.12	192.168.1.10	BACnet-	275 Complex-ACK	readProperty[0]
192.168.1.10	192.168.1.12	BACnet-	59 Confirmed-REQ	readProperty[0]
192.168.1.12	192.168.1.10	BACnet-	275 Complex-ACK	readProperty[0]

Fig. 13 – Packet Transfer Displayed by Wireshark

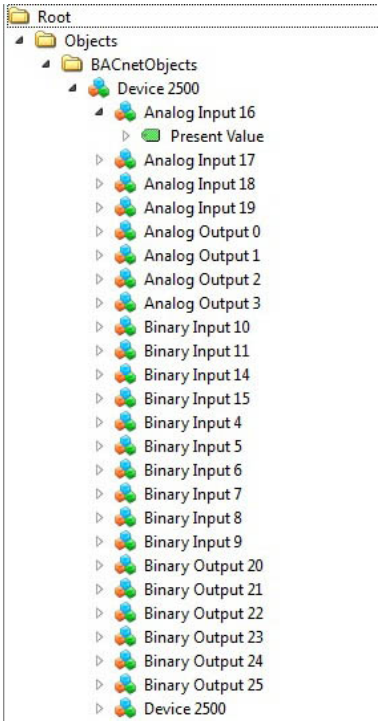


Fig. 14 – Result at OPC UA Client displayed by UaExpert

9.2 BACNET IN OPC UA MODEL

With the result of browsing BACnet information in subsection 9.1 and mapped in the created OPC UA information model for BACnet information by the procedure described in chapter 8. An additional folder named 'BACnetObjects' is also created to store the result in OPC UA model. This result is displayed in a freeware OPC UA client 'UaExpert' from Unified Automation while browsing to the OPC UA server [13] as shown in Figure 14.

With the OPC UA 'ObjectTypes' and 'VariableTypes' created for BACnet information, they can also be displayed in the OPC UA client as well in Figure 15 and Figure 16.

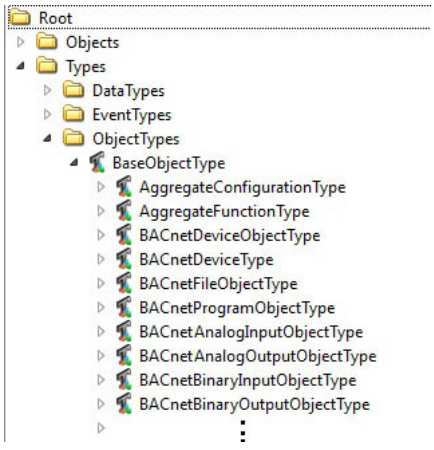


Fig. 15 – Result of OPC UA ObjectTypes for BACnet displayed by UaExpert

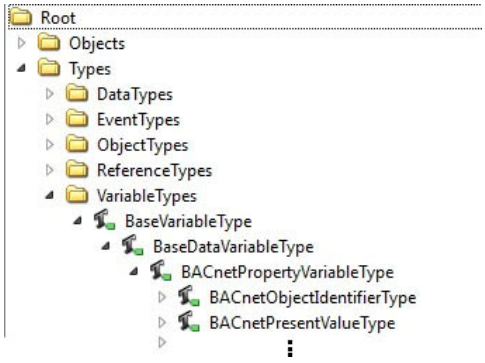


Fig. 16 – Result of OPC UA VariableTypes for BACnet displayed by UaExpert

10. CONCLUSION

Interoperability of the devices between different protocols allows significant benefits to the building automation industry. This paper shows an approach how to integrate BACnet devices to OPC UA, which can be adapted to many use cases in real work. However, the result of this work might not cover all of the BACnet mandatory properties but it is to show that the integration can also be done in the embedded device with the restriction in memory and processing power. Although most of the modern embedded devices are now much more vigorous and capable of this integration task without any problem but the old embedded devices are still in operation.

Not only BACnet information can be integrated to OPC UA information model since OPC UA built-in can support more complex data model so it is also promising and feasible to integrate other devices from other protocols especially the ones that have a concept of object oriented data model.

11. REFERENCES

[1] A. Fernbach, W. Granzer, W. Kastner, *Interoperability at the Management Level of Building Automation Systems: A Case Study for BACnet and OPC UA*, IEEE ETFA, 2011.

[2] W. Granzer, W. Kastner, Information Modeling in Heterogeneous Building Automation Systems, *IEEE International Workshop*, 2012.

[3] OPC Unified Architecture Specification 1.01 Part1: Concepts and Overview, Part 3: Address Space Model and Part 5: Information Model, *OPC Foundation*, 2009.

[4] ASHRAE SSPC 135 BACnet Official Website. http://www.bacnet.org/.

[5] S.T. Bushby, BACnet: a standard communication infrastructure for intelligent buildings, in: *Automation in Construction* 6, Elsevier, pp. 529-540.

[6] *Information Processing Systems – Open Systems Interconnection – Basic Reference Model*. ISO 7498, 1984.

[7] B. Swan, The Language of BACnet, in: *Engineered Systems*, (13) 7 (1996), pp. 24-32.

[8] Gumstix Overo AIR COM specifications Sheet, Gumstix Offical Website. https://www.gumstix.com/store/product_info.php?products_id=226.

[9] Gumstix Tobi specifications Sheet, Gumstix Offical Website. https://www.gumstix.com/store/product_info.php?products_id=230.

[10] Sauter EY-modulo 5 Product Data Sheet, Sauter AG Website. http://www.sauter-controls.com/pdm/docs/en_ds_en690613.pdf.

[11] BACnet/IP for Java Developer Website, Serotonin, 2008; http://bacnet4j.sourceforge.net/

[12] Prosys OPC Official Website http://www.prosysopc.com/

[13] Unified Automation Official Website http://www.unified-automation.com/

Naksit Anantalapochai*, born in Brussels Belgium in 1988, graduated Bachelor of Computer Engineering from Faculty of Engineering, Chiang Mai University, Thailand in 2011. Since then, he is studying Master degree of Science "Communication and Media engineering" from University of Applied Science Offenburg, Germany. He has a high interest in modern innovations and technologies especially in the field of embedded systems, communication networks and computer science altogether.*

Dr.-Ing. Axel Sikora *holds a diploma of Electrical Engineering and a diploma of Business Administration, both from Aachen Technical University. He has done a Ph.D. in Electrical Engineering at Fraunhofer Institute of Microelectronics Circuits and Systems, Duisburg. After various positions in the telecommunications and semiconductor industry, he became a professor at the Baden-Wuerttemberg Cooperative State University Loerrach in 1999. In 2011, he joined Offenburg University of Applied Sciences, where he holds the professorship of Embedded Systems and Communication Electronics.*

His major interest is in the system development of efficient, energy- aware, autonomous, secure, and value-added algorithms and protocols for wired and wireless embedded communication. He is founder and head of Steinbeis Transfer Center Embedded Design and Networking (sizedn).

Dr. Sikora is author, co-author, editor and co-editor of several textbooks and numerous papers, as well as conference committee member to various international conferences in the field of embedded design and wireless and wired networking.

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

REASONING UNDER UNCERTAINTY WITH BAYESIAN BELIEF NETWORKS ENHANCED WITH ROUGH SETS

Andrew J. Kornecki [1], Slawomir T. Wierzchon [2], Janusz Zalewski [3]

[1] Dept. of Electrical, Computer, Software and System Engineering, Embry Riddle Aeronautical University
Daytona Beach, FL 32114, USA
kornecka@erau.edu, http://faculty.erau.edu/korn/
[2] Faculty of Mathematics, Physics and Informatics, University of Gdańsk
Wita Stwosza 57, 80-952 Gdańsk-Oliwa, Poland
stw@ipipan.waw.pl, http://www.ipipan.waw.pl/~stw/
[3] Dept. of Software Engineering, Florida Gulf Coast University
Fort Myers, FL 33965, USA
zalewski@fgcu.edu, http://www.fgcu.edu/zalewski/

Abstract: *The objective of this paper is to present a new approach to reasoning under uncertainty, based on the use of Bayesian belief networks (BBN's) enhanced with rough sets. The role of rough sets is to provide additional reasoning to assist a BBN in the inference process, in cases of missing data or difficulties with assessing the values of related probabilities. The basic concepts of both theories, BBN's and rough sets, are briefly introduced, with examples showing how they have been traditionally used to reason under uncertainty. Two case studies from the authors' own research are discussed: one based on the evaluation of software tool quality for use in real-time safety-critical applications, and another based on assisting the decision maker in taking the right course of action, in real time, in the naval military exercise. The use of corresponding public domain software packages based on BBN's and rough sets is outlined, and their application for real-time reasoning in processes under uncertainty is presented.*

Keywords: *Bayesian Belief Networks, Rough Sets, Decision Uncertainty, Soft Computing.*

1. INTRODUCTION

Bayesian Belief Networks (BBN's) have been widely used in Industrial Information Systems for solving all types of computational problems with insufficient information and uncertainty [1,2]. This includes applications such as: water contamination [3], fault detection in an industrial process [4], fog forecasting at the airports [5], predicting software defects [6], inferring certification metrics of software [7], predicting hospital admissions for emergency [8], multisensor fusion for landmine detection [9], evaluation of risk in software development [10], modeling an air traffic control [11], cell signaling pathway modeling [12], reliability estimation [13], safety assessment [14] and risk evaluation [15] in computer-based systems, to name only a few from a long list. They have been also studied theoretically by a number of researchers, for example [16-17].

Although, in general, BBN's have been very effective, because they allow reasoning and making predictions based on small sets of probabilities with backwards inference, they are still based on probability theory. A significant disadvantage of BBN's is that, in realistic cases, they require extensive computations of the conditional probability values. In most of these studies, it has been recognized that this is one of the method's major limitations. Another disadvantage of BBN's is that they become less effective in case of missing probability values.

With this in mind, one wants to look at a complementary method of evaluating data in the input data set, which would not rely strictly on probability densities and could deal with missing values. One of the theories that offer such an approach, with values of data attributes and events measured by likelihoods rather than probabilities, is the rough sets theory [18-19].

Rough sets have been used since the early eighties [19], in a wide range of applications to reason about uncertainty, including data mining[20], medical diagnosis [21], robotic systems [22], decision making in medicine [23], cost estimation [24], modeling software processes [25-26], safety analysis [27], controller design [28], quality analysis

[29], fault diagnosis [30] and many others, for example prognostics [31].

The objective of this paper is to look at the combination of using BBN's and rough sets in decision making under uncertainty, and suggest the enhancement of pure Bayesian reasoning by additional use of rough sets for preliminary evaluation of data. The paper is structured as follows. The next two sections describe briefly basic concepts of Bayesian belief networks and rough sets, respectively. Then, in a separate section, several examples and two case studies are presented, giving an overview of the method developed for combining BBNs and rough sets for real-time computations. The final sections present general conclusions and suggestions for future work.

2. BAYESIAN BELIEF NETWORKS

The following section describes Bayesian Belief Networks from an application point of view rather than the underlying mathematics and statistics. Rev. Thomas Bayes developed this method of updating probabilities based on new information in the 1760s. It has been widely applied in probability and statistics for over 250 years.

"Essay Towards Solving a Problem in the Doctrine of Chances" was published after Bayes' death in 1763 [32]. It is the basis for the popular inversion formula for belief updating from evidence (E) about a hypothesis (H) using probability measurements of the prior truth of the statement updated by posterior evidence

$$P(H|E) = (P(E|H) * P(H)) / P(E)$$

where H is the hypothesis, E is the evidence, and P(x|y) is the conditional probability of x given y.

It is derived by the use of the joint probability definition

$$P(x, y) = P(x|y) * P(y) = P(y|x) * P(x)$$

that is then arranged as

$$P(x|y) = P(y|x) * P(x) / P(y)$$

where x = H and y = E.

Suppose we know from historical medical records that meningitis causes a stiff neck in 1 of 2 patients. We also know that 1 in 50,000 people have meningitis and that 1 in 20 people have a stiff neck. If you wake up with a stiff neck what is the probability that you have meningitis? How do you estimate it?

The hypothesis is that you have meningitis and the evidence is your stiff neck. Applying the Bayes formula yields:

P(E|H) = 1 / 2 = 0.5 or probability of a stiff neck when you have meningitis

P(H) = 1 / 50,000 = 0.00002 or probability of meningitis in the population;

P(E) = 1/ 20 = 0.05 or probability of a stiff neck in the population;

which yields in turn:

$$P(H|E) = (P(E|H) * P(H)) / P(E) = (0.5 * 0.00002) / 0.05 = 0.0002$$

which is the probability of having meningitis when you have a stiff neck. Much more complex models with multiple hypotheses and evidence sources can be constructed usually in a graph form relating cause to effect.

These belief networks are more recent concepts credited to Professor Judea Pearl with his construction and solution algorithms along with the work of many others [33].

A Bayesian belief network is a form of probabilistic graphical model. The belief network represents the joint probability distribution of a set of random variables with explicit independence assumptions described by a directed graph. In this research a Bayesian network is defined by a directed acyclic graph of nodes representing variables and arcs representing probabilistic dependency relations among the variables.

If there is an arc from node A to another node B, then variable B depends directly on variable A and A is called a parent of B. If the variable represented by a node has a known value then the node is said to be observed as an evidence node. A node can represent any kind of variable, be it a measured parameter, a latent variable or a hypothesis. Nodes are not restricted to representing random variables; this is what is "Bayesian" about a belief network.

In the following, an example is presented of three node networks that are structured as linear, converging, and diverging (Figure 1), with the use of Netica software program [34]. A different software package for Bayesian belief networks, named Hugin [35], is equally effective and simple to use.

The above examples are causal Bayesian networks where the directed arcs of the graph are interpreted as representing causal relations in some real domain with prior information. The directed arcs do not have to be interpreted as representing causal relations; however in practice knowledge about causal relations is very often used as a guide in drawing Bayesian network graphs, thus resulting in cause and effect Bayesian belief networks.

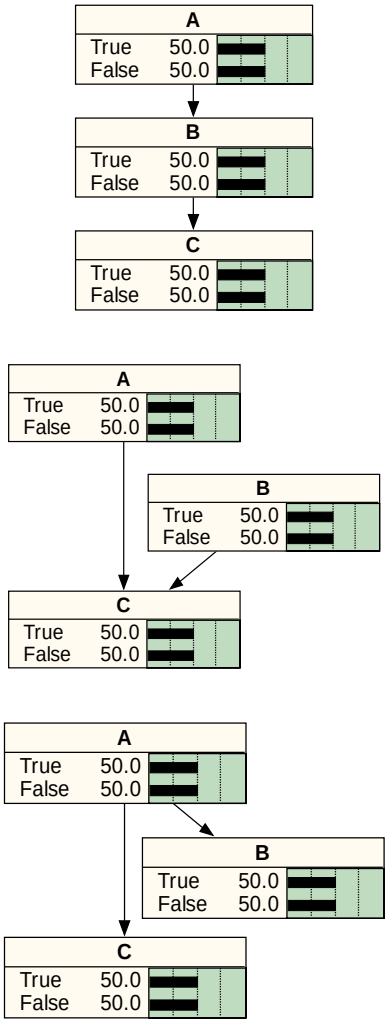


Fig. 1 – Examples of three-node Bayesian networks (top to bottom): linear, converging and diverging.

In the linear case on the top of Figure 2, A causes B that causes C, while in the converging model in the center, A is conditionally independent of B and both cause C, while in the diverging model at the bottom, A causes both B and C. In each case an effect is observed at node C illustrating the update of the joint probabilities when new information is incorporated into the network.

In the simplest case, a Bayesian network is specified by an expert and is then used to perform inference after some of the nodes are fixed to observed values. In order to fully specify the Bayesian network and fully represent the joint probability distribution, it is necessary to further specify for each node X the probability distribution for X conditional upon X's parents. The distribution of X conditional upon its parents may have many forms.

The following data (Table 1) are the conditional probabilities for the previous linear A, B, and C node network example (the top one) from Figure 2, where we observe that if C is true then column B is 0.8 true and 0.2 false. Prior to any observation as shown in Figure 1 the symmetry of the conditional probabilities makes the probability of true and false states equal to 0.5 for all nodes.

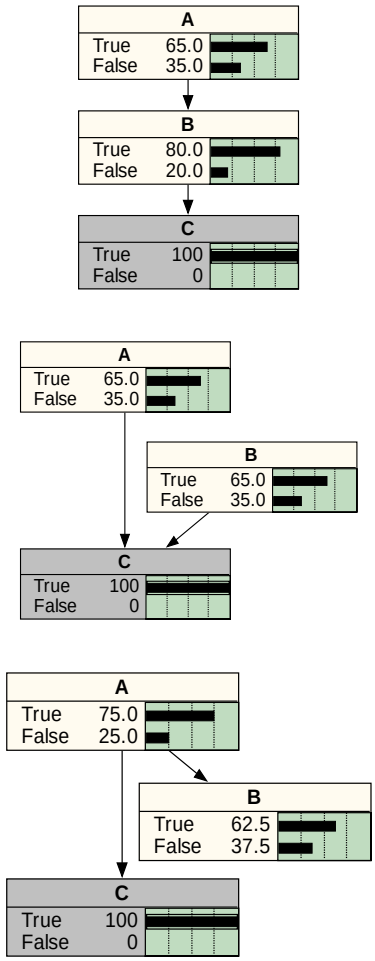


Fig. 2 – Causal relationships represented in a Bayesian belief network.

Table 1. Conditional probabilities for Figure 2

Node A		
True	False	
0.5	0.5	
Node B		
True	False	A
0.75	0.25	True
0.25	0.75	False
Node C		
True	False	B
0.80	0.20	True
0.20	0.80	False

The goal of inference is typically to find the distribution of a subset of the variables, conditional upon some other subset of variables with known values called the evidence or observations, with any remaining variables integrated out. This is known as

the posterior distribution of the subset of the variables given the evidence. The posterior gives us a universal sufficient statistic for detection applications, when one wants to choose values for the variable subset which minimize some expected loss function, for instance the probability of decision error.

An example of inference in the center converging example from Figure 2 is to specify A as observed true and estimate the inferred values for B and C, with C updated by the new information but B unchanged since it is conditionally independent of A.

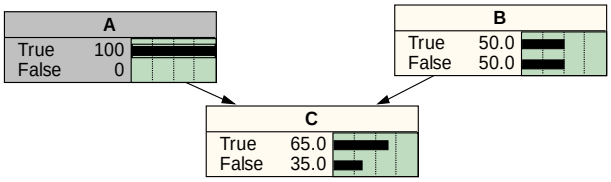


Fig. 3 – Effect of introducing evidence into a converging node in a Bayesian network.

In the divergence example from bottom of Figure 2, if A is observed then it infers new probability states for B and C that are dependent on A.

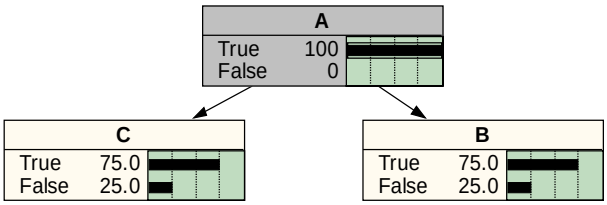


Fig. 4 – Effect of introducing evidence into a diverging node in a Bayesian network.

Questions about the dependence among variables can be answered by studying the graph alone. It can be shown that the conditional independence is represented in the graph by the graphical property of d-separation: nodes A and B are d-separated in the converging graph, given specified evidence nodes.

For belief reasoning a typical network is organized into three layers. The top layer is the causal variables, the middle layer is the reasoning variables, and the bottom layer is the effects variables. Four general classes of reasoning are defined for this three layer architecture. As an example the following five node network with three layers using binary random variables is used to illustrate the four principal reasoning strategies used in belief networks. The example network prior distribution has an equal probability for each variable state and is symmetric to illustrate the various conditional computations.

Diagnostic reasoning, illustrated in Figure 5, observes the effects of evidence and updates the middle reasoning variables and the top layer causal variables as shown in the example. This reasoning process diagnoses from an effect E of True to the cause B or in medical terms it is reasoning from symptom to disease. It also adjusts the probabilities for the middle reasoning layer C and the causal variable A and effect variable D.

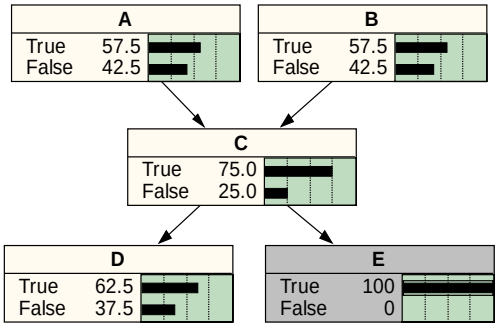


Fig. 5 – Illustration of diagnostic reasoning.

Predictive reasoning, illustrated in Figure 6, observes causal evidence and updates the middle reasoning variables and bottom layer effects variables as shown in the example. This reasoning process predicts from cause B to the effects D and E such as a patient saying that he is a smoker may focus on a certain set of symptoms. It also adjusts the probabilities for the middle reasoning layer C but not the independent causal variable A.

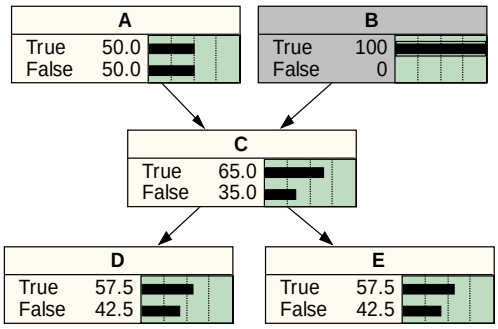


Fig. 6 – Illustration of predictive reasoning.

Intercausal or explaining reasoning, illustrated in Figure 7, observes both causal evidence and the middle layer reasoning evidence to update other causal variables as shown in the example. This reasoning process on C explains the mutual known cause B with unknown cause A and the common effects D and E. It is often interpreted as performing an experiment for explaining away cause A.

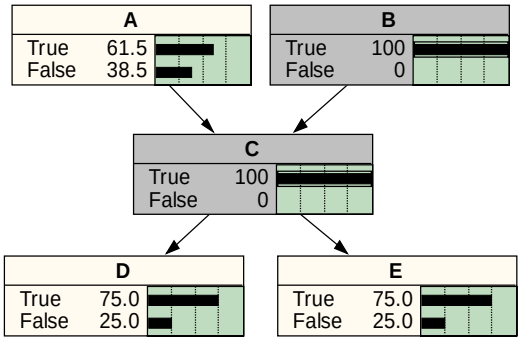


Fig. 7 – Illustration of intercausal (explaining) reasoning.

Combined reasoning, shown in Figure 8, observes causal evidence and effects evidence to update the middle reasoning variables as shown in the example. This reasoning process combines cause B and effect E to investigate the network conditional structure for reasoning variable C and the other cause A and the other effect D. This is a useful reasoning test for building and validating complex belief networks based on limited data and expert knowledge.

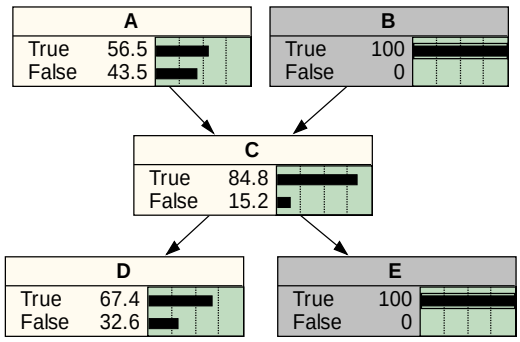


Fig. 8 – Illustration of combined reasoning.

3. ROUGH SETS

Rough Set Theory was invented by Zdzisław Pawlak to cope with limited perception of the surrounding world. The theory is especially helpful in dealing with vagueness and uncertainty in decision situations. Its main purpose is the "automated transformation of data into knowledge" [18]. The data are perceived in terms of objects and their features, i.e., values of the attributes used to characterize these objects. The knowledge deduced from these data is expressed in terms of *surely* and *possibly* statements describing notions of interests. More formally, such descriptions can be divided into so-called lower and upper approximations of entire notions. In the rest of this section, we describe a qualitative procedure containing all steps needed to form appropriate description of the concepts under considerations.

3.1 EXPLANATION OF A NOTION OF A ROUGH SET

First, let us illustrate intuitively a concept of a rough set, comparing it to an ordinary set and a fuzzy set, in a single dimension. Figure 9 shows such an intuitive illustration. For an *ordinary set*, the interval [*A,B*] in Figure 9, all its elements, that is, real numbers from this interval (assuming *x* represents a real axis), have values of their membership function equal to 1.0.

For a *fuzzy set*, elements on the set boundaries, that is, in the intervals [*A,C*] and [*D,B*], have values of their membership function equal to a fraction, a number from the interval [*0.0, 1.0*]. This means that these elements only partially belong to the set, to the extent specified by the value of a membership function.

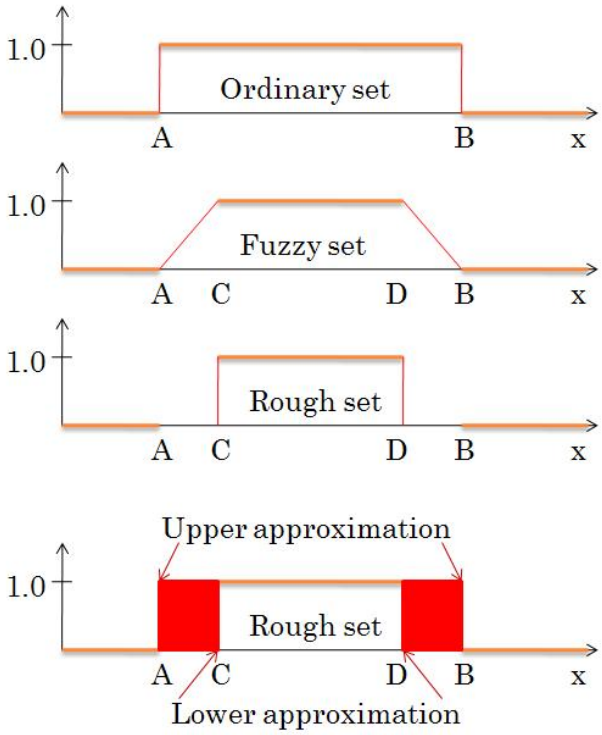


Fig. 9 – Intuitive illustration of a rough set vs an ordinary and a fuzzy set.

In contrast to the traditional concepts of a set, whether ordinary or fuzzy, for a rough set one cannot determine, even partially, the membership of the elements on the set boundary. Therefore, the value of the membership function for boundary elements of a set is undetermined. A rough set can only be described by its approximations, as illustrated in the lower part of Figure 9.

To express these intuitive concepts a bit more formally, we start from a relational database, i.e., a table with rows corresponding to objects and columns corresponding to the attributes. Each entry

of the table represents attribute value of a corresponding object (i.e., its feature). In rough set formalism the database is considered as an information system, i.e., a quadruple

$IS = (U, A, V, f)$,

where $U = \{u_1, \ldots, u_n\}$ stands for a (usually finite) set of objects,

$A = \{a_1, \ldots, a_m\}$ *is a set of attributes*,

$V = V_1 \cup \ldots \cup V_m$,

where V_i is the domain of i-th attribute, and

$f: U \times A \rightarrow V$

is a so-called information function providing the description of objects, i.e., $f(u_i, a_j)$ assigns a value of j-th attribute to i-th object. The above mentioned concepts are illustrated in Table 2, in which

$U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$,

$A = \{a_1, a_2, a_3\}$, and

$V = V_1 \cup V_2 \cup V_3$, = {Low, Med, High}

$\cup$ {Min, Under, Over, Max} $\cup$ {yes, no}.

Usually the set A is decomposed into two disjoint subsets $A = C \cup D$ and the attributes from C are used to characterize objects and form so-called *condition* attributes, while the attributes in D are so-called *decision* attributes and they are used in decision-making or classification tasks. An information system with specified condition and decision attributes is called *decision table*. For the example in Table 2, attributes a_1 and a_2 could be interpreted as condition attributes, that is, certain parameters of an object, with values from the range {Low, Med, High} and {Min, Under, Over, Max}, respectively, and attribute a_3 – as a decision attribute, with values "yes" and "no." Hence, Table 2 can be viewed as a decision table.

Table 2. Example of an information system

$f: U \times A \rightarrow V$	**Attributes A**		
Objects U	a_1	a_2	a_3
u_1	Low	Max	yes
u_2	Low	Min	no
u_3	Med	Under	no
u_4	Med	Under	yes
u_5	High	Over	no
u_6	Low	Over	yes
u_7	High	Over	no
u_8	Low	Min	no

Because of the limited knowledge, we cannot fully discern objects, i.e., there are such objects u, v in U that $f(u, c) = f(v, c)$ for all the condition attributes c. This fact leads to the notion of *indiscernibility* relation E being in fact an equivalence relation on U. For example, for the information system in Table 2, objects u_2 and u_8 are indiscernible. So are objects u_5 and u_7.

It appears that in many cases we can identify proper subsets C' of C such that the indiscernibility relation $\mathbf{E}_{C'}$ induced by the attributes in C' is identical with the original relation $\mathbf{E}$. Such sets of attributes are called *reducts*. Existence of reducts proves that not all of the attributes are necessary to form the equivalence classes. In other words identifying reducts allows more economic description of objects as we need smaller number of descriptors (features) to characterize these objects. Unfortunately, from a computational point of view this is an NP-hard task. No such reducts exist for the example shown in Table 2.

3.2 DEFINITION OF A ROUGH SET

Now we are ready to introduce the key concepts of rough set theory. Let B be a subset of the condition attributes and let $[v]_B$ stand for an equivalence class, i.e., a set of objects u in U with identical description (narrowed to the set B) as the object v. The subset X of U can be characterized using information in B by means of so-called *B-lower* and *B-upper* approximations defined as:

$$B(X)_* = \{u \in U \mid [u]_B \subseteq X\} \quad (1a)$$

$$B(X)^* = \{u \in U \mid [u]_B \cap X \neq \emptyset\} \quad (1b)$$

The lower approximation of X is the collection of objects which can be viewed *surely* as members of the set X, while the upper approximation of X is the collection of objects that *possibly* are members of X. Obviously $B(X)_* \subseteq B(X)^*$. If $B(X)_* = B(X)^*$ we say that X is B-definable and otherwise it is only partially definable. The set $BN_B = B(X)^* - B(X)_*$ is called a *B-boundary* region; it specifies the objects that cannot be classified with certainty to be either inside X, or outside X.

There are many grades of partial definability. We say that the set X is:

- roughly B-definable
  iff $B(X)_* \neq \emptyset$ and $B(X)^* \neq U$,
- internally B-indefinable
  iff $B(X)_* = \emptyset$, $B(X)^* \neq U$,
- externally B-indefinable
  iff $B(X)_* \neq \emptyset$, $B(X)^* = U$,
- totally B-indefinable
  iff $B(X)_* = \emptyset$, $B(X)^* = U$.

Obviously, if $B = C$, i.e., the full set of condition attributes is used, we omit the prefix B- in all above definitions. In such a case, a set X is characterized by the pair (X_*, X^*) and we say that X is a rough set (or B-rough set).

To illustrate these newly introduced concepts for the information system in Table 2, let's distinguish between condition attributes a_1 and a_2, and a decision attribute a_3. Values of a_1 and a_2, can be

interpreted as vague measurements (evaluations) of certain parameters of a technical system, and a_3 can be viewed as a decision (control) based on these evaluations. Let the set B be the entire set of condition attributes, $C = B = \{a_1, a_2\}$. If the equivalence class $[v]_B$ is defined as

$$[v]_B = \{ \{u_1\}, \{u_2, u_8\}, \{u_3, u_4\}, \{u_5, u_7\}, \{u_6\} \}$$

then the set $X = \{ u \mid a_3(x) = \text{yes} \}$ has the following approximations:

$$B(X)_* = \{u_1, u_6\}$$
$$B(X)^* = \{u_1, u_3, u_4, u_6\}$$

This determination can be also illustrated in the table representing the information system under consideration (Table 3).

Table 3. Illustration of lower and upper approximations for a sample information system

f: $U \times A \rightarrow V$	Condition attributes C		Decision attribute D
Objects U	a_1	a_2	a_3
u_1	Low	Max	yes
u_2	Low	Min	no
u_3	Med	Under	no
u_4	Med	Under	yes
u_5	High	Under	no
u_6	Low	Over	yes
u_7	High	Over	no
u_8	Low	Min	no

The set X for a decision variable's value equal "*yes*" has three corresponding objects, u_1, u_4 and u_6. Given values of the specific condition attributes $B = \{a_1, a_2\}$, two of these objects, u_1 and u_6, lead *surely* to this decision value (*yes*). Thus, u_1 and u_6, form the lower approximation of set X. This is illustrated with heavy shading in Table 3. If, however, we take the third object, u_4, the values of its condition attributes, { a_1=Med, a_2=Under}, can produce two different values of the decision attribute: "*yes*" for object u_4, and "*no*" for object u_3. Thus, objects with these values of the condition attributes belong *possibly* to the set X, which is illustrated with light shading in Table 3. Obviously, objects in the non-shaded lines do not belong to X.

To get a numerical characterization of the "roughness" of a set X we introduce so-called *accuracy of approximation*

$$\alpha_{B(X)} = |B(X)_*| \,/\, |B(X)^*| \qquad (2)$$

where the symbol $|Y|$ stands for the cardinality of the set Y. X is said to be crisp (or precise) with respect to the set of attributes B if and only if $\alpha_{B(X)} = 1$, and otherwise X is said to be rough (or vague) with respect to B.

Another characterization of the set of objects can be obtained by introducing so-called *rough membership* function $\mu_{B,X} : U \rightarrow [0,1]$ defined as follows

$$\mu_{B,X}(u) = |[u]_B \cap X| \,/\, |[u]_B| \qquad (3)$$

With such a definition a relationship between rough and fuzzy sets theory is established. More particularly, $\mu_{B,X}(u)$ determines the degree in which object u described by the set B of attributes belongs to the concept (equivalence class) X. Further, we can relax the definitions of the lower and upper approximation, namely

$$B_\beta(X)_* = \{u \in U \mid \mu_{B,X}(x) \geq \beta\} \qquad (4a)$$
$$B_\beta(X)^* = \{u \in U \mid \mu_{B,X}(x) > 1-\beta\} \qquad (4b)$$

where $0 \leq \beta \leq 1$. If $\beta = 1$ we obtain original definitions (1a) and (1b).

3.3 ROUGH RULES

In practical applications of interest are the sets of objects with identical set of decision attributes, that is, we define X as the set of objects satisfying the equality $f(x_1, d) = f(x_2, d)$ for all attributes d in D. If D is, for example, a set of diseases then X is a set of persons suffering on a particular disease, and the equivalence classes $[x]_B$ contain patients with identical symptoms (restricted to the set B). Hence, it is natural to find such condition attributes which can be used to discriminate between different diseases. This leads us to the practical aspects of rough set theory: rough rules.

More formally, given an information system IS = (U, A, V, f) and a subset $B \subseteq A$ we start from the set of atomic formulae, called also descriptors, being expressions of the form $a = v$, where $a \in B$ and $v \in V_a$. Next, we define the set of all possible formulae $F(B,V)$ containing all atomic formulae and being closed with respect to the logical connectives: $\neg$ (negation), $\wedge$ (conjunction) and $\vee$ (disjunction).

If φ is an atomic formula of the form $a = v$, then its meaning (semantics) is as follows:

$$\|\varphi\| = \{u \in U \mid f(u,a) = v\}$$

If φ is a compound formula then

$$\|\neg\varphi\| = U \setminus \|\varphi\|,$$
$$\|\varphi \wedge \varphi'\| = \|\varphi\| \cap \|\varphi'\|, \text{ and}$$
$$\|\varphi \vee \varphi'\| = \|\varphi\| \cup \|\varphi'\|.$$

Now a decision rule is any expression of the form

$$\varphi \Rightarrow (d = v),$$

where d is a decision attribute; the formula φ is said to be predecessor (or ancestor) of the rule and the

formula ($d = v$) – its successor (or consequent). We say that the decision rule:

$$\varphi \Rightarrow (d = v)$$

is true in the information system IS, if

$$\|\varphi\| \subseteq \|(d = v)\| \text{ and } \|\varphi\| \neq \varnothing.$$

Deeper classification of the rules is given in [36].

For instance the rule

$$r_1: (a_1 = \text{Low}) \wedge (a_2 = \text{Max}) \Rightarrow (a_3 = \text{yes})$$

is true in the information system from Table 3, while the rule

$$r_2: (a_1 = \text{Med}) \wedge (a_2 = \text{Under}) \Rightarrow (a_3 = \text{yes})$$

is only partly true because

$$\|(a_1 = \text{Med}) \wedge (a_2 = \text{Under})\| = \{u_3, u_4\}$$
$$\text{and } \|(a_3 = \text{yes})\| = \{u_1, u_4, u_6\};$$

thus $\|(a_1 = \text{Med}) \wedge (a_2 = \text{Under})\| \not\subset \|(a_3 = \text{yes})\|$. Detailed remarks on inducing rules from information systems can be found, for example, in [37].

To characterize the rules numerically, a number of measures can be introduced; *support* and *confidence* are most popular. The former is defined as the number of objects satisfying both predecessor and successor, while the latter as the conditional probability that the consequent is satisfied provided the ancestor is satisfied. In case of rule r_2 its support, $sup(r_2) = 1$, and its confidence, $conf(r_2) = ½$.

The already mentioned process of "transformation of data into knowledge" translates now into refining the dependencies between sets of attributes. Intuitively, if C and D are two sets of attributes, we say that D depends totally on C, if all values of the attributes from D are uniquely determined by values of attributes from C. This is functional dependency known from database theory.

Rough set theory enables relaxing this definition by introducing a dependency in a degree $k \in (0, 1]$. An interested reader is referred to [19] and [38] for details. There are at least two successful computer programs allowing rough data analysis: Rosetta [39] downloadable from the following website: http://rosetta.lcb.uu.se/general/ and LERS [40].

Finally if a new object is introduced into the data set with the attribute value missing, one could attempt to determine this value by using the previously generated rules. This is explained in the next section.

3.4 HANDLING THE MISSING VALUE IN A ROUGH SET

Grzymala-Busse describes several algorithms of dealing with missing values in information systems, based on three types of such values [41]:

- those which are lost and no longer available
- totally irrelevant values, and
- partially relevant values.

They are marked in Table 4, using the following symbols: a question mark "?" for not available values, an asterisk "*" for irrelevant values, and a dash "-" for partially relevant values.

Table 4. Information system with some missing values

$f: U \times A \rightarrow V$	Condition attributes C		Decision attribute D
Objects U	a_1	a_2	a_3
u_1	?	Max	yes
u_2	Low	Min	no
u_3	Med	Under	no
u_4	-	Under	yes
u_5	High	Over	no
u_6	Low	Over	yes
u_7	High	Over	no
u_8	Low	*	no

To calculate the approximations, one has to start with the meaning of the atomic formulas in a given information system. For the information system in Table 2, these meanings, called also *blocks* in [41] are as follows:

$\|a_1 = \text{Low}\| = \{ u_1, u_2, u_6, u_8 \}$
$\|a_1 = \text{Med}\| = \{ u_3, u_4 \}$
$\|a_1 = \text{High}\| = \{ u_5, u_7 \}$
$\|a_2 = \text{Min}\| = \{ u_2, u_8 \}$
$\|a_2 = \text{Under}\| = \{ u_3, u_4 \}$
$\|a_2 = \text{Over}\| = \{ u_5, u_6, u_7 \}$
$\|a_2 = \text{Max}\| = \{ u_1 \}$

These sets have to be modified for an information system with missing values in Table 4, as follows. For the missing value of the attribute a_1, which is not available for object u_1 and marked "?", object u_1 has to be removed from all blocks created for this attribute, that is, block $\|a_1 = \text{Low}\|$ will change to:

$$\|a_1 = \text{Low}\| = \{ u_2, u_6, u_8 \}$$

with two other blocks for a_1 remaining unchanged, because they do not include objects with lost value of a_1.

For the missing value of the attribute a_2, which is irrelevant and marked "*", its corresponding object, u_8, has to be included in blocks for all values of this attribute, which will lead to the following modifications:

$\|a_2 = \text{Min}\| = \{ u_2, u_8 \}$
$\|a_2 = \text{Under}\| = \{ u_3, u_4, u_8 \}$
$\|a_2 = \text{Over}\| = \{ u_5, u_6, u_7, u_8 \}$
$\|a_2 = \text{Max}\| = \{ u_1, u_8 \}$

Finally, for the missing value of the attribute a_1, which is marked "-", as partially relevant, respective object u_4 has to be added to the blocks containing

objects corresponding to the decision attribute's value the same as the value of this decision attribute for the partially relevant value. In case of Table 4, the partially relevant value of attribute a_1 for object u_4 corresponds to the decision attribute's value "yes". Thus, this attribute's value is relevant to this particular decision attribute, and this is the meaning of the term "partially relevant". Two other objects exist, which have "yes" as their decision attribute's value: u_1, whose value of attribute a_1 is unavailable, so we drop it from consideration, and u_6, whose value of a_1 equals *Low*; therefore u_4 has to be added to the block, which contains a_1 = Low, because it is partially relevant to corresponding decision attribute.

So the final list of blocks looks as follows:

$||a_1 = \text{Low}|| = \{ u_2, u_4, u_6, u_8 \}$
$||a_1 = \text{Med}|| = \{ u_3, u_4 \}$
$||a_1 = \text{High}|| = \{ u_5, u_7 \}$
$||a_2 = \text{Min}|| = \{ u_2, u_8 \}$
$||a_2 = \text{Under}|| = \{ u_3, u_4, u_8 \}$
$||a_2 = \text{Over}|| = \{ u_5, u_6, u_7, u_8 \}$
$||a_2 = \text{Max}|| = \{ u_1, u_8 \}$

Because of the limited length of this paper, we can only mention here that for further computations the so called *characteristic sets* have to be calculated, for each object, which is done as follows:

1) The characteristic set ***K*** of an object is defined as an intersection of blocks for specific values of the attributes for this object.
2) If the value of an attribute is irrelevant "*" or unavailable "?", then the entire universe U is taken as a corresponding block for this attribute.
3) If the value of an attribute is partially relevant "-", then for this specific block it is substituted by a union of blocks representing particular values of the attributes for the corresponding decision attribute's value.

A more formal presentation of these concepts, with respective algorithms, is given in [41]. Below we present the computation of characteristic sets for the list of blocks corresponding to Table 4.

$K_{u1} = U \cap \{ u_1, u_8 \} = \{ u_1, u_8 \}$
$K_{u2} = \{ u_2, u_4, u_6, u_8 \} \cap \{ u_2, u_8 \} = \{ u_2, u_8 \}$
$K_{u3} = \{ u_3, u_4 \} \cap \{ u_3, u_4, u_8 \} = \{ u_3, u_4 \}$
$K_{u4} = \{ u_2, u_4, u_6, u_8 \} \cap \{ u_3, u_4, u_8 \} = \{ u_4, u_8 \}$
$K_{u5} = \{ u_5, u_7 \} \cap \{ u_5, u_6, u_7, u_8 \} = \{ u_5, u_7 \}$
$K_{u6} = \{ u_2, u_4, u_6, u_8 \} \cap \{ u_5, u_6, u_7, u_8 \} = \{ u_6, u_8 \}$
$K_{u7} = \{ u_5, u_7 \} \cap \{ u_5, u_6, u_7, u_8 \} = \{ u_5, u_7 \}$
$K_{u8} = \{ u_2, u_4, u_6, u_8 \} \cap U = \{ u_2, u_4, u_6, u_8 \}$

As explained in [41], computation of lower and upper approximations, depends on their definitions. The author presents three such definitions and for one of them:

$$B(X)_* = \{ u_1, u_4, u_6, u_8 \}$$
$$B(X)^* = \{ u_1, u_2, u_4, u_6, u_8 \}$$

The interpretation of this result is such that the missing values cause broadening of the potential span for the lower approximation, because they have to be inferred from the rest of the set. The upper approximation can change either way, because the missing values change the entire structure of a set.

4. COMBINATION OF BBN'S WITH ROUGH SETS

This section describes several examples and two case studies related to Bayesian networks and rough sets. First, we give a background on applying Bayesian networks to software quality evaluation. Next, we discuss a case study on the assessment and qualification of software tools for real-time safety-critical systems. Finally, we present a method for combining Bayesian networks and rough sets in decision making under uncertainty, and discuss the operation of two public domain tools, assisting in real-time decision making.

4.1 USE OF BBN'S FOR SOFTWARE QUALITY EVALUATION

In recent years, these authors have dealt with various aspects of assessing software quality in real-time safety-critical applications [42-44]. The basic idea to apply Bayesian networks in such problems comes from multiple previous attempts to assess various software properties in critical applications, which are briefly outlined below.

A. Application of BBN's to Assess Software Quality. In one of the first studies reported [45], the authors addressed the eternal question: "Can we predict the quality of our software *before* we can use it?", by applying BBN's to assess the *defect density* as a measure of software quality. A simplified diagram from their study is presented in Figure 10. The nodes were built based on the understanding of life-cycle processes, from requirements specification through testing.

The probabilities of respective states were based on the analysis of literature and common-sense assumptions about the relations between variables. The node variables are shown on histograms of the predictions obtained by execution of the network after the evidence entered (the evidence is represented by nodes with probabilities equal to 1.0). As the authors say, the advantage of their model is that it "provides a way of simulating different events and identifying optimum courses of action based on

uncertain knowledge."

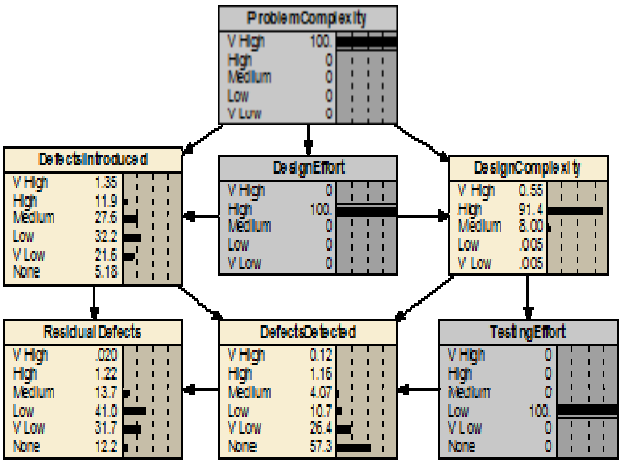


Fig. 10 – Simplified BBN for assessing software fault density [45].

B. Use of BBN's in Assessment of Software Safety. Gran et al. [46] applied BBN's to address safety assessment of software for acceptance purposes, in a more comprehensive way, using multiple information sources, such as complexity, testing, user experience, system quality, etc.

Their BBN network for system quality, which is only a part of the entire model, is shown in Figure 11. It involves two root nodes: *UserExperience* and *VendorQuality*, and a number of leaf nodes, corresponding to observable variables, of which *QualityMeasures* is of particular importance. This node shows evidence about the system quality, grouping quality attributes, such as readability, structuredness, etc., and can be expanded further.

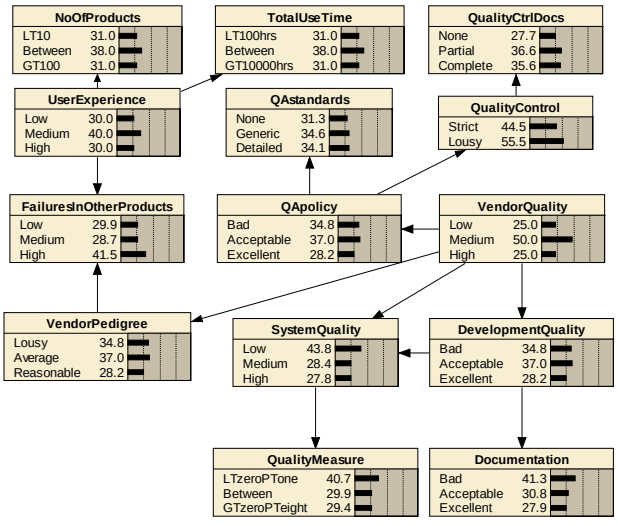


Fig. 11 – BBN for the system quality in safety assessment [46]

Other observable variables include *FailuresInOtherProducts*, those related to the user experience (*NoOfProducts* and *TotalUseTime*), as well as those related to quality assurance policy. When evidence becomes available, entering respective observation data into these nodes and executing the network provides assessment of the variable in question, which in this case is *SystemQuality.*

The authors note, however, that their example is intended more as an illustration of the method rather than as a real attempt to compute the quality of the system. Their probability assignments to the node variables were chosen somewhat *ad hoc*, and not based on any deeper analysis of the problem. However, as the authors say in conclusion, the results of the study were positive and showed "that the method reflects the way of an assessor's thinking during the assessment process."

C. BBN's in Dependability and Reliability Assessment. [47] used BBNs to formalize reasoning about software dependability to facilitate the software assessment process. They constructed a network for evaluating dependability of a software-based safety system. It used the data associated with two primary assumptions: the excellence in development (called a process argument) and failure-free statistical testing (called a product argument). The network topology includes taking into consideration variables such as: Test Failures, Operational Failures, Initial Faults, Faults Found, Faults Delivered, and System PFD (Probability of Failure per Demand). The probability distributions have been derived from a sample of programs from an academic experiment.

The authors were interested in estimating the probabilities of failure during acceptance testing and during the operational life of the product (represented by two variables mentioned in previous paragraph), given the prior probabilities and observed events. In particular, positive results of an acceptance test allowed deriving numerical estimates about the PFD and operational performance of the product.

Helminen [13] used BBN's to attack the problem of software reliability estimation. His primary motivation to apply BBN's was that they allow all possible evidence (large number of variables, different potential sources, etc.) to be used in the analysis of the reliability of a programmable safety-critical system. The essential characteristic of such systems is that they involve a significant number of variables related to reliability, with very limited evidence.

The reliability of such systems is modeled as a *probability of failure*, that is, the probability that the programmable system fails when it is required to operate correctly. To develop an estimate of

probability of failure, the authors built a series of BBN models, using evidence from such sources, as the system development process, system design features, and pre-testing, before the system is deployed. This is later enhanced by data from testing and operational experience.

The essential part of this work was building BBN models for various operational profiles for multiple test cycles, involving continuous probability distributions. As a result, using BUGS software that combines Bayesian inference with Gibbs sampling, via Markov chain Monte Carlo (MCMC) simulation, it was possible to estimate, how many tests had to be run for a single system in a particular operational environment to achieve certain level of reliability. To decrease the huge number of necessary tests, multiple operational profiles for the same system were used, which required building replicated BBN models to include other profiles' evidence. In essence, by expanding the BBN models further, this approach also allows reliability estimation over the entire lifespan of the software product, but respective experiments have not been conducted in this study.

4.2 CASE STUDY IN SOFTWARE TOOL EVALUATION

To test the applicability of BBNs in software assessment, we applied this technique to evaluate the software development tools used in real-time safety critical applications in avionics. The data for the project were taken from experiments described in detail elsewhere [43,48]. The experiments involved applying a number of specific criteria, including: *efficiency* of the generated code, to conduct forward evaluation regarding the quality of code, and *traceability*, to allow backward evaluation regarding the tool capability of maintaining the right requirements. To evaluate the tool during its operation from perspective of the functions it provides and the ease of use, two additional criteria seemed to be appropriate: *functionality* and *usability*. The exact process of choosing the criteria is described in [48]. For criteria selected that way, a series of experiments were conducted, with six industry-strength tools applied to embedded software development. The above mentioned criteria were quantified using the following measures:

- *Efficiency* measured as code size (in LOC)
- *Usability* measured as development effort (in hours)
- *Functionality* measured via the questionnaire (on a 0-5 points scale)
- *Traceability* measured by manual tracking (in number of defects).

Data for some measures were collected in multiple aspects, for example, data involving the development effort were divided into four categories: preparation, modeling and code generation, measurements, and postmortem (including report writing). Details of the software requirements and actual experimental results are discussed in [43].

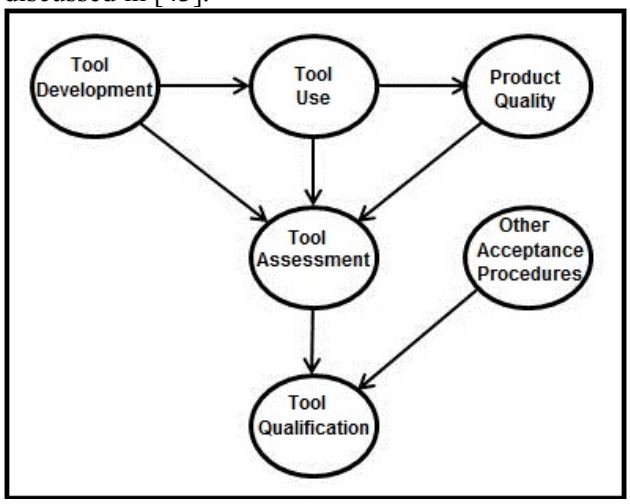


Fig. 12 – High-level BBN model for software tool evaluation.

Based on the adopted model of the tool evaluation process, and the results of experiments with the selected evaluation criteria outlined above, our high-level model of a BBN for tool assessment is illustrated in Figure 12. Its primary assumptions are that the tool assessment process should involve the following mutually interrelated factors:

a) development of the tool itself (including the process, vendor quality and reputation, their quality assessment procedures, etc.),

b) the tool use (including experimental evaluation based on predefined criteria, but also previous user experiences with this tool, etc.)

c) quality of the products developed with this tool, based on product execution, static code analysis, etc.

Based on the results of this analysis and other acceptance procedures (such as, legal aspects, independent experts opinions, etc.), the tool qualification process can be completed, as reflected in a BBN in Figure 13. Because of the limited data obtainable from experiments, we only deal with *ToolUse* part of the diagram in Figure 12. The logic of the BBN is similar to the ones reported in [14], where they had no real probability data, and [46], where the conditional probability values "were estimated based on judgments in a brainstorming activity among the project participants."

For the experimentally collected data for six tools, nicknamed L, M, N, O, P and Q, a sample tool

assessment BBN is shown in Figure 13 for a tool, which is likely to pass the qualification process with 80% confidence at the level *MediumToHigh* or *High.*

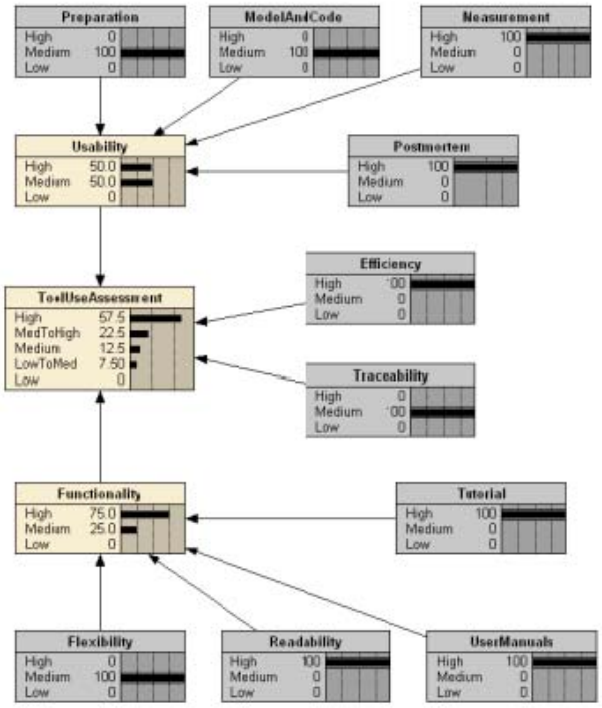


Fig. 13 – BBN to assess numerically quality of tool L.

4.3 REAL-TIME APPLICATION: THE AUSTRALIAN NAVY EXERCISE

As visible from previous examples, the principle of using a BBN for reasoning under uncertainty is that when the evidence about the state of one of the nodes (variables) becomes available, the rest of the network is also updated according to the conditional probability tables and dependency relations among the nodes. However, an updating process becomes a problem, if the new evidence is distorted or missing. This situation does not look that difficult in off-line computations, such as those discussed in subsections above, because one can do additional experiments and wait for the data when they will become complete. But if one wants to use BBN's for situation assessment in real time, when missing or distorted data come into play, as in circumstances such as sensor noise or sensor failure, especially over extended period of time, then the value of Bayesian reasoning may become problematic.

In general, this issue comes into play when there is no information on certain behavior or some information previously available becomes scarce or unavailable. Then using a rough set theory can help filling the gap caused by such circumstances. To illustrate this concept, we present a case study of the Australian Naval Exercise [49].

In this case study, there are two naval military forces called Blue and Orange that are hostile towards each other, and a country that the Orange forces obtain fuel supplies from and the Blue forces treat as neutral. The Blue forces have communications and surveillance facilities that the Orange forces want to destroy. Blue have set up a restricted area that contains the communication facilities and will consider any military activity or transportation of supplies hostile. Orange have a supply route that passes through the restricted area that it wants to defend.

Blue monitor the restricted area via sensors and reconnaissance. Orange vessels that are likely to be detected are Guided Missile Frigates (*FFG* in Figure 14), Free Mantle Class Patrol Boards (*FCPB*), and *Communication* vessels. Oil *Tankers* from the neutral country can also be detected. The position, mobility, and communications activity of the vessel are also recorded to try to determine the intent of the Orange forces.

The Bayesian network in Figure 14 is used to try to determine what the intentions of the Orange forces are and how to respond to it by entering the findings from the sensors and reconnaissance into the appropriate nodes. In essence, based on this information entered into the bottom nodes, the Bayesian network recalculates the variables in all other nodes, and the value of a variable in node *BlueCOA* makes a suggestion to the decision maker, what would be the most appropriate Course of Action (COA) at any given time.

The situation is more complicated when some of the sensor or reconnaissance data are missing, for example, due to a sensor failure or temporary or permanent unavailability of the reconnaissance. The BBN, which does the calculations, still expects receiving new data, because the command unit has to assess the situation and make respective decisions in real time. Even though the BBN can still operate, the missing data make its assessments less and less accurate when the time progresses.

In such case, we try to employ a rough set theory, particularly in its part dealing with the missing values. The essential idea is as follows. If we treat specific variables from the BBN network as attributes of the information system (rough set), with one of them being the decision attribute and all remaining ones – condition attributes, then we can determine (with some level of accuracy) the missing values of the attributes, using the reasoning presented briefly in the section on rough sets and described in more detail in [41]. In plain language, this would be equivalent to deriving the approximate value of a certain variable based on the context information. A sample of a respective information system is illustrated in Figure 15, for the Australian

Naval Exercise, using a rough set tool Rosetta [39].

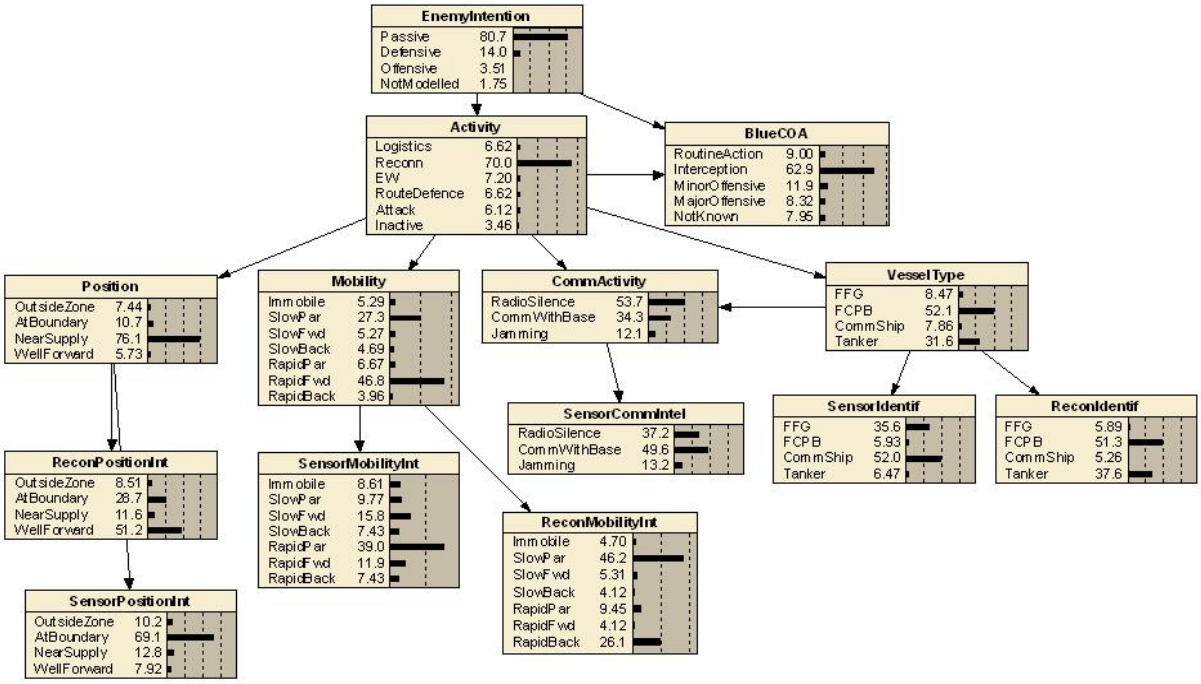


Fig. 14 – Sample BBN for an Australian Naval Exercise

AustralianSample0

	SensorMobilityInt	Activity	BlueCOA	VesselType	CommActivity	Position	Mobility	SensorIdentif	ReconIdentif	SensorCommIntel	ReconPositionInt	SensorPositionInt	ReconMobilityInt	EnemyIntention
1	RapidPar	Reconn	Interception	Tanker	CommWithBa	NearSupply	SlowPar	FFG	Tanker	CommWithBa	AtBoundary	AtBoundary	RapidBack	**Passive**
2	SlowPar	Reconn	Interception	FFG	Jamming	NearSupply	SlowFwd	FFG	Tanker	CommWithBa	OutsideZone	AtBoundary	Immobile	**Defensive**
3	RapidPar	EW	MajorOffensi	FFG	RadioSilence	NearSupply	RapidPar	Tanker	Tanker	Jamming	NearSupply	OutsideZone	SlowPar	**Defensive**
4	RapidPar	RouteDefenc	MinorOffensi	FFG	Jamming	NearSupply	RapidFwd	FFG	Tanker	CommWithBa	OutsideZone	WellForward	RapidPar	**Passive**
5	RapidFwd	Reconn	MinorOffensi	FCPB	RadioSilence	AtBoundary	RapidFwd	FCPB	FFG	CommWithBa	NearSupply	OutsideZone	RapidPar	**Passive**
6	Immobile	RouteDefenc	NotKnown	Tanker	CommWithBa	NearSupply	RapidPar	FCPB	CommShip	RadioSilence	AtBoundary	NearSupply	SlowPar	**Passive**
7	RapidFwd	Logistics	MinorOffensi	FCPB	Jamming	NearSupply	RapidPar	Tanker	Tanker	RadioSilence	AtBoundary	NearSupply	RapidPar	**Passive**
8	SlowFwd	Attack	Interception	FCPB	CommWithBa	AtBoundary	RapidPar	CommShip	FFG	RadioSilence	OutsideZone	WellForward	SlowPar	**Defensive**
9	Immobile	Attack	MinorOffensi	FCPB	Jamming	NearSupply	SlowPar	Tanker	FCPB	CommWithBa	NearSupply	NearSupply	SlowPar	**Passive**
10	Immobile	RouteDefenc	MinorOffensi	FCPB	CommWithBa	NearSupply	SlowBack	FCPB	CommShip	CommWithBa	NearSupply	NearSupply	RapidPar	**Passive**
11	SlowPar	Logistics	RoutineAction	Tanker	CommWithBa	OutsideZone	Immobile	FCPB	CommShip	CommWithBa	NearSupply	NearSupply	RapidPar	**Passive**
12	SlowPar	EW	RoutineAction	Tanker	CommWithBa	NearSupply	SlowFwd	FFG	Tanker	CommWithBa	NearSupply	AtBoundary	RapidPar	**Defensive**
13	RapidPar	Reconn	Interception	Tanker	RadioSilence	AtBoundary	RapidFwd	Tanker	Tanker	RadioSilence	NearSupply	AtBoundary	RapidPar	**NotModelle**
14	RapidPar	EW	NotKnown	CommShip	RadioSilence	AtBoundary	RapidPar	FFG	Tanker	RadioSilence	WellForward	OutsideZone	RapidPar	**Passive**
15	RapidFwd	Reconn	NotKnown	FCPB	Jamming	OutsideZone	RapidPar	FFG	Tanker	RadioSilence	NearSupply	OutsideZone	SlowFwd	*
16	*	Reconn	MajorOffensi	FCPB	RadioSilence	NearSupply	RapidFwd	CommShip	FCPB	*	WellForward	*	SlowPar	*

Fig. 15 – Sample information system in Rosetta for an Australian Naval Exercise

All fourteen nodes from the Bayesian network are mapped onto attributes of an information system. In each time instant, depending on the frequency of measurements in the decision making process, a new case (an object with fourteen attributes) is created. The values of respective attributes may be obtained directly by the measurement process, or from a BBN if necessary. For example, the first attribute in Figure 15, *SensorMobilityInt*, corresponds to the node of the same name in the BBN in Figure 14, and has a value of *RapidParallel*. If some measurements are missing, this is illustrated by an asterisk in Figure 15.

The operation of software tools to conduct this process in real time is illustrated in Figure 16, with evidence meaning the new sensor measurements or reconnaissance data. Such process can be easily automated with existing tools, since a Netica version exists that has a Java API and can read cases from a text file. In turn, Rosetta, which also has a command language interface, can export its tables as text files to be grabbed by Netica. With a converter software reading Rosetta files, making respective adjustments if some data are missing, and transforming them to the Netica format, the whole system shown in Figure 16 can operate smoothly and enhance the decision making process in real time.

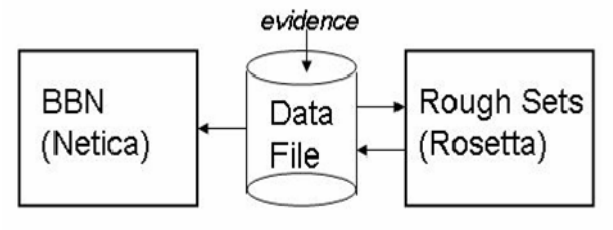


Fig. 16 – Real-Time operation of a BBN tool with a rough set tool.

5. SUMMARY AND CONCLUSION

This paper discussed basic concepts of Bayesian belief networks and rough sets, and showed how they can be combined to enhance the process of reasoning under uncertainty in case of missing values of certain attributes of objects. Bayesian networks and rough sets are individually very adequate tools to solve computational problems with insufficient information and reason about uncertainty. The use of rough sets helps making BBN's more valuable in case of the occasional lack of evidence. It becomes particularly important, when BBN's are used in applications such as real-time decision making or active safety diagnostics, with information being supplied to the nodes during operation. In such cases, losing the source of information for one of the BBN nodes impairs the inference process in the next steps. Using rough set reasoning helps in keeping the BBN in good standing, disregarding the lost source of information.

This logic of this process is very similar to the use of a Kalman filter [50], when the information about the system is updated based on its previous behavior. However, in case of rough sets the information does not have a statistical nature, as in the case of Kalman filtering. Comparing the concepts outlined in this article with the operation of a Kalman filter would be a good topic for further study.

There are several important questions still to be addressed. For example, to apply this method in practice, one would need to know how computationally intensive are the rough set calculations? It seems that for typical applications of Bayesian belief networks, which are used in decision support systems, the deadlines for completing the computations are most likely in the order of minutes or hours, so this issue should not cause problems.

6. ACKNOWLEDGEMENT

The authors gratefully acknowledge contribution to this paper of the late Dr. Henry Pfister.

Part of this work has been funded by a grant SBAHQ-10-I-0250 from the U.S. Small Business Administration (SBA). SBA's funding should not be construed as an endorsement of any products, opinions, or services. All SBA-funded projects are extended to the public on a nondiscriminatory basis.

7. REFERENCES

[1] N.E. Fenton, M. Neil, The use of Bayes and causal modelling in decision making, uncertainty and risk, *Agena Risk White Paper,* June 2011. Available at: http://www.agenarisk.com/resour ces/white_papers/fenton_neil_white_paper2011 .pdf.

[2] F.V. Jensen, T.D. Nielsen, *Bayesian Networks and Decision Graphs. Second Edition*, Springer-Verlag, 2007.

[3] W.J. Dawsey, *Bayesian belief networks to integrate monitoring evidence of water distribution system contamination,* Master Thesis, University of Illinois at Urbana-Champaign, February 2012.

[4] M. Azhdari, N. Mehranbod, Application of Bayesian belief networks to fault detection and diagnosis of industrial processes, *Proc. ICCCE 2010, Intern. Conf. on Chemistry and Chemical Engineering*, Kyoto, Japan, August 1-3, 2010, pp. 92-96.

[5] P. Newham et al., Fog forecasting at Melbourne airport using Bayesian networks, *Proc. Fourth Intern. Conf. on Fog, Fog Collection and Dew*, Santiago, Chile, July 22-27, 2007, pp. 291-294.

[6] N. Fenton et al., Predicting software defects in varying development lifecycles using Bayesian nets, *Information and Software Technology*, (49) 1 (2007), pp. 32-43.

[7] C. Lee et al., Inferring certification metrics of package software using Bayesian belief networks, *Proc. ICIC 2006 – Intern. Conf. on Intelligent Computing*, Kunming, China, August 16-19, 2006, pp. 915-920.

[8] A. Leger et al., Predicting hospital admission for Emergency Department patients using a Bayesian network, *Proc. AMIA Annual Symposium*. Washington, DC, October 22-26, 2005, p. 1022.

[9] S. Ferrari, A. Vaghi, Multisensor fusion for landmine detection by a Bayesian network approach, *Proc. ECSC – 3rd European Conf. on Structural Control*, Vienna, Austria, July 12-15, 2004.

[10] A.K.T. Hui et al., A Bayesian belief network model and tool to evaluate risk and impact in software development projects, *Proc. RAMS 2004 – Annual IEEE Reliability and Maintainability Symposium*, Los Angeles, Calif., January 26-29, 2004, pp. 297-301.

[11] M. Neil, B. Mancom, R. Shaw, Modelling an air traffic control environment using Bayesian belief networks, *Proc. ISSC'03 – 21st Intern. System Safety Conf.*, Ottawa, Ontario, August 4-8, 2003.

[12] K. Sachs et al., Bayesian network approach to cell signaling pathway modeling, *Science STKE*, 148, PE38, 2002.

[13] A. Helminen, *Reliability Estimation of Safety-Critical Software-Based Systems Using Bayesian Networks*, Report STUK-YTO-TR 178, Radiation and Nuclear Safety Authority, Helsinki, Finland, 2001.

[14] G. Dahll, B.A. Gran, The use of Bayesian belief nets in safety assessment of software based systems, *International Journal of General Systems*, (29) 2 (2000), pp. 205-229.

[15] N.E. Fenton, M. Neil, Bayesian belief nets: a causal model for predicting defect rates and resource requirements, *Software Testing and Quality Engineering*, (2) 1 (2000), pp. 48-53.

[16] P. Smets, Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem, *International Journal on Approximate Reasoning*, (9) 1 (1993), pp. 1-35.

[17] J.A. Bernardo, A.F.M. Smith, *Bayesian Theory*, John Wiley and Sons, 1994.

[18] Z. Pawlak, Rough Sets, *International Journal of Information and Computer Sciences*, (11) 5 (1982), pp. 341-356, Available at: http://chc60.fgcu.edu/Images/articles/PawlakOriginal.pdf

[19] Z. Pawlak, *Rough Sets – Theoretical Aspects of Reasoning about Data,* Kluwer Academic Publishers, 1991.

[20] Ji Zhang et al., Neighborhood rough sets for dynamic data mining, *Intern. Journal of Intelligent Systems*, (27) 4 (2012), pp. 317-342.

[21] Y. Li et al., A generalized model of covering rough sets and its application in medical diagnosis, *Proc. ICMLS 2010, Intern. Conf. on Machine Learning and Cybernetics*, Qingdao, China, July 11-14, 2010, pp. 145-150.

[22] M. Bit, T. Beaubouef, Rough set uncertainty for robotic systems, *Journal of Computing Sciences in Colleges*, (23) 6 (2008), pp. 126-132.

[23] G. Ilczuk, A. Wakulicz-Deja, Visualization of rough set decision rules for medical diagnosis systems, *Proc. RSFDGrC 2007 – 11th Intern. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Toronto, Canada, May 14-16, 2007, pp. 371-378.

[24] J. Stefanowski, An empirical study of using rule induction and rough sets to software cost estimation, *Fundamenta Informaticae*, (71) 1 (2006), pp. 63-82.

[25] P.A. Laplante, C.J. Neill, Modeling uncertainty in software engineering using rough sets, *Innovations in Systems and Software Engineering – A NASA Journal*, (1) 1 (2005), pp. 71-78.

[26] Z. Li, G. Ruhe, Uncertainty handling in tabular-based requirements using rough sets, *Proc. RSFDGrC 2005 – Intern. Conf. on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Regina, Canada, 31 August – 3 September, 2005.

[27] R. Wasniowski, A framework for software safety analysis with rough sets, *Proceedings of the 4th WSEAS Intern. Conf. on Software Engineering, Parallel & Distributed Systems*, Salzburg, Austria, February 13-15, 2004.

[28] J.F. Peters, H. Feng, S. Ramanna, Adaptive granular control of an HVDC system: A rough set approach, *Proc. of RSFDGrC 2003 – 7th Intern. Conf. on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Chingqing, China, May 26-29, 2000, pp. 213-220.

[29] N. Meskens, P. Levecq, F. Lebon, Multivariate analysis and rough sets: two approaches for software-quality analysis, *International Transactions in Operational Research*, (9) 3 (2002), pp. 353-369.

[30] L. Shen et al., Fault diagnosis using rough sets theory, *Computers in Industry*, (43) 1 (2000), pp. 61-72.

[31] J. Zalewski, Z. Wojcik, Use of Artificial Intelligence Techniques for Prognostics: New Application of Rough Sets, *Intern. Journal of Computing*, (11) 1 (2012), pp. 73-81.

[32] T. Bayes, Essay towards solving a problem in the doctrine of chances, *Philosophical Transactions of the Royal Society of London*, 53, (1763), pp. 370-418.

[33] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan-Kaufmann, 1988.

[34] Norsys Software Corporation. *Netica. Version 4.08*, Vancouver, Canada, URL: http://www.norsys.com

[35] Hugin Expert A/S. *Hugin Developer Software Package*. Aalborg, Denmark. URL: http://www.hugin.com/

[36] M. Kryszkiewicz, Comparative study of alternative types of knowledge reduction. *International J. of Intelligent Systems*, (16) 1 (2001), pp. 105-120.

[37] W. Ziarko, Rough sets as a methodology for data mining, In: *Rough Sets in Knowledge Discovery 1: Methodology and Applications*. Physica-Verlag, 1998, pp. 554-576.

[38] J. Komorowski, L. Polkowski, A. Skowron, Rough sets: a tutorial. In: S.K. Pal and A.

Skowron (Eds.), *Rough-Fuzzy Hybridization: A New Method for Decision Making*, Springer-Verlag, 1998.

[39] A. Øhrn, J. Komorowski, J. Rosetta, A rough set toolkit for analysis of data, *Proc. RSSC'97 – Third Intern. Joint Conf. on Information Sciences, Fifth Inter. Workshop on Rough Sets and Soft Computing*, Durham, NC, (3) (1997), pp. 403-407.

[40] J. Grzymała-Busse. LERS – A system for learning from examples based on rough sets. In: R. Słowiński (Ed.), *Intelligent Decision Support: Handbook of Applications and Advances of Rough Set Theory* (pp. 3-18), Kluwer Academic Publishers, 1992.

[41] J. Grzymala-Busse, Three approaches to missing attribute values – a rough set perspective, *Proc. Workshop on Foundation of Data Mining – 4th IEEE Intern. Conf. on Data Mining*, Brighton, UK, November 1-4, 2004.

[42] I.E. Chen-Jimenez, A. Kornecki, J. Zalewski, Software safety analysis using rough sets, *Proc. IEEE Southeastcon'98*, Orlando, Fla., April 24–26, 1998, pp. 15-19.

[43] A. Kornecki, J. Zalewski, Experimental evaluation of software development tools for safety-critical real-time systems, *Innovations in Systems and Software Engineering – A NASA Journal*, (1) 2 (2005), pp. 176-188.

[44] A. Kornecki, J. Zalewski, Software development tools qualification from the DO-178B certification perspective, *Crosstalk – The Journal of Defense Software Engineering*, (19) 4 (2006), pp. 19-22.

[45] M. Neil, N. Fenton, Predicting software quality using Bayesian belief networks, *Proc. SEW-21 – Annual NASA Goddard Software Engineering Workshop*, Washington, DC, December 4-5, 1996, pp. 217-230.

[46] B.A. Gran et al., Estimating dependability of programmable systems using BBNs, *Proc. SAFECOMP 2000 – 19th Intern. Conference on Computer Safety, Reliability and Security*, Rotterdam, The Netherlands, October 24-27, 2000, pp. 309-320.

[47] K.A. Delic, F. Mazzanti, L. Strigini, Formalising engineering judgement on software dependability via belief networks, *Proc. DCCA-6 – 6th IFIP Intern. Working Conf. on Dependable Computing for Critical Applications*, Garmisch-Partenkirchen, Germany, March 5-7, 1997, pp. 291-305.

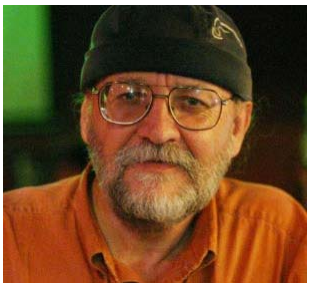
[48] A. Kornecki, N. Brixius, J. Zalewski, *Assessment of Software Development Tools for Safety-Critical, Real-Time Systems*, Report DOT/FAA/AR-06/36, Federal Aviation Administration, Washington, DC, 2007.

[49] B. Das, *Representing Uncertainties Using Bayesian Networks*, Report DSTO-TR-0918, Defence Science and Technology Organisation, Electronics and Surveillance Research Laboratory, Sydney, Australia, 1999.

[50] R.G. Brown, P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering, Third Edition*, John Wiley and Sons, 1997.

Andrew J. Kornecki *is a professor at the Department of Electrical, Computer, Software and System Engineering at the Embry Riddle Aeronautical University. He has more than 20 years of research and teaching experience in areas of real-time computer systems. He has been conducting industrial training on real-time, safety-critical software in medical and aviation industries and for the FAA Certification Services. Recently, he has been engaged in work on certification issues and assessment of development tools for real-time, safety-critical systems.*

Slawomir T. Wierzchon *is a professor of computer science at the Institute of Informatics of the University of Gdansk in Poland, and at the Institute of Computer Science of the Polish Academy of Sciences. He has been the Program Chair of multiple international conferences and serves on the Editorial Board of the International Journal of Biometrics. His research interests include computational intelligence, biologically inspired computations, machine learning, data analysis, and spectral graph theory.*



Janusz Zalewski *is a professor of Computer Science and Software Engineering at Florida Gulf Coast University, in Ft. Myers, Florida, USA. He previously worked at nuclear research labs in Europe and the U.S. and consulted for the government and industry. He also had fellowships at NASA and Air Force Research Labs. His research interests include real-time embedded and cyberpysical systems, prognostics of complex systems, and software engineering education.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

ОПТИМАЛЬНИЙ СИНТЕЗ ЗВОРОТНИХ КВАНТОВИХ СУМАТОРІВ З ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Віталій Дейбук, Іон Грицку

Кафедра комп'ютерних систем та мереж, факультет комп'ютерних наук, Чернівецький національний університет ім. Юрія Федьковича, вул. Коцюбинського 2, 58012, Чернівці, Україна,
e-mail: v.deibuk@chnu.edu.ua, ion_grytsku@mail.ru

Резюме: *У роботі запропоновано новий спосіб кодування хромосом у генетичному алгоритмі для моделювання схем зворотних повних однорозрядних суматорів з функцією транзиту у базисі елементів Фредкіна. Отримані з допомогою такого підходу схеми мають кращі параметри затримки та кількості зайвих виходів(входів) порівняно з відомими аналогами, що демонструє ефективність та застосовність такого підходу.*

Ключові слова: *генетичні алгоритми, еволюційна електроніка, зворотні сумматори, елемент Фредкіна.*

OPTIMAL SYNTHESIS OF REVERSIBLE QUANTUM SUMMATORS USING GENETIC ALGORITHM

Vitaly Deibuk, Ion Grytsku

Computer systems and networks department of Chernivtsi National University, 2 Kotsubyns'kogo str., 58012, Chernivtsi, Ukraine, e-mail v.deibuk@chnu.edu.ua, ion_grytsku@mail.ru

Abstract: *The paper suggests a new way of chromosome coding in a genetic algorithm for simulation of reversible one-bit full summators with propagate function in Fredkin basis. The circuits obtained with the use of such an approach demonstrate better delay parameters and better number of inputs/outputs compared with the known analogs. It confirms the effectiveness and applicability of the proposed approach.*

Keywords: *genetic algorithm, evolutionary electronics, reversible full adder, Fredkin gate.*

1. ВСТУП

Зростання ступеня інтеграції сучасних мікроелектронних пристроїв, підвищення їх складності веде до того, що питання затримки, розсіяння потужності та розмірів стають чи не найважливішими цілями комп'ютерної схемотехніки. При цьому мільйони вентилів, які виконують логічні операції в комп'ютерах, є незворотними. Тобто, кожного разу виконання логічної операції приводить до втрати або стирання деякої частини вхідної інформації, яка розсіюється у вигляді теплової енергії. Для незворотної логіки кожен біт втраченої інформації випромінює $kT\ln 2$ Дж теплової енергії, де k – постійна Больцмана, T – абсолютна температура [1]. При кімнатних температурах на гігагерцових частотах сучасних процесорів, що містять сотні тисяч транзисторів, розсіювана енергія наближається до кількох Вт, що є наслідком втрати інформації і веде до виникнення помилок у розрахунках та зменшення часового ресурсу мікросхем. Вирішення цих проблем лежить у площині використання нових революційних технологій, які спроможні кардинально зменшити як споживану потужність, так і розсіяння теплової енергії в комп'ютерних системах.

Вдалою альтернативою в цьому питанні можна вважати використання зворотної логіки, яка останнім часом досить швидко розвивається [2,3], оскільки знаходить застосування у різноманітних областях, таких як квантовий комп'ютинг, нанотехнології, біоінформатика, оптичний комп'ютинг тощо, де, поряд з іншими, важливою умовою є екстремально низьке

розсіяння тепла. Можливість використання зворотних логічних операцій, які не знищують вхідну інформацію, теоретично не веде до розсіяння енергії в системах, що їх реалізують. Зворотними є кола (вентилі), в яких вектор вхідних станів завжди можна відновити з вектора вихідних станів. Розробці цифрових зворотних пристроїв присвячена велика кількість досліджень [2-6], які охоплюють як схемотехнічний, так і фізичний аспекти. Однак на сьогодні проблема знаходження оптимального дизайну таких пристроїв ще не розв'язана з практичної точки зору [2].

Операція додавання є однією з базових операцій, а суматори належать до найбільш фундаментальних компонентів довільного цифрового процесора. Схемотехніка зворотних суматорів з різним типом перенесення вимагає розробки повних однорозрядних зворотних суматорів [7], які у випадку квантового комп'ютингу мають ряд особливостей. Питання оптимального синтезу зворотних повних однорозрядних суматорів для квантових мереж може бути розв'язане з допомогою різних стратегій еволюційної електроніки [8,9].

Генетичні алгоритми належать до адаптивних мета-евристичних алгоритмів пошуку оптимального розв'язку різного роду проблем на основі еволюційної ідеї природної селекції та генетики. Вони використовують інтелектуальний випадковий пошук для вирішення проблеми оптимізації у великих просторах станів за багатьма критеріями. Ідеї використання генетичних алгоритмів до синтезу квантових мереж детально розглянуті в роботі [10] і набули подальшого розвитку. Однак більшість підходів основані на використанні базису квантових примітивів (NOT, CNOT, V, V^+ та ін.), в якому синтезовані деякі комбінаційні зворотні пристрої[9].

У даній роботі запропоновано вдосконалений підхід до синтезу квантових зворотних комбінаційних пристроїв, що ґрунтується на використанні генетичних алгоритмів. Такий підхід до зворотного логічного синтезу пов'язаний з необхідністю врахування кількох додаткових умов, а саме, заборона розгалуження за входом та виходом (теорема про заборону клонування [1]) та заборона обернених зв'язків, тобто подача вихідних сигналів логічних елементів на їх входи. В якості базових елементів при синтезі повного однорозрядного зворотного суматора з функцією транзиту використано елемент Фредкіна.

2. ПОВНИЙ ОДНОРОЗРЯДНИЙ ЗВОРОТНИЙ СУМАТОР

Суматори є одними з основних блоків, які входять до складу більшості обчислювальних пристроїв. Описані вище необхідні зміни у логіці квантових обчислень вимагають відповідних змін у реалізації суматорів як на логічному, так і на фізичному рівні. В якості базисних логічних елементів можна вибрати зворотні функціонально повні елементи Тоффолі, Фредкіна, Переса та ін. [1,9], які схемотехнічно добре відпрацьовані на рівні примітивів.

Розглянемо коротко логіку основних зворотних елементів. Двокубітовий елемент Фейнмана (CNOT-вентиль, контрольоване НЕ) може бути описаний виразом:

$$(A,B) \Rightarrow (P,Q) = (A, A \oplus B).$$

Це означає, що вихідний біт P повторює вхідний (контролюючий) біт A, а на виході Q формується сигнал $A \oplus B$, де додавання для вхідних бітів виконується за модулем 2. Очевидно, що при $A = 0$ на контрольованому виході $Q = B$, а при $A = 1$ на виході $Q = \overline{B}$. Графічне позначення елемента Фейнмана (FG) наведено на Рис. 1.

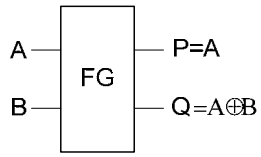


Рис. 1 – Елемент Фейнмана

Трикубітовий елемент Тоффолі (CCNOT – двічі контрольоване НЕ), зображений на Рис. 2, виконує функцію:

$$(A,B,C) \Rightarrow (P,Q,R) = (A,B,AB \oplus C).$$

Цей елемент є універсальним, тобто з його допомогою можна отримати довільну логічну функцію, однак він не зберігає парність ($A \oplus B \oplus C \neq P \oplus Q \oplus R$).

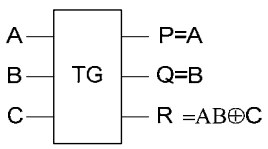


Рис. 2 – Елемент Тоффолі

Трикубітовий зворотний елемент Переса, як видно з Рис. 3, поєднує функції елементів Фейнмана і Тоффолі:

$$(A,B,C) \Rightarrow (P,Q,R) = (A, A \oplus B, AB \oplus C).$$

Хоча він не зберігає парність, однак має найменшу квантову вартість (кількість необхідних для побудови примітивів) серед усіх трикубітових квантових елементів, є універсальним, а тому знайшов широке використання в комбінаційних зворотних схемах [4]. Зокрема, квантова вартість елемента Фейнмана становить 1, Переса – 4, елементів Тоффолі та Фредкіна – 5 [9].

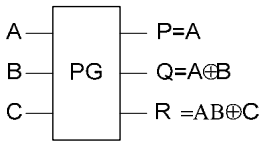


Рис. 3 – Елемент Переса

У роботі [5] нами була проаналізована фізична модель квантового елемента Фредкіна (Рис. 4), який є функціонально повним, зворотним логічним елементом, що зберігає парність, тобто вага за Хеммінгом вхідних сигналів зберігається на виході ($A \oplus B \oplus C = P \oplus Q \oplus R$).

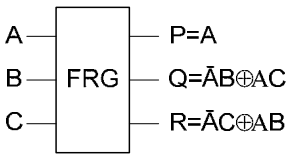


Рис. 4 – Елемент Фредкіна

Трикубітовий елемент Фредкіна (елемент контрольованого обміну) виконує функцію:

$$(A,B,C) \Rightarrow (P,Q,R) = \left(A, \overline{A}B \oplus AC, \overline{A}C \oplus AB\right)$$

Дослідимо деякі схеми суматорів, побудовані на зворотних логічних елементах і оцінимо їх ефективність з точки зору використання для квантового комп'ютингу. Порівняно з класичними схемами у квантових є ряд обмежень. По-перше, у квантових схемах не допускаються «цикли», тобто зворотний зв'язок від однієї частини квантової схеми до іншої, схема має бути «ациклічною». По-друге, заборонена операція FANIN (можливість з'єднувати проводи в один, який містить побітове OR входів), яка є незворотною, а, отже, неунітарною. По-третє, у квантових схемах недопустима й обернена операція FANOUT (розгалуження за виходом) через теорему про заборону клонування квантових станів [1]. Важливими є також умови мінімальної кількості зайвих виходів та постійних входів, мінімальної кількості логічних елементів та мінімальної затримки схеми. Для можливості побудови багаторозрядних суматорів необхідно також, крім суми (S) та перенесення (C), передбачити наявність у однорозрядних суматорів виходу транзиту $P = A \oplus B$. Наявність такого виходу дозволяє збудувати багаторозрядні суматори з пропущеним перенесенням (CSA), які за швидкодією є компромісним варіантом між паралельними суматорами з послідовним (RCA) та прискореним (CLA) перенесенням. Разом з тим CSA вигідно відрізняється від двох останніх апаратною складністю, що й зумовлює підвищений інтерес до них, особливо у квантовому комп'ютингу [3,6,7,11]. У таких суматорах (CSA) зменшення затримки розрахунку перенесення відбувається за рахунок передачі вхідного перенесення C_{i-1} на вихід C_i повного однорозрядного суматора, якщо $P = A \oplus B = 1$, тобто, якщо на один з входів подано 1.

Абстрагуючись від апаратної частини, побудуємо функціонально-логічні схеми повних однорозрядних суматорів на зворотних логічних елементах Фредкіна, відповідно до наведених вище критеріїв. Такі схеми будуть зберігати парність, оскільки їх базовим елементом є елемент Фредкіна.

Повний однорозрядний суматор виконує дві функції:

– додавання:

$$S = A \oplus B \oplus C_{i-1};$$

– формування перенесення в наступний розряд

$$C_i = (A \oplus B)C_{i-1} \vee AB.$$

На базі елемента Фредкіна можна побудувати різноманітні схеми повного однорозрядного зворотного суматора [7,11]. Наведена на Рис. 5 схема синтезована евристичним методом [11] і не містить циклів та розгалужень за виходом елементів, що відповідає описаним вище критеріям до квантових схем і містить вихід транзиту P.

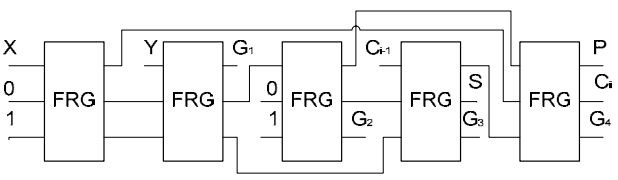


Рис. 5 – Логічна схема повного однорозрядного зворотного суматора з функцією транзиту [11]

Так як квантова ціна вентиля Фредкіна Q_c=5, то повна квантова вартість всієї схеми (c) рівна кількості примітивів, з яких складається схема. У розглядуваному випадку схема складається виключно з вентилів Фредкіна (FRG), а отже $c = 5n$, де n – кількість вентилів Фредкіна. Схема повного зворотного суматора з функцією транзиту (P), зображена на Рис. 5, складається з п'яти елементів Фредкіна, має квантову вартість рівну $c = 25$, кількість постійних входів – 4, зайвих виходів (G_i) – 4, затримка такого суматора становить 5. За вказаними параметрами дана схема на сьогодні вважається кращою. Підійдемо до знаходження оптимального схемотехнічного рішення, використавши еволюційну стратегію пошуку [8,9].

3. ГЕНЕТИЧНИЙ ПОШУК ОПТИМАЛЬНОГО СУМАТОРА

Кодування та генерування початкової популяції

Для пошуку оптимального зворотного суматора за наведеними вище критеріями використано генетичний алгоритм. Генетичні алгоритми відрізняються від традиційних оптимізаційних та пошукових процедур тим, що вони працюють з кодами параметрів, а не з самими параметрами; пошук ведеться за множиною точок, а не за однією точкою; в алгоритмах використовується лише інформація функції пристосованості (цільової функції), а не її похідних або інша додаткова інформація; використовуються ймовірнісні, а не детерміновані правила переходів. У нашому випадку розв'язком задачі генетичного пошуку є оптимізована комбінаційна схема, тому хромосома повинна представлятись у вигляді певного запису даної схеми.

Оскільки хромосома є набором генів, а схема – певне послідовно-паралельне розташування логічних вентилів Фредкіна, а також інформація про вхідні сигнали, то кожен ген представляє собою один послідовний крок обробки сигналів, що може складатися з кількох паралельно розміщених логічних елементів. Таким чином, аналізовані логічні схеми (хромосоми) представимо як набір горизонтальних ліній (пінів), вздовж яких передається інформація і вертикальних секцій, які можуть складатися з кількох логічних елементів, в яких відбувається паралельна обробка інформації, і які відповідають генам. При цьому кожен ген кодується у вигляді масиву пар цілих чисел, де перше число в кортежі позначає порядковий номер відповідного логічного елемента в гені, а друге число – порядковий номер піна даного логічного елемента (Рис. 6).

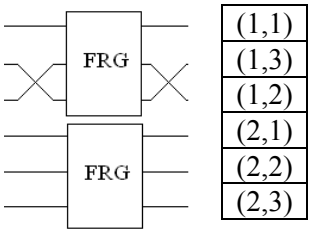


Рис. 6 – Приклад кодування гена

У свою чергу хромосоми, крім наборів зв'язаних між собою генів, складаються також із вектора значень вхідних сигналів. Ці значення записуються в першому елементі хромосоми і також складають окремий ген. Для простоти моделювання комбінаційних схем вважатимемо, що вхідні інформаційні сигнали завжди подаються на верхні лінії, а постійні – на нижні. Приклад представлення хромосоми (схеми) та її кодування подано на Рис. 7. Генерація початкової популяції відбувається методом випадкового створення масиву хромосом. При створенні гена випадковим чином визначається кількість логічних елементів в гені, а також спосіб їх розміщення, при цьому враховуються тип вентилів та їх розмір. Таким чином, на початку роботи ми отримуємо набір різних комбінаційних схем (хромосом).

Моделювання схеми

Знаючи спосіб розміщення елементів у схемі, а також значення постійних входів, робота схеми моделювалася наступним чином. Спочатку генеруються початкові дані (змінні та постійні) і записуються в таблицю значень. Потім почергово подаються початкові значення на вхід гена, а отримані дані подаються як початкові на вхід наступного гена. Це продовжується, поки не змоделюється вся схема. Таким чином, отримаємо таблицю значень вихідних сигналів схеми. Після цього вона поелементно порівнюється з таблицею істинності суматора та підраховується кількість значень, що не збігаються (*Error*).

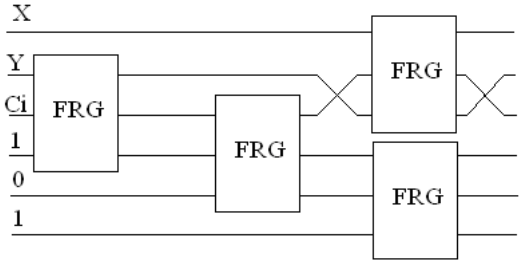


Рис. 7 – Приклад представлення хромосоми

Функція пристосованості (фітнес-функція)

Важливою рисою генетичного алгоритму є вибір функції пристосованості кожної хромосоми. Вона визначає придатність кожної хромосоми з точки зору оцінки помилки. У роботі запропонована зважена функція пристосованості наступного вигляду:

$$f = \alpha\left(\frac{1}{Error+1}\right) + \beta G(c) + \gamma H(g_i) + \delta I(s), \quad (1)$$

де *Error* – кількість помилок у вихідних значеннях модельованої схеми (сума, перенесення та транзит) порівняно зі схемою суматора;

$$G(c) = \exp\left[-\left(1-\frac{5}{c}\right)^2\right] \quad (2)$$

– функція оцінки квантової вартості (c) схеми;

$$H(g_i) = \frac{1}{1+g_i} \quad (3)$$

– функція оцінки кількості постійних входів (надлишкових виходів) (g_i);

$$I(s) = \exp\left[-\left(1-\frac{1}{s}\right)^2\right] \quad (4)$$

– функція оцінки затримки схеми (s)

Вагові коефіцієнти α, β, γ, δ задовольняють умову

$$\alpha + \beta + \gamma + \delta = 1. \quad (5)$$

Пошук оптимальної за вказаними параметрами схеми пов'язаний зі знаходженням максимального значення функції пристосованості *f*. Мінімальна кількість постійних входів шуканої схеми забезпечить мінімальну кількість надлишкових виходів. Величина затримки (s) схеми оцінювалась у відносних одиницях часу затримки одного логічного елемента.

Селекція, схрещування та мутація

У роботі використовувались турнірна селекція та двоточкове схрещування, що відбувається наступним чином. Випадково вибираються дві п'ятірки хромосом, порівнюються їх функції пристосованості в межах кожної п'ятірки і вибираються по одній найкращій хромосомі. Ці дві особини відбираються в батьківську популяцію. Після цього з імовірністю 0,8 відбувається схрещування батьківських хромосом: випадковим чином вибираються дві алелі (але не

Вхідні знач.	Ген1	Ген2	Ген3
X	(0,0)	(0,0)	(1,1)
Y	(1,1)	(0,0)	(1,3)
C_i	(1,2)	(1,1)	(1,2)
1	(1,3)	(1,2)	(2,1)
0	(0,0)	(1,3)	(2,2)
1	(0,0)	(0,0)	(2,3)

рівні між собою) і батьківські хромосоми обмінюються між собою відповідними генами, що знаходяться між цими алелями. Отриманий результат записується в популяцію-нащадок. Мутація здійснюється з імовірністю 0,02 наступним чином: у певній хромосомі вибирається випадковим чином ген і замінюється новоствореним випадкового вигляду.

Завершення алгоритму

У випадку, якщо функція пристосованості деякої хромосоми стане максимально близькою до одиниці, це означатиме, що вона досягла свого максимуму і ми отримали шукану схему. Максимальна кількість ітерацій обмежується.

4. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Алгоритм було використано для синтезу повного однорозрядного зворотного суматора в базисі елементів Фредкіна (таблиці 1–4). Для описаного генетичного алгоритму було використано наступні параметри:

ймовірність схрещування – 0,8;
ймовірність мутації – 0,02;
кількість хромосом в популяцій – 100;
кількість генів у хромосомі – 10;
розмірність гена – 7;
максимальна кількість поколінь – 3000.

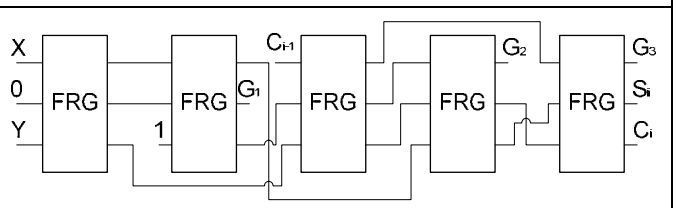
Оскільки ваговий коефіцієнт α визначає внесок функції помилок у загальну фітнес-функцію (1), то для коректної роботи алгоритму (отримання схеми суматора) було прийнято $\alpha \geq 0{,}8$. Залежно від параметрів, за якими потрібно здійснювати оптимізацію схеми суматора, відповідні коефіцієнти β, δ, γ змінювалися при обов'язковому дотриманні умови (5). При багатопараметричній оптимізації одночасно за всіма параметрами вагові коефіцієнти вибиралися з умови $\beta = \delta = \gamma = (1-\alpha)/3$. В середньому, для отримання шуканої хромосоми було використано порядку 200 поколінь.

У таблиці 1 наведено результати моделювання повного однорозрядного зворотного суматора, оптимізованого за кількістю постійних входів (g_i). За наведеною вихідною хромосомою побудовано логічну

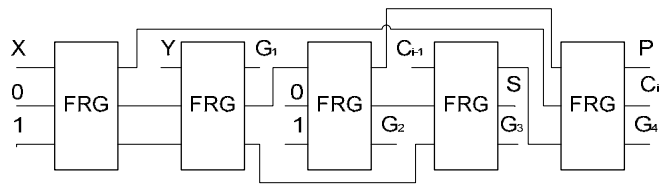
схему, параметри якої – c = 25; g_i = 2; s = 5. Врахування умови наявності у схемі суматора функції транзиту (P) дозволило отримати схему, яка повністю співпала з синтезованою в роботі [11]. Отримана схема та відповідна їй хромосома подані в таблиці 2 (c = 25; g_i = 4; s = 5). Повний однорозрядний зворотний суматор з функцією транзиту, оптимізований за часом затримки (s), та відповідна хромосома (таблиця 3) порівняно з попередньою схемою має в 1,66 рази менший час затримки при тій же апаратній складності і тій же кількості постійних входів та зайвих виходів (c = 25; g_i = 4; s = 3). Моделювання суматора з функцією транзиту при оптимізації за кількістю постійних входів та зайвих виходів (g_i) дозволило отримати схему і відповідну їй хромосому, наведені у таблиці 4 (c = 25; g_i = 3; s = 4). Хоча затримка останньої схеми перевищує затримку попередньої схеми, однак кількість зайвих виходів вдалося зменшити до 3, що у квантовому комп'ютингу є доволі важливо. Останні дві схеми зворотних суматорів з функцією транзиту отримані нами вперше і порівняно з відомими аналогами [3,6,7,11] при тій же апаратній складності мають кращі параметри як за часом затримки (s = 3), так і за кількістю зайвих виходів (постійних входів) g_i=3.

На Рис. 8 наведено графік залежності максимального значення фітнес-функції від номера покоління.

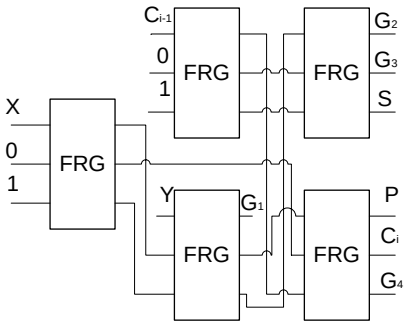
Таблиця 1. Повний однорозрядний зворотний суматор, оптимізований за кількістю постійних входів (c=25; g_i=2; s=5)

Отримана хромосома							Відповідна комбінаційна схема
X	(1,1)	(1,1)	(0,0)	(1,3)	(1,2)	S	
Y	(1,3)	(0,0)	(1,3)	(1,2)	(1,3)	C_i	
C_{i-1}	(0,0)	(0,0)	(1,1)	(0,0)	(1,1)	G_3	
1	(0,0)	(1,3)	(1,2)	(1,1)	(0,0)	G_2	
0	(1,2)	(1,2)	(0,0)	(0,0)	(0,0)	G_1	

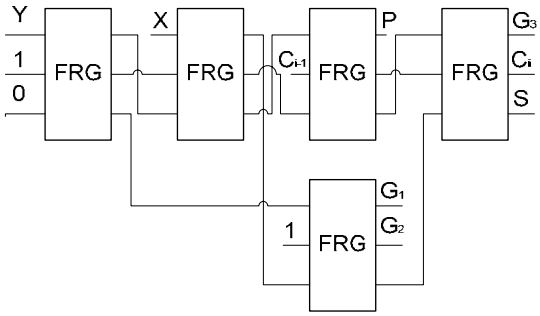
Таблиця 2. Повний однорозрядний зворотний суматор з функцією транзиту(P) (c=25; g_i=4; s=5)

Отримана хромосома							Відповідна комбінаційна схема
X	(1,1)	(0,0)	(0,0)	(0,0)	(1,2)	C_i	
Y	(0,0)	(1,1)	(0,0)	(0,0)	(0,0)	G_1	
C_{i-1}	(0,0)	(0,0)	(0,0)	(1,1)	(1,3)	G_4	
0	(1,2)	(1,2)	(1,1)	(0,0)	(1,1)	P	
1	(1,3)	(1,3)	(0,0)	(1,3)	(0,0)	G_3	
0	(0,0)	(0,0)	(1,2)	(1,2)	(0,0)	S	
1	(0,0)	(0,0)	(1,3)	(0,0)	(0,0)	G_2	

Таблиця 3. Повний однорозрядний зворотний суматор з функцією транзиту(P), оптимізований за часом затримки (c=25; g_i=4; s=3)

Отримана хромосома					Відповідна комбінаційна схема
X	(1,1)	(2,2)	(2,1)	P	
Y	(0,0)	(2,1)	(0,0)	G_1	
C_{i-1}	(0,0)	(1,1)	(2,3)	G_4	
1	(0,0)	(1,3)	(1,3)	S	
1	(1,3)	(2,3)	(1,1)	G_2	
0	(1,2)	(0,0)	(2,2)	C_i	
0	(0,0)	(1,2)	(1,2)	G_3	

Таблиця 4. Повний однорозрядний зворотний суматор з функцією транзиту(P), оптимізований за кількістю постійних входів та надлишкових виходів (c=25; g_i=3; s=4)

Отримана хромосома						Відповідна комбінаційна схема
X	(0,0)	(1,1)	(2,3)	(1,3)	S	
Y	(1,1)	(1,3)	(1,1)	(0,0)	P	
C_{i-1}	(0,0)	(0,0)	(1,2)	(1,2)	C_i	
1	(0,0)	(0,0)	(2,2)	(0,0)	G_2	
0	(1,3)	(0,0)	(2,1)	(0,0)	G_1	
1	(1,2)	(1,2)	(1,3)	(1,1)	G_3	

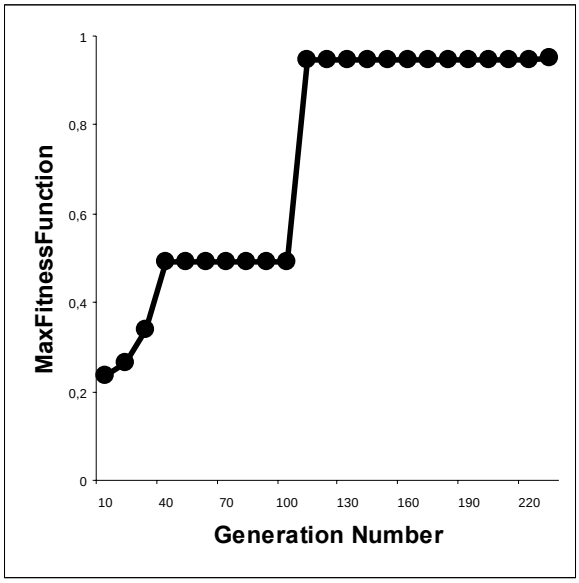


Рис. 8 – Залежність максимального значення фітнес-функції від номера покоління

5. ВИСНОВКИ

У роботі запропоновано новий спосіб кодування хромосом у генетичному алгоритмі для моделювання схем зворотних повних однорозрядних суматорів з функцією транзиту у базисі елементів Фредкіна. Отримані з допомогою такого підходу схеми мають кращі параметри затримки та кількості зайвих виходів(входів) порівняно з відомими аналогами, що демонструє ефективність та застосовність такого підходу.

6. СПИСОК ЛІТЕРАТУРИ

[1] M.A. Nielsen, I.L. Chuang, *Quantum computation and quantum information*. Cambridge University Press, NY, 2001, p. 676.

[2] P. Kerntopf, M. Perkowski, K. Podlaski, Synthesis of reversible circuits: A view on the state-of-the-art, *12 International Conference on Nanotechnology, Birmingham, UK* (20–23 August 2012), pp. 1-6.

[3] Santanu Maity, Bishnu Prasad De, Aditya Kr. Singh, Design and implementation of low-power high-performance carry skip adder, *Int. J. of Engineering and Advanced Technology*, (1) 4 (2012), pp. 212-218.

[4] V. G. Deibuk. I.M. Yuriychuk, R.I. Yuriychuk, Spin model of full summator on Peres gates, *Int. J. Computing*, (11) 3 (2012), pp. 282-292 (in Ukrainian).

[5] G. P. Gorskyi, V. G. Deibuk, Four spins model of universal quantum Fredkin gate, *Informational technologies and computer engineering*, (2) 21 (2011), pp. 56-63 (in Ukrainian).

[6] S. Islam, R. Islam, Minimization of reversible adder circuits, *Asian J. of Information Technology*, (4) 12 (2005), pp. 1146-1151.

[7] P.K. Lala, J.P. Parkerson, P. Chakraborty, Adder designs using reversible logic gates, *WSEAS Transactions on Circuits and Systems*, (9) 6 (2010), pp. 369-378.

[8] R.S. Zebulum, M.C. Pachecco, M.M. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*, CRC Press, 2002, p. 304.

[9] M. Lukac, M. Perkowski et al., Evolutionary approach to quantum and reversible circuits synthesis, *Artificial Intelligence Review*, (20) 3-4 (2003), pp. 361-417.

[10] L. Spector, *Automatic Quantum Computer Programming: A Genetic Programming Approach,* Kluwer Academic Publishers, 2004, p. 153.

[11] J.W. Bruce, M.A.Tornton et al., Efficient Adder Circuits Based on the Conservative Reversible Logic Gates, *Proc. IEEE Comp. Soc. Ann. Symp. on VLSI*, Pittsburgh, PA (April 25–26, 2002), pp. 83-88.

***Віталій Дейбук**, професор, доктор фізико-математичних наук, професор кафедри "Комп'ютерні системи та мережі" Чернівецького національного університету ім. Юрія Федьковича. Стаж науково-педагогічної діяльності у вищій школі 30 років. Автор понад 150 наукових та науково-методичних праць, в тому числі 3 навчальних посібники з грифом Міністерства освіти та науки України.*

Наукові інтереси – комп'ютерне моделювання фізичних властивостей конденсованих систем, проблеми квантової інформатики, дослідження властивостей матеріалів для квантових комп'ютерів

***Іон Грицку**, магістр комп'ютерних системи та мереж, фахівець кафедри системи та мереж Чернівецького національного університету імені Юрія Федьковича. Наукові інтереси – комп'ютерна схемотехніка, квантовий комп'ютинг, генетичні алгоритми та нейронні мережі.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

OPTIMAL SYNTHESIS OF REVERSIBLE QUANTUM SUMMATORS USING GENETIC ALGORITHM

Vitaly Deibuk, Ion Grytsku

Computer systems and networks department of Chernivtsi National University, 2 Kotsubyns'kogo str., 58012, Chernivtsi, Ukraine, e-mail v.deibuk@chnu.edu.ua, ion_grytsku@mail.ru

Abstract: *The paper suggests a new way of chromosome coding in a genetic algorithm for simulation of reversible one-bit full summators with propagate function in Fredkin basis. The circuits obtained with the use of such an approach demonstrate better delay parameters and better number of inputs/outputs compared with the known analogs. It confirms the effectiveness and applicability of the proposed approach.*

Keywords: *genetic algorithm, evolutionary electronics, reversible full adder, Fredkin gate.*

1. INTRODUCTION

Increase of integration in modern microelectronic devices along with their complexity enhancement makes the issues of delay, power dissipation and scale the most important goals of computer design. For irreversible logic computations every bit of lost information generates $kT\ln 2$ J of heat energy, where k – Boltzmann constant, T – absolute temperature [1].

Successful alternative with this respect can be the use of reversible logic [2, 3]. Reversible circuits (gates) are those in which input vector states can always be restored from the output state vector. However, today the problem of finding the optimal design of such devices has not been solved from a practical point of view [2]. Adders are among the most fundamental components of any digital processor.

This paper presents an improved approach to the synthesis of combinational quantum reversible devices, based on the use of genetic algorithms. Such an approach to the reversible logic synthesis is associated with the need to consider several additional conditions, namely, the prohibition of branching for input and output (non-cloning theorem [1]) and the suppression of inverse relations, i.e. logic elements output signals supply on their inputs . As a basic element in the synthesis the full adder with propagate Fredkin gate was used.

2. OPTIMAL SCHEME GENETIC SEARCH

Since the chromosome is a set of genes, and the circuit is some series-parallel arrangement of logic Fredkin gates, let's consider the chromosome representation as a set of horizontal lines (pins), along which information is transmitted and vertical sections, which may consist of several logical elements, implementing information parallel processing, and correspond to genes. While each gene is encoded as an array of pairs of integers, where the first number is the number of logical gate in the gene, and the second number is the index of pin in the given logic element (Fig. 6). In turn chromosomes, except interconnected sets of genes, include a vector of values of the input signals. These values are stored in the first element of the chromosome and also form a separate gene (Fig. 7). The paper proposes a balanced fitness function of the following form:

$$f = \alpha\left(\frac{1}{Error+1}\right)+\beta G(c)+\gamma H(g_i)+\delta I(s), \quad (1)$$

where *Error* – number of errors in the initial values of the simulated circuit (sum, carry or propagate), compared with the adder circuit;

$$G(c) = \exp\left[-\left(1-\frac{5}{c}\right)^2\right] \quad (2)$$

- evaluation function of quantum cost (c) of the circuit;

$$H(g_i) = \frac{1}{1+g_i} \quad (3)$$

- evaluation function of constant inputs (garbage outputs) (g_i);

$$I(s) = \exp\left[-\left(1-\frac{1}{s}\right)^2\right] \quad (4)$$

- evaluation function of circuits delay (s).

Weights α, β, γ, δ satisfy the condition

$$\alpha+\beta+\gamma+\delta = 1. \quad (5)$$

Finding the optimal circuit in accordance with specified parameters is associated with finding the maximum value of fitness function *f*. Minimum constant inputs in the desired circuit will provide the minimum number of garbage outputs.

Latency (s) of circuit was estimated in relative units of delay time of one logic element. We used tournament selection and two-point crossover.

3. RESULTS AND DISCUSSION

The genetic algorithm has been used for the synthesis of reversible full adder based on Fredkin gates (Table 1-4). For the described genetic algorithm the following parameters were used:

probability of crossover – 0.8;
probability of mutation – 0.02;
number of chromosomes in the population – 100;
number of genes in the chromosome – 10;
gene dimension – 7;
maximum number of generations – 3000.

Since the α weighting factor determines the contribution of errors function in general fitness function (1), for the correct algorithm operation (an adder circuit) $\alpha \geq 0{,}8$ was taken. Depending on the parameters needed to optimize the adder circuit, the corresponding coefficients of β, γ, δ were changed while the (5) condition was met. In multiparameter optimization for all parameters simultaneously weights were chosen from the condition $\beta = \delta = \gamma = (1-\alpha)/3$. On average, to get a needed chromosome it took about 200 generations.

Table 1 shows the results of the simulation reversible full adder optimized for the number of garbage outputs (g_i). For given initial chromosome a logic circuit with parameters of $c = 25$; $g_i = 2$; $s = 5$ was constructed. Considering the propagate function (*P*) is present in the circuit adder, we may obtain a circuit, which completely coincides with the one synthesized in [4]. The circuit and the corresponding chromosome are given in Table 2 ($c = 25$; $g_i = 4$; $s = 5$). Reversible full adder with propagate function, optimized for delay time (s). The corresponding chromosome (Table 3) demonstrates 1.66 lower delay time in comparison with the previous circuits for the similar hardware complexity and the same number of constant inputs and garbage outputs ($c = 25$; $g_i = 4$; $s = 3$). Simulation of the full adder with propagate optimized by the number of constant inputs and garbage outputs (g_i) allowed to obtain the circuit and the corresponding chromosome, shown in Table 4 ($c = 25$; $g_i = 3$; $s = 4$). The last circuit delay exceeds the previous circuit delay, whereas the number of garbage outputs was reduced to 3. The last two circuits of reversible adders with propagate received by us for the first time and compared with known analogues [3,4] with the same hardware complexity possess better parameter both of delay ($s = 3$) and the number of garbage outputs (constant inputs) $g_i = 3$. Fig. 8 shows a dependency of the maximum value of the fitness function versus the number of generations.

4. CONCLUSIONS

The paper suggests a new way of chromosome coding in genetic algorithm for simulation of reversible full adder with propagate based on Fredkin gates. The circuits obtained with the use of such an approach demonstrate better delay parameters and the number of garbage outputs compared to the known analogs, pointing out the effectiveness and applicability of the approach proposed.

5. REFERENCES

[1] M.A. Nielsen, I.L. Chuang, *Quantum computation and quantum information.* Cambridge University Press, NY, 2001, p. 676.

[2] P. Kerntopf, M. Perkowski, K. Podlaski, Synthesis of reversible circuits: A view on the state-of-the-art, *12 International Conference on Nanotechnology, Birmingham, UK* (20–23 August 2012), pp. 1-6.

[3] Santanu Maity, Bishnu Prasad De, Aditya Kr. Singh, Design and implementation of low-power high-performance carry skip adder, *Int. J. of Engineering and Advanced Technology*, (1) 4 (2012), pp. 212-218.

[4] J.W. Bruce, M.A. Tornton et al., Efficient Adder Circuits Based on the Conservative Reversible Logic Gates, *Proc. IEEE Comp. Soc. Ann. Symp. on VLSI*, Pittsburgh, PA (April 25–26, 2002), pp. 83-88.

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

INTERACTION BETWEEN LEARNING AND EVOLUTION IN POPULATIONS OF AUTONOMOUS AGENTS

Vladimir G. Red'ko

Scientific Research Institute for System Analysis, Russian Academy of Science,
Vavilova Str., 44/2, Moscow, 119333, Russia,
vgredko@gmail.com, http://www.niisi.ru/iont/staff/rvg/index_eng.php

Abstract: *The model of interaction between learning and evolution for the evolving population of modeled organisms is designed and investigated. The mechanism of genetic assimilation of the acquired features during the numerous generations of Darwinian evolution is studied. The mechanism of influence of the learning load is analyzed. It is showed that the learning load leads to a significant acceleration of an evolution. The hiding effect is also studied. This effect means that a strong learning inhibits the evolutionary search in some situations.*

Keywords: *Speed and efficiency of evolutionary search, Baldwin effect, genetic assimilation.*

1. INTRODUCTION

In the XIX century, the concepts, suggesting that interaction between learning and evolutionary processes is possible, appeared [1-3]. According to these concepts, learning can contribute significantly to an evolutionary process. This type of influence of learning (or other acquisition of useful features during the life of the organism) on the evolutionary process is often called the Baldwin effect [1]. According to this effect initially acquired features can become inherited during a number of generations. The evolutionary "re-invention" of useful features, initially obtained by means of learning, is often called genetic assimilation [4].

Number of works attempted to model and to analyze interactions between learning and evolution by means of computer simulations [5-10]. In particular, Hinton and Nowlan demonstrated that learning can learning can guide an evolutionary process to find the optimum [7]. Mayley investigated different aspects of interaction between learning and evolution [8] and demonstrated that the hiding effect can take place if the learning is sufficiently strong. The hiding effect means that if learning increases the chances of finding a good phenotype independently on the genome, then learning can also inhibit the evolutionary optimization and genetic assimilation. In addition, the learning load (the cost of learning) was taken into account in [8]. The learning load means that the process of learning has an additional load for the organism and fitness of the organism is reduced under the influence of the load.

The interaction between learning and evolutionary optimization of a neural network control system of autonomous agents was modeled in [10]. The genetic assimilation of the acquired features of agent was observed during several generations of evolution. It was demonstrated that learning can significantly accelerate a process of evolutionary optimization. However, it was difficult to analyze the detailed mechanism of interaction between learning and evolution in these models [10], because these mechanisms were «hidden» in the dynamics of numerous synapse weights of neural networks of agents.

This article develops the mentioned works; it uses works [7, 8] as background. In contrast to [7, 8], the current work uses one of the most clear evolution model, namely, the quasispecies model, proposed by Eigen [11, 12], and quantitative estimations of evolutionary rate and the effectiveness of evolutionary algorithms, obtained in [13, 14]. The quasispecies model considers the process of evolution that is based on the selection and mutations of the genomes of organisms (without crossovers) and describes the main properties of the evolutionary process. The use of models and methods of works [11-14] allows getting a better understanding of the mechanisms of interaction between learning and evolution. In particular, our approach allows analyzing quantitatively the mechanism of genetic assimilation.

2. DESCRIPTION OF THE MODEL

The evolving population of modeled organisms (or individuals) is considered. Similar to [7] we assume that there is a strong correlation between the genotype and the phenotype of the modeled organisms. We assume that the genotype (or the genome) and the phenotype of the organism have the same form, namely, they are chains; symbols of both chains are equal to 0 or 1. The length of these chains is equal to N. For example, similar to [7], we can assume that the genome encodes a model chain DNA, «letters» of which are equal to 0 or 1, and the phenotype is determined by the neural network of organisms, the synaptic weights of the neural network are equal to 0 or 1 too. The initial synaptic weights, received at birth of the organism, are determined by the genome (more precisely, the synaptic weights are equal to the genome symbols). These synaptic weights are changed by means of learning during organism's life.

Similar to the quasispecies model, we assume that each organism has its own genome $\mathbf{S_0}$. The population consists of n organisms, organism's genomes are equal to $\mathbf{S_{0}}_k$, $k = 1,..., n$. The organism genome $\mathbf{S_{0}}_k$ is a chain of symbols, S_{0ki}, $i = 1,..., N$. We also assume that the length of the chains N and the number of organism in population n are large: N, $n >> 1$. The values N and n don't change in the course of evolution. Symbols S_{0ki} are equal to 0 or 1. We assume that N is so large that only a small part of possible 2^N genomes can be presented in a particular population: $2^N >> n$. Typical values of N and n in our computer simulations are as follows: $N \sim n \sim 100$.

The evolutionary process consists of a sequence of generations. The new generation is obtained from the old one by selection and mutations. Genomes of organisms of the initial generation are random.

In order to consider learning processes, we introduce two types of sequences: 1) genomes or initial sequence $\mathbf{S_{0}}_k$ that is received by the organism at its birth, and 2) the current sequence of the organism $\mathbf{S_T}_k$.

Organisms inherit the genomes $\mathbf{S_{0}}_k$ from their parents, these genomes don't change during the organism life and are transmitted (with small mutations) to their descendants. Mutations are random changes of symbols S_{0ki}. The organism receives the genome at its birth, the current sequence $\mathbf{S_T}_k$ at the birth time moment is equal to the genome: $\mathbf{S_T}_k(t = 1) = \mathbf{S_{0}}_k$. The life time of any organism is equal to T. The time is discrete: $t = 1,...,T$. The duration of the generation is equal to T. The sequence of $\mathbf{S_T}_k$ is modified during the organism life by means of learning. The current sequence $\mathbf{S_T}_k$ determines the organism's phenotype.

As descendants of organisms obtain just genomes $\mathbf{S_{0}}_k$ that organisms received from their parents and not sequences $\mathbf{S_T}_k$ that are optimized by learning, the evolutionary process has the Darwinian character.

It is assumed that there is an optimal sequence of $\mathbf{S}_m$ (components of which are also equal to 0 or 1), which is searched for in the processes of evolution and learning. At computer simulation, the sequence $\mathbf{S}_m$ was set to be the random one.

Learning is performed by the following method of trial and error. Every time moment t each symbol of the sequence $\mathbf{S_T}_k$ is randomly changed to 0 or 1, and if this new symbol coincides with the corresponding symbol of the optimal sequence $\boldsymbol{S}_m$, then this symbol is fixed in the $\mathbf{S_T}_k$; otherwise, the old symbol of the sequence $\mathbf{S_T}_k$ is restored. So, during learning, the current sequence $\mathbf{S_T}_k$ moves towards the optimal sequence $\mathbf{S}_m$.

It should be noted that if we consider symbols of sequences $\mathbf{S_T}_k$ as synaptic weights of the neural network, then the learning process has a simple meaning: learning is searching for optimal weights of the synapses.

At the end of the generation, the selection of individuals in accordance with their fitness takes place. The fitness is determined by the sequence $\mathbf{S_T}_k$ at the time moment $t = T$. We denote this sequence of $\mathbf{S_F}_k$, i.e. we set $\mathbf{S_F}_k = \mathbf{S_T}_k(t = T)$. The fitness of the organism $\mathbf{S}_k$ is determined by the Hamming distance $\rho = \rho(\mathbf{S_F}_k,\mathbf{S}_m)$ between the sequences $\mathbf{S_F}_k$ and $\mathbf{S}_m$:

$$f(\mathbf{S}_k) = \exp[-\beta\rho(\mathbf{S_F}_k,\mathbf{S}_m)] + \varepsilon\,, \qquad (1)$$

where β is the positive parameter, that characterizes the intensity of selection, $0 < \varepsilon << 1$. The role of the parameter ε in (1) can be considered as the influence of random factors of the environment on the fitness of individuals.

The selection of the individuals into a new generation is made by the well-known method of the fitness proportionate selection (or the roulette wheel selection). In this method individuals are selected into a new generation probabilistically. The choice of an individual into the next generation takes place n times, so the number of individuals in the population in all generations is equal to n. At any choice, the probability of the selection of a particular individual is proportional to its fitness.

Thus, individuals are selected at the end of a generation in accordance with their phenotype codes $\mathbf{S_F}_k = \mathbf{S_T}_k(t = T)$, i.e. in accordance with the final result of learning, whereas initial genomes $\mathbf{S_{0}}_k$ (modified by small mutations) are transmitted from parents to descendants.

In addition, similar to the work [8], we take into account the learning load (or the cost of learning), namely, we consider the fact that the learning

process has a certain burden on the individual and fitness of the individual may be reduced under the influence of the load. For this purpose we use the modified fitness of individuals:

$$f_m(\mathbf{S}_k) = \exp(-\alpha d)\ \{\exp[-\beta\rho(\mathbf{S}_{Fk},\mathbf{S}_m)] + \varepsilon\}\ , \qquad (2)$$

where α is the positive parameter, which takes into account the learning load, $d = \rho(\mathbf{S}_{0k},\mathbf{S}_{Fk})$ is the Hamming distance between the initial $\mathbf{S}_{0k}$ and final sequence $\mathbf{S}_{Fk}$ of the individual, i.e. the value that characterizes the intensity of the whole learning process of the individual during its life. The factor $\exp(-\alpha d)$ decreases the fitness of an individual, this decrease clearly depend on the change of the current sequence $\mathbf{S}_{Tk}$ at the learning process.

It should be noted that since initial sequences $\mathbf{S}_{0k}$ of the individuals in the initial population are random, the average Hamming distance between these sequences and the optimal one $\mathbf{S}_m$ is approximately equal to $N/2$. The sequences $\mathbf{S}_k$ should overcome this distance by means of learning and evolution in order to reach $\mathbf{S}_m$.

3. RESULTS OF SIMULATION

3.1. SCHEME AND PARAMETERS OF SIMULATION

Two modes of operation of the model are consider below: 1) evolution combined with learning, as described above, 2) "pure evolution", that is evolution without learning, in this case, the learning doesn't take place and it is believed that $\mathbf{S}_{Tk} = \mathbf{S}_{0k}$. In addition, the influence of the learning load is analyzed; in this case, the fitness of an individual is calculated according to (2). Analysis of the model was carried out by means of computer simulation.

The parameters of the model at simulation are chosen in such manner that the evolutionary search is effective; the experience of the work [13] for the case of pure evolution is used at this choice. The fitness of the individuals in [13] was determined analogously to the expression (1), only the influence of random factors wasn't taken into account (formally this means that the value ε was equal to 0).

The choice of parameters of simulation is as follows. We believe that the length of the chain is rather large: $N = 100$. We also set $\beta = 1$, this corresponds to a sufficiently high intensity of selection, so the selection time is small, the time of evolutionary search is determined mainly by mutations. The intensity of mutations must not be too large; in order to remove the possibility of mutation losses of already found good individuals. However, the intensity of mutations must not be too small, in order to ensure sufficiently large efficiency of mutational search during evolutionary optimization. Taking this into account, we believe that the probability to substitute any symbol in the sequence $\mathbf{S}_{0k}$ at mutations in one generation is $p_m = N^{-1} = 0.01$. At this mutation intensity p_m approximately one symbol in the genome of any individual is replaced at one generation, i.e. during one generation of the Hamming distance ρ between sequences of individuals $\mathbf{S}_k$ in population and the optimal sequence $\mathbf{S}_m$ changes on average by 1 by means of mutations. Selection leads to a decrease of this distance ρ. Since the intensity of selection is large, and the Hamming distance between sequences of the initial population and the optimal sequence $\mathbf{S}_m$ is of the order of N, the whole process of evolution will take approximately $G_T \sim N$ generations. Such an estimation of the rate of evolution is true, if the population size is large enough and the fluctuation effects and neutral selection of individuals (that is selection independent on fitness of individuals) can be neglected. To satisfy this condition, it is enough to require that the characteristic time of the neutral selection (which is of the order of the population size n [13, 15]), should be greater or of the order of G_T, so we believe that $n = G_T = N$.

Thus, the parameters of simulation in accordance with the experience of the work [13] are chosen as follows: $N = 100$, $\beta = 1$, $p_m = N^{-1} = 0.01$, $n = G_T = N = 100$.

In the current model we also believe that the probability of a random replacement of the symbols during learning p_l is rather large: $p_l \sim 1$, the number of time moments during the generation T is equal to 2 (choice of such parameters p_l and T means that learning is rather fast), the parameter ε is small: $\varepsilon = 10^{-6}$.

The results of simulation are averaged over 1000 or 10000 calculations corresponding to different random number generators. The results of simulation are described below.

3.2. COMPARISON OF REGIMES OF PURE EVOLUTION AND EVOLUTION COMBINED WITH LEARNING

Fig. 1 shows the dependence of the average of the Hamming distance $\rho = \rho(\mathbf{S}_k,\mathbf{S}_m)$ between the sequences $\mathbf{S}_k$ of the individuals in the population and the optimal sequence $\mathbf{S}_m$ at the beginning of generations (i.e. in this case $\rho(\mathbf{S}_k,\mathbf{S}_m) = \rho(\mathbf{S}_{0k},\mathbf{S}_m)$) on the generation number G . Curve 1 characterizes the regime of evolution combined with learning; curve 2 characterizes the regime of pure evolution. The dependences are averaged for all individuals of population and for 1000 calculations. Fitness of individuals is determined by the expression (1). We can see that pure evolution without learning (curve 2) doesn't optimize individuals $\mathbf{S}_k$ at all even at

small values ε; whereas evolution combined with learning (curve 1) obviously ensures the movement towards the optimal individual $\mathbf{S}_m$.

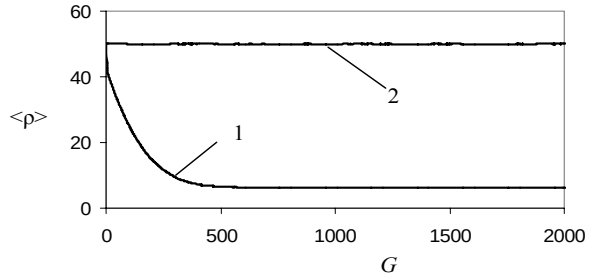


Fig. 1 – The dependence of <ρ> on the generation number G. Curve 1 characterizes the regime of evolution combined with learning; curve 2 characterizes the regime of pure evolution

To understand, why the pure evolution doesn't ensure a decrease the value ρ, let's estimate the value of fitness (1) in the original population. Individuals of the initial population $\mathbf{S}_{0k}$ are far from the optimal one $\mathbf{S}_m$: the Hamming distance $\rho = \rho(\mathbf{S}_{0k},\mathbf{S}_m)$ is of the order of $N/2 = 50$, therefore, $\exp(-\rho) \sim 10^{-22}$ and $\exp(-\rho) << \varepsilon$. This means that all the individuals of the population have approximately the same value of the fitness $f(\mathbf{S}_k) \approx \varepsilon$. Consequently, a selection of the individuals doesn't occur in the case of the pure evolution. Thus, the movement towards $\mathbf{S}_m$ occurs only in the presence of learning; this movement leads to the decrease of ρ. A similar influence of learning on evolutionary optimization (though in another context) was described in the work [7].

Let's consider the effect of the acceleration of the evolutionary process by learning (curve 1 in Fig. 1). Analysis of the results of simulations shows that the gradual decrease in the values $\rho = \rho(\mathbf{S}_k,\mathbf{S}_m)$ occurs as follows. During learning, the distribution of individuals $n(\rho)$ on the value ρ in the population is shifted towards smaller ρ, so the values $\rho = \rho(\mathbf{S}_{Fk},\mathbf{S}_m)$ becomes small enough, then $\exp(-\rho(\mathbf{S}_{Fk},\mathbf{S}_m))$ is of the order of ε. Consequently, different individuals in the population in accordance with (1) have different fitness; so individuals with small values $\rho(\mathbf{S}_{Fk},\mathbf{S}_m)$ are selected into the population of the next generation. It is intuitively clear that the genomes of $\mathbf{S}_{0k}$ of selected individuals should be rather close to the final sequences $\mathbf{S}_{Fk}$ (obtained as a result of the learning) of these individuals. Thus, the result of selection is the selection of individuals, which genomes are also moving to the optimal sequence $\mathbf{S}_m$. Therefore, the value ρ in the new population decreases.

The described mechanism is characterized by Fig. 2, which shows the distribution the $n(\rho)$ on ρ in the population at different moments of the first generation. Curve 1 shows the distribution of the $\rho = \rho(\mathbf{S}_{0k},\mathbf{S}_m)$ for the genomes of individuals at the beginning of the generation. Curve 2 shows the distribution of the $\rho = \rho(\mathbf{S}_{Fk},\mathbf{S}_m)$ for individuals after the learning, but before selection. Curve 3 shows the distribution of the $\rho = \rho(\mathbf{S}_{Fk},\mathbf{S}_m)$ for individuals, selected in accordance with the fitness (1). Curve 4 shows the distribution of the $\rho = \rho(\mathbf{S}_{0k},\mathbf{S}_m)$ for the genomes of selected individuals at the end of the generation. The genomes of selected individuals $\mathbf{S}_{0k}$ are sufficiently close to the sequences of trained and selected individuals $\mathbf{S}_{Fk}$, therefore the distribution of the $\rho = \rho(\mathbf{S}_{0k},\mathbf{S}_m)$ for genomes (curve 4) moves towards the distribution for finite sequences $\mathbf{S}_{fk}$ (curve 3). Finally, after the selection at the end of the generation, the distribution of the genomes of ρ (curve 4) is formed; this distribution is closer to the distribution, which is represented by curve 3, than the initial distribution of genomes (curve 1). Similar displacement of the distribution of $n(\rho)$ towards smaller values ρ takes place in the next generations.

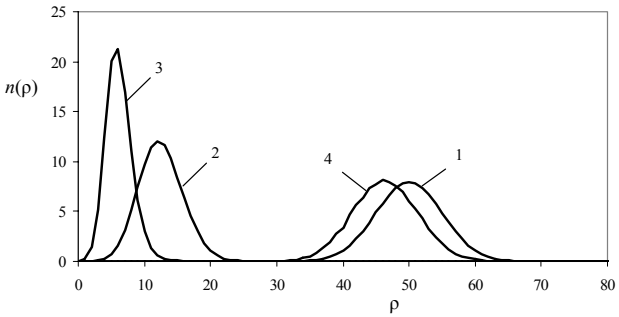


Fig. 2 – The distribution n(ρ) in the first generation of evolution: curve 1 is the distribution of ρ = ρ(S0k,Sm) for the original genomes before learning, curve 2 is the distribution of ρ = ρ(SFk,Sm) for individuals after the learning, but before the selection, curve 3 is the distribution of ρ = ρ(SFk,Sm) for selected individuals, curve 4 is the distribution of ρ = ρ(S0k,Sm) for the genomes of selected individuals at the end of the generation (results are averaged on 10,000 calculations)

Such displacement reveals the mechanism of reduction of <ρ> in the presence of learning: the selection leads to the genomes of individuals $\mathbf{S}_{0k}$, which are closer to sequences of learned and selected individuals $\mathbf{S}_{Fk}$, than the initial genomes of individuals at the beginning of the generation. As a result, a transition from curve 1 to curve 4, i.e. the decrease in the values ρ, takes place during a generation.

It should be underlined that the decrease of values ρ at learning should be sufficiently large in order to ensure small role of the parameter ε and significant difference of the fitness (1) of individuals, and therefore, the effective selection of individuals with small values $\rho(\mathbf{S}_{Fk},\mathbf{S}_m)$. This selection corresponds to the essential decrease of the values of ρ at transition from curve 2 to curve 3 in Fig. 2. It is clear that in order to guarantee the effective operation this mechanism, learning should

be enough strong. The other role of a strong learning is characterized in the next subsection.

The described results show that learning can lead to the effective genetic assimilation and to the radical acceleration of the evolutionary search.

3.3. HIDING EFFECT

However, a strong learning can not only accelerate the evolutionary search, but it can prevent to find the optimal genome. Curve 1 in Fig. 1 shows that at large G, the decrease of $<\rho> = <\rho(\mathbf{S}_{0k},\mathbf{S}_m)>$ is limited: the final value of the $<\rho>$ remains quite large, the asymptotic value of the $<\rho>$ is approximately equal to 6.2. This is due to the fact that at large G (G ~1000) the strong learning ($p_l = 1$, $T = 2$) results in finding the optimal sequence $\mathbf{S}_m$ independently on the genome $\mathbf{S}_{0k}$. Therefore, at the final stages of evolution the genomes $\mathbf{S}_{0k}$ don't move towards the optimum $\mathbf{S}_m$. So, the hiding effect [8] is observed.

Thus, the mechanism of the hiding effect is analyzed. This effect means that the strong leaning prevents the evolutionary optimization, because it increases the chances of finding a good phenotype independently on the genome of the individual.

3.4. INFLUENCE OF LEARNING LOAD ON MODELED PROCESSES

We also analyzed the influence of the learning load on the modeled processes. For this case, fitness is determined by the expression (2). The simulation is performed for the mentioned parameters ($N = n = 100$, $\beta = 1$, $p_m = 0.01$, $p_l = 1$, $T = 2$, $\varepsilon = 10^{-6}$), the value α is equal to 1. The simulation results are represented by Fig. 3, 4. Fig. 3 shows the dependence of the average distance $<\rho>$ between sequences $\mathbf{S}_k$ and the optimal sequence of $\mathbf{S}_m$ on the generation number G. Fig. 4 shows the dynamics of the distribution $n(\rho)$ of values ρ at different moments of the first generation of the evolutionary process.

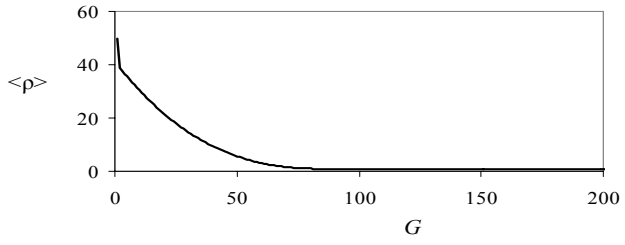


Fig. 3 – The dependence of the value <ρ> on the generation number G; the fitness of individuals is determined by the expression (2); the decrease of values <ρ> is much faster than that of in Fig. 1 (results are averaged on 1000 calculations)

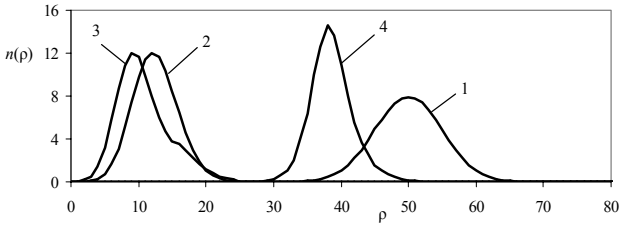


Fig. 4 – The distribution n(ρ) in the first generation of evolution; the fitness of individuals is determined by the expression (2): curve 1 is the distribution of ρ = ρ(S0k,Sm) for the original genomes before learning, curve 2 is the distribution of ρ = ρ(SFk,Sm) for individuals after the learning, but before the selection, curve 3 is the distribution of ρ = ρ(SFk,Sm) for selected individuals, curve 4 is the distribution of ρ = ρ(S0k,Sm) for the genomes of selected individuals at the end of the generation; the displacement of the distributions to smaller values ρ is significantly faster than in Fig. 2 (results are averaged on 10,000 calculations)

It should be underlined that the genetic assimilation for cases of the fitness, determined by the expression (1) and the expression (2), has the same nature. In both cases, the genomes of selected individuals $\mathbf{S}_{0k}$ approach to sequences $\mathbf{S}_{Fk}$ of trained and selected individuals. That is in both Fig. 2 and Fig. 4 the curve 4 moves towards the curve 3. A significant difference consists only in the fact that the learning load makes this movement more evident and more effective. Thus, the learning load leads to more effective selection of individuals with the genomes of $\mathbf{S}_{0k}$, which are close to $\mathbf{S}_m$, and the evolution process is significantly accelerated.

4. CONCLUSION

Thus, the model of interaction between learning and evolutionary optimization has been constructed and investigated.

The mechanism of the genetic assimilation is studied in details. The genetic assimilation can be described as follows: 1) learning and selection shift the distribution of individuals towards the optimum; 2) genomes of selected individuals also move towards the optimum. The mechanism of the genetic assimilation is illustrated by Fig. 2. It is shown that the genetic assimilation can lead to a radical acceleration of evolutionary processes.

The mechanism of the hiding effect is analyzed. This effect means that strong learning inhibits the evolutionary search of the optimal sequence, if this learning increases the chances of finding a good phenotype regardless of the genome of the individual.

The influence of the learning load on the evolutionary processes is studied. It is shown that the learning load leads to effective genetic

assimilation and to a considerable acceleration of evolution.

Future plans of our research include investigations of cognitive features of autonomous agents. These agents can be optimized by means of learning and evolution.

5. ACKNOWLEDGMENTS

This work is partially supported by the Russian Foundation for Basic Research, Grant No 13-01-00399. The author thanks the anonymous reviewer for useful comments and recommendations.

6. REFERENCES

[1] J.M. Baldwin, A new factor in evolution, *American Naturalis*t, (30) (1896), pp. 441-451.

[2] C.L. Morgan, On modification and variation, *Science*, (4) (1896), pp 733-740.

[3] H.F. Osborn, Ontogenetic and phylogenetic variation, *Science*, (4) (1896), pp. 786-789.

[4] C.H. Waddington, Canalization of development and inheritance of acquired characters, *Nature*, (150) (1942), pp. 563-565.

[5] *Adaptive Organisms in Evolving Populations: Models and Algorithms.* Eds. Belew R.K. and Mitchell M. Massachusetts: Addison-Wesley, 1996.

[6] *Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect*. Eds. Turney P., Whitley D., Anderson R. Special Issue of Evolutionary Computation on the Baldwin Effect, (4) 3 (1996).

[7] G.E. Hinton, S.J. Nowlan, How learning can guide evolution, *Complex Systems*, (1) (1987), pp. 495-502.

[8] G. Mayley, Guiding or hiding: Explorations into the effects of learning on the rate of evolution, In: *Proceedings of the Fourth European Conference on Artificial Life (ECAL 97)*. Eds. Husbands P. and Harvey I. Cambridge, Massachusetts: MIT Press,1997, pp. 135-144.

[9] D. Ackley, M. Littman, Interactions between learning and evolution, In: *Artificial Life II: Proceedings of the Second Artificial Life Workshop*. Eds. Langton C. G., Taylor C., Farmer J. D., Rasmussen S. Redwood City CA: Addison-Wesley, 1992, pp. 487-509.

[10] V.G. Red'ko, O.P. Mosalov, D.V. Prokhorov, A model of evolution and learning, *Neural Networks*, (18) 5-6 (2005), pp. 738-745.

[11] M. Eigen, Selforganization of matter and the evolution of biological macromolecules, *Naturwissenschaften*, (58) 10 (1971), pp. 465-523.

[12] M. Eigen, P. Schuster, *The hypercycle: A principle of natural self-organization*. Springer Verlag: Berlin etc, 1979.

[13] V.G. Red'ko, Yu.R. Tsoy, Estimation of the efficiency of evolution algorithms, *Doklady Mathematics,* (72) 2 (2005), pp. 810-813.

[14] V.G. Red'ko, Yu.R. Tsoy, Efficiency of evolutionary search in quasispecies model, *Fuzzy Systems and Soft Computing*, (1) 1 (2006).

[15] M. Kimura, *The Neutral Theory of Molecular Evolution*, Cambridge University Press, 1983.

Red'ko Vladimir Georgievich, *Deputy Director for Research of Center of Optical Neural Technologies, Scientific Research Institute for System Analysis, Russian Academy of Sciences. V.G. Red'ko graduated from the Moscow Institute of Physics and Technology in 1971. He is the doctor of sciences (physics and mathematics). He is the author of more than 150 scientific publications, including two monographs. His scientific interests includes: the problem of origin of human intelligence, cognitive evolution, models of adaptive behavior, neuroinformatics.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

ПРОПУСКНА СПРОМОЖНІСТЬ ТРАНСПОРТНОГО РІВНЯ БЕЗПРОВІДНИХ ЛОКАЛЬНИХ МЕРЕЖ IEEE 802.11g ЩО ФУНКЦІОНУЮТЬ В ІНФРАСТРУКТУРНОМУ РЕЖИМІ

Віктор Чернега

Севастопольський національний технічний університет,
вул. Університетська, 33, м. Севастополь 99053 Україна
v_chernega@rambler.ru

Резюме: *Детально проаналізована процедура обміну кадрами між клієнтськими комп'ютерами і точкою доступу безпровідної локальної комп'ютерної мережі стандарту 802.11g і отримані вирази, що дозволяють розрахувати потенційну пропускну спроможність такої мережі.*

Ключові слова: *безпровідні локальні мережі 802.11g, інфраструктурний режим, пропускна спроможність, транспортний рівень.*

PERFORMANCE OF A TRANSPORT LEVEL OF WLANS IEEE 802.11g FUNCTIONING IN INFRASTRUCTURAL MODE

Victor Chernega

Sevastopol National Technical University
33, University Str., Sevastopol 99053, Ukraine
v_chernega@rambler.ru

Abstract: *The procedure of frames exchange between client computers and a wireless access point based on 802.11g standard is analyzed in details. We have obtained the expressions which allow calculate the potential bandwidth of such wireless network.*

Keywords: *WLANS of 802.11g, TCP, UDP, infrastructural mode, carrying capacity.*

ВСТУП

Проблемам теоретичного розрахунку і виміру реальної пропускної спроможності у безпровідних локальних комп'ютерних мережах присвячено ряд публікацій. Проте в цих роботах визначається пропускна спроможність лише на канальному рівні для локальних мереж IEEE 802.11g [1], що функціонують у незалежному (Ad Hoc) режимі, або на транспортному рівні лише для мереж стандарту IEEE 801.11b [2], формати кадрів у яких істотно відрізняються від мереж, що виконані за стандартом IEEE 802.11g. Крім того, ні у базовому стандарті IEEE 802.11, ні в літературі не розглянуті часові діаграми обміну кадрами між клієнтськими станціями і точкою доступу при організації сеансів зв'язку в мережі на транспортному рівні. Більшість авторів публікацій з даної проблеми обмежуються ілюстрацією обміну лише між клієнтськими станціями в Ad Hoc-сетях, а деякі навіть помилково трактують передачу кадрів через точку доступу [3, рис. 2.12].

Метою даної роботи є детальне розкриття процедури обміну кадрами між клієнтськими комп'ютерами і базовою станцією у безпровідних локальних мережах WiFi при передачі даних по протоколах транспортного рівня TCP і UDP для отримання виразів, що дозволяють оцінити максимально можливу швидкість передачі корисної інформації між користувачами мережі IEEE 802.11g як у базовому режимі, так і наявності прихованих станцій за відсутності колізій і перешкод у каналі зв'язку. При цьому

передбачається, що мережа працює в інфраструктурному режимі, при якому обмін кадрами в мережі здійснюється через точку доступу AP (Access Point).

Пропускна спроможність (ефективна швидкість передачі даних) віртуального каналу між двома клієнтськими станціями мережі, створеного у процесі встановлення з'єднання на транспортному рівні, обчислюється кількістю корисної інформації у бітах N_B, що видається одержувачеві за час сеансу зв'язку в секундах T_S,

$$V_{eff} = N_B / T_S. \quad (1)$$

Процедура сеансу зв'язку по протоколу TCP між двома клієнтськими станціями STA-1 і STA-2 через точку доступу AP включає фазу встановлення з'єднання між джерелом і одержувачем, фазу передачі даних, фази закриття і роз'єднання [2, 3]. З урахуванням цього час сеансу зв'язку представимо у наступному вигляді:

$$T_S = T_C + T_{DT\text{-}TCP} + T_{FIN}, \quad (2)$$

де T_C −час встановлення з'єднання; $T_{DT\text{-}TCP}$ − час фази передачі даних; T_{FIN} − час фази завершення та роз'єднання.

Для визначення тривалості фаз з'єднання, передачі даних і роз'єднання побудуємо часові діаграми сеансу зв'язку між двома клієнтськими комп'ютерами. З цією метою був проведений детальний аналіз процесу обміну кадрами в безпровідній локальній мережі, що працює в інфраструктурному режимі, за допомогою програми моніторингу та аналізу мережевих пакетів CommView for WiFi.

Часова діаграма обміну кадрами і пакетами на канальному і транспортному рівнях у процесі встановлення TCP-з'єднання зображена на рис.1. Для розрізнення блоків повідомлень, що передаються на канальному рівні від блоків, передачі на транспортному рівні, на діаграмі використані відповідно назви "Кадр − Frame" і "Пакет − Packet".

Клієнтська станція STA-1, виявивши, що канал зв'язку вільний, витримує обов'язкову паузу T_{DIFS}, після закінчення якої запускається генератор випадкових чисел і обчислюється час зворотного відліку (*Backoff*) T_{BO}. Якщо канал зв'язку після закінчення інтервалу T_{BO} залишається вільним, то станція STA-1 формує TCP-сегмент зі встановленим прапором синхронізації SYN1:1 і нульовим полем даних. Цей сегмент на мережевому рівні доповнюється IP-заголовком, в якому вказуються IP-адреси станцій призначення STA-2 і відправника STA-1. Сформований пакет інкапсулюється в кадр канального рівня, у якому містяться MAC-адреса точки доступу (BSSID), одержувача STA-2 і джерела STA-1. Точка доступу приймає TCP-пакет з бітом синхронізації і, за відсутності в ньому помилок, після витримки обов'язкової короткої міжкадрової паузи SIFS, відправляє станції STA-1 кадр підтвердження на канальному рівні (кадр MAC-ACK).

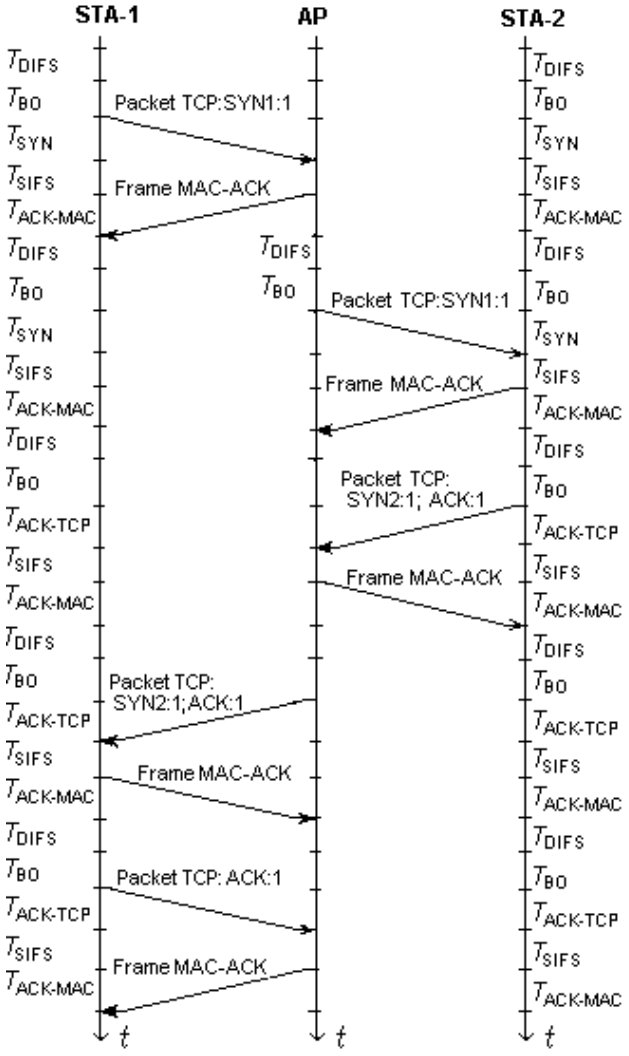


Рис.1 − Часова діаграма обміну кадрами і пакетами між клієнтськими станціями і точкою доступу в процесі встановлення TCP-з'єднання

На наступному етапі точка доступу, нарівні з клієнтськими станціями мережі, включається у конкурентну боротьбу за отримання доступу до середовища. Для цього вона відлічує обов'язкові інтервали T_{DIFS} та T_{BO} і, за відсутності несучого сигналу в каналі, відправляє станції призначення пакет запиту на встановлення з'єднання (пакет TCP: SYN1:1).

STA-2 підтверджує правильність прийому кадру на канальному рівні і після конкурентної боротьби за канал відправляє пакет зі

встановленими у заголовку TCP прапорами підтвердження встановлення з'єднання (прапор ACK встановлений в 1) і запиту на встановлення з'єднання з боку станції 2 (прапор SYN2:1). Потім аналогічним чином пакет TCP:SYN2:1; ACK:1 доставляється станції STA-1. Підтвердження встановлення з'єднання з боку STA-1 передається аналогічно окремим пакетом зі встановленим у заголовку TCP бітом ACK, або у складі наступного пакету даних. Цей пакет на часовій діаграмі не показаний.

Прикладне повідомлення передається одержувачеві фрагментами розміром не більше 1460 байтів у зв'язку з тим, що розмір пакету на IP-рівні з врахуванням TCP- і IP-заголовків не повинен перевищувати 1500 байтів. У крайньому, гіршому, випадку в процесі обміну інформацією між клієнтськими станціями безперервно передається лише один пакет даних. Вочевидь, що ефективна швидкість в такому разі буде мінімальною унаслідок наявності великої питомої частини службової інформації і тривалих обов'язкових інтервалів чекання між передачею кадрів.

У реальних умовах кількість пакетів даних, що безперервно передаються без чекання пакету підтвердження TCP:ACK залежить від ширини вікна Window, максимальний розмір якого може досягати 64 Кбайт, тобто 44-х пакетів максимальної довжини. Експериментальні дослідження показали, що кількість пакетів, що безперервно передаються у локальній безпровідній мережі коливається від 2-х до 6-ти.

Часова діаграма обміну кадрами і пакетами між клієнтськими станціями і точкою доступу по протоколу TCP у фазі передачі даних зображена на рис.2. На діаграмі показано випадок безперервної передачі двох пакетів даних. Перед початком передачі пакетів станція STA-1 повинна виграти у конкурентній боротьбі за доступ до середовища. Це відбувається лише в разі, якщо час зворотного відліку T_{BO} в даній станції буде найменше у порівнянні з аналогічним параметром в інших станціях локальної мережі, що намагаються дістати доступ до каналу, включаючи і точку доступу. На діаграмі, з метою зменшення її розмірів, інтервал T_{BO}, що витримується станціями перед відправкою пакету підтвердження TCP:ACK, не показаний.

Процес закриття з'єднання полягає в тому, що одна із станцій повинна відправити пакет зі встановленим в одиничний стан прапором FIN. Підтвердження транспортування такого пакету здійснюється окремо як на канальному, так і на транспортних рівнях.

Відправка однією із станцій пакету зі встановленим прапором RST призводить до розриву з'єднання. Часова діаграма цієї процедури аналогічна діаграмі, що зображена на рис.1.

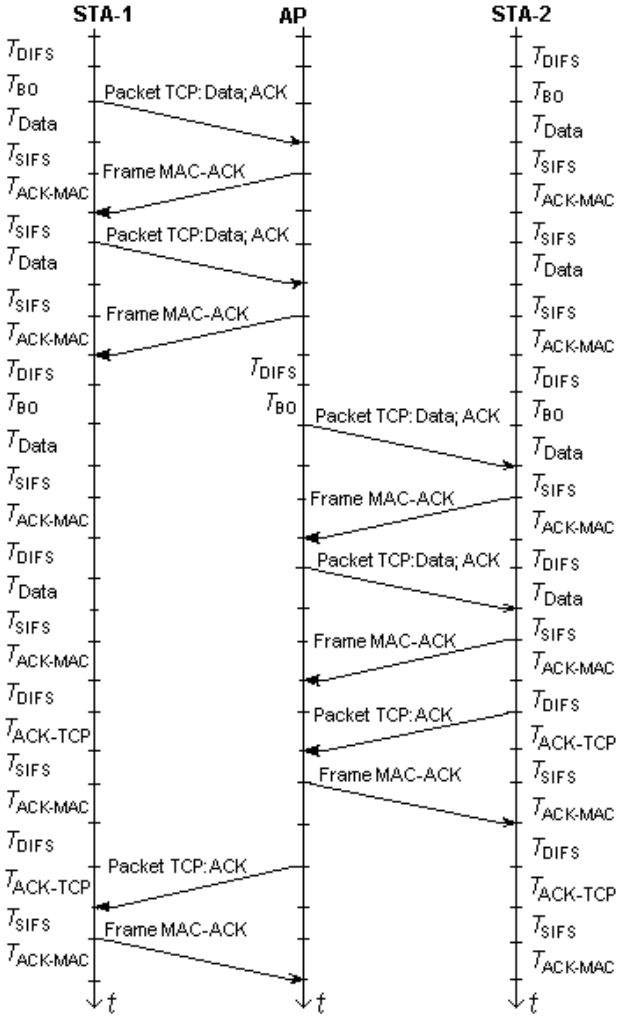


Рис. 2 − Часова діаграма передачі даних між клієнтськими станціями по TCP-протоколу

Сумарні часові витрати на встановлення з'єднання Tc, передачу даних $T_{DT\text{-}TCP}$ і завершення з'єднання T_{FIN} без врахування втрат кадрів і їх повторної передачі визначається на основі приведених часових діаграм.

$$T_C = 4(T_{DIFS}+T_{BO}+T_{ACK\text{-}MAC}) + 2(T_{SYN} + T_{ACK\text{-}TCP}), \quad (3)$$

$$T_{TD\text{-}TCP} = 4(T_{DIFS} + T_{BO} + T_{SIFS} + T_{ACK\text{-}MAC}) + 2(T_{DATA} + T_{ACK\text{-}TCP}), \quad (4)$$

де $T_{ACK\text{-}MAC}$, $T_{ACK\text{-}TCP}$ і T_{SYN} − тривалість кадрів підтвердження на MAC-рівні, TCP-рівні і пакету синхронізації відповідно; T_{DATA} − час передачі пакету даних з розміром інформаційного поля в кадрі даних N_{DATA} байтів.

У зв'язку з тим, що часова діаграма фази роз'єднання аналогічна діаграмі встановлення з'єднання, то $T_{FIN} = T_C$. Для розрахунку

інтервалів часу, потрібних для передачі службових кадрів і кадрів передачі даних, розглянемо структуру кадрів, що регламентована стандартом IEEE 802.11g [5]. До складу будь-якого кадру входять преамбула, службове поле SIGNAL і поле даних користувача (рис. 3).

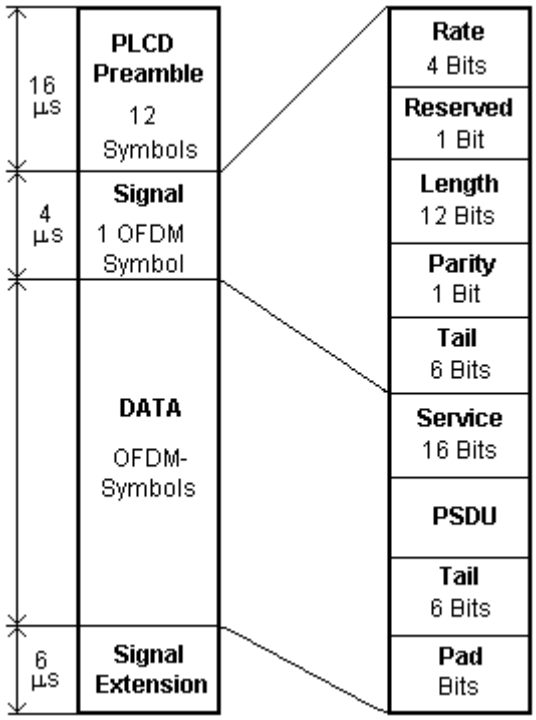


Рис. 3 − Структура кадру передачі даних по стандарту IEEE 802.11g

Преамбула складається з коротких і довгих послідовностей налаштування, загальна тривалість яких складає 16 мкс. У полі SIGNAL містяться відомості про швидкість передачі інформаційної частини кадру і довжину пакету. У нього також включений біт контролю парності та шестибітовий хвостовик (Tail), біти якого встановлені в нульове значення. Хвостовик призначений для розмежування поля SIGNAL і поля даних і служить для установлення в початковий стан згортального кодера. Поле SIGNAL кодується одним OFDM-символом, тривалість якого разом із захисним інтервалом дорівнює 4 мкс. Перед інформаційними символами поля даних розташовано 16-бітове поле SERVICE, що виконує службову функцію і яке формально відноситься до заголовку. Сім його перших бітів мають нульове значення і служать для ініціалізації псевдовипадкової послідовності скремблера. Наступні 9 бітів є резервними і також мають нульове значення.

Поле даних завершують 6 хвостових нульових бітів (Tail) і декілька додаткових бітів (Pad). Хвостовик Tail служить для ініціалізації згортального кодера, а біти Pad доповнюють останні інформаційні біти до потрібної кількості, якої бракує для кодування їх одним OFDM-символом. За преамбулою слідує PLCP-заголовок кадру, що включає поля SIGNAL і SERVICE. Після передачі OFDM-символів поля даних введено інтервал розширення тривалістю 6 мкс, протягом якого сигнали в канал зв'язку не передаються. Цей інтервал введено для завершення приймачем процедури згортального декодування.

З урахуванням того, що пакети підтвердження канального і транспортного рівнів, а також пакети даних передаються у складі кадрів фізичного рівня, то витрати часу на їх передачу обчислюються відповідно по формулах.

$$T_{SYN} = T_{PR} + T_{SIG} + T_{EX} + T_{SYM} \times \lceil (N_{SERV} + N_{Tail} + 8(H_{MAC} + H_{SNAP} + H_{IP} + H_{TCP\text{-}SYN})) / N_{DBPS} \rceil, \quad (5)$$

$$T_{ACK\text{-}MAC} = T_{PR} + T_{SIG} + T_{EX} + T_{SYM} \times \lceil (N_{SERV} + N_{Tail} + 8N_{ACK\text{-}MAC}) / N_{DBPS} \rceil, \quad (6)$$

$$T_{ACK\text{-}TP} = T_{PR} + T_{SIG} + T_{EX} + T_{SYM} \times \lceil (N_{SERV} + N_{Tail} + 8(H_{MAC} + H_{SNAP} + H_{IP} + H_{TCP\text{-}ACK})) / N_{DBPS} \rceil, \quad (7)$$

$$T_{DATA} = T_{PR} + T_{SIG} + T_{EX} + T_{SYM} \times \lceil (N_{SERV} + N_{Tail} + 8(H_{MAC} + H_{SNAP} + H_{IP} + H_{TCP} + N_{DATA})) / N_{DBPS} \rceil, \quad (8)$$

де T_{PR} і T_{SYM} − час передачі преамбули і одного символа відповідно; H_{TCP}, H_{IP}, H_{MAC} і H_{SNAP} − розмір заголовків у байтах відповідно транспортного, мережевого, канального і SNAP рівнів; $N_{ACK\text{-}MAC}$−розмір кадру підтвердження прийому на MAC-рівні у байтах; N_{DBPS} − кількість біт на один OFDM-символ; T_{EX} − інтервал розширення.

Параметри полів кадру для стандарту IEEE 802.11g [5], що використовувались у розрахункових формулах (5-10) мають наступні значення:

тривалість тайм-слоту Δt=9 мкс; міжкадрової паузи SIFS T_{SIFS}=10 мкс; міжкадрової паузи DIFS T_{DIFS}=28 мкс; преамбули T_{PR}=16 мкс; поля SYMBOL T_{SYM}=4 мкс; поля SIGNAL T_{SIG} = 4 мкс; тривалість розширення T_{EX}=6 мкс; розмір поля SERVICE N_{SERV} =16 бітів; поля Tail N_{Tail} =6 бітів; заголовку MAC H_{MAC}=34 байти; розмір кадру ACK-MAC $N_{ACK\text{-}MAC}$=14 байтів; розмір заголовку SNAP H_{SNAP}=5 байтів; заголовку IP H_{IP}=20 байтів; заголовку TCP H_{TCP}=32 байта; заголовку

UDP H_{UDP} =8 байтів; мінімальне значення лічильника зворотного відліку C_{min}=15.

Як показав аналіз мережевого потоку між клієнтськими станціями і точкою доступу, для управління потоком використовується віконний алгоритм, згідно якому передавач відправляє N_W пакетів даних, не чекаючи підтвердження від одержувача. Пакет підтвердження ACK-TCP відправляється на групу пакетів даних, що задається шириною вікна N_W. У цьому випадку формула (4) приймає наступний вигляд:

$$T_{TD\text{-}TCP} = 2\,N_W\,(T_{DIFS}+T_{BO}+T_{DATA}+T_{SIFS}+T_{ACK\text{-}MAC}) + \\ + 2(T_{DIFS}+T_{BO}+T_{SIFS}+T_{ACK\text{-}MAC}+T_{ACK\text{-}TCP}), \quad (9)$$

Обмін даними на транспортному рівні по протоколу UDP здійснюється без попереднього встановлення з'єднання і без підтвердження правильності прийому пакету. В такому разі час передачі пакету від станції відправника до станції одержувача з врахуванням ретрансляції пакету точкою доступу розраховується по наступній формулі:

$$T_{TD\text{-}UDP} = 2(T_{DIFS}+T_{BO}+T_{SIFS}+T_{DATA}+T_{ACK\text{-}MAC}), \quad (10)$$

У безпровідних мережах з функцією розподіленої координації доступу час зворотного відліку T_{BO} визначається як добуток довжини тайм-слоту на випадкове число *Ci*, що генерується генератором випадкових чисел з рівномірним законом розподілу [4]. У даній роботі при розрахунку T_{BO} в якості *Ci* приймалося середнє значення, яке вибране з випадкового інтервалу, що формується генератором випадкових чисел у діапазоні від 0 до C_{min}, тобто $T_{BO}=(\Delta t\times C_{min})/2$. Цей час, виходячи із стандартних значень Δt і C_{min}, дорівнює 67,5 мкс.

При розрахунку потенційної пропускної спроможності передбачалося, що за час сеансу зв'язку передається дуже велика кількість пакетів даних (>1000). В такому разі часом, що витрачається на установку з'єднання і на його завершення, можна нехтувати зважаючи на його малу величину. Розмір вікна N_W було узято рівним 7, який прийнято за умовчанням у протоколі TCP. Змінюючи кількість байтів у блоці від 256 до 1460, та підставляючи його у формули (9), (4-8) і (1) отримуємо графік залежності теоретичної максимальної ефективної швидкості передачі даних з віконним алгоритмом $V_{TCP\text{-}W3}$ з передачею одного підтвердження на три передані пакети і з підтвердженням кожного пакету $V_{TCP\text{-}W1}$ (рис. 4). На цьому ж графіку показана залежність ефективної швидкості передачі даних V_{UDP} по протоколу UDP. Значками × на малюнку позначена середня швидкість передачі даних по протоколу TCP, яка визначена експериментально з використанням пакету IxChariot.

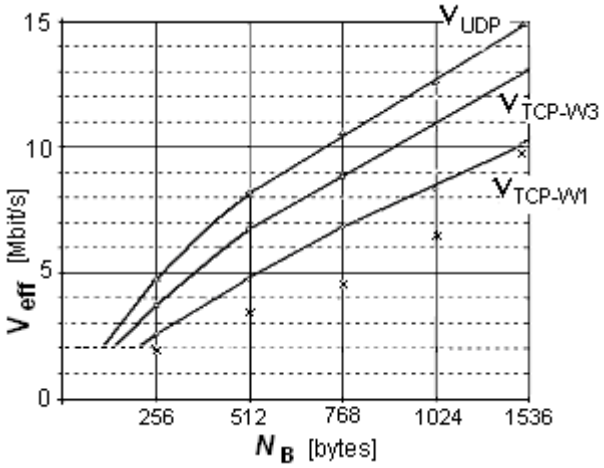


Рис. 4 − Залежності теоретичної і реальної ефективної швидкості передачі даних на транспортному рівні від розміру поля даних

Вимір ефективної швидкості проводився при технічній швидкості передачі 54 Мбіт/с. За кожен сеанс зв'язку передавалося 10 Мбайт текстових повідомлень. В процесі вимірів змінювалася довжина інформаційної частини пакету даних. Передача здійснювалася між двома стаціонарними комп'ютерами, розташованими на відстані 3 м один від одного, забезпеченими радіоадаптерами типа D-Link, що працюють по алгоритму, визначуваному стандартом IEEE 802.11g у режимі "only g". В якості базової станції використовувалася точка доступу типу LinkSys WRT-54GL.

Як видно з наведених графіків, розрахована по отриманих формулах ефективна швидкість передачі даних добре збігається з даними, що отримані при експериментальних вимірах пропускної спроможності на реальних мережах. Відхилення реальної пропускної спроможності від теоретичної пов'язане з втратою пакетів за рахунок перешкод і переповнювання буфера, а також передачею дублюючих пакетів. У процесі експерименту спостерігалися випадки, коли максимальна експериментальна швидкість перевищувала теоретичну для випадку передачі одного кадру підтвердження на кожен переданий джерелом пакет ($V_{TCP\text{-}W1}$), що імовірно пов'язане з передачею кадру підтвердження на декілька пакетів даних. Проте випадків перевищення максимальної швидкості величини $V_{TCP\text{-}W3}$ не спостерігалося.

СПИСОК ЛІТЕРАТУРИ

[1] A.V. Barbosa, M.F. Caetano, J.I. Bordim, The theoretical maximum throughput calculation for the IEEE 802.11g standard, International *Journal of Computer Science and Network Security*, (11) 4 (2011), pp.136-142.

[2] V. Chernega, E. Glatz, S. Vinichenco, Estimation of effective data rate at transport level in networks of 802.11b, *SevNTU Journal, Informatics, Electronics, Communications*, Sevastopol, (131) (2012), pp.32-36 (in Russian).

[3] P. Roshan, J. Leary, 802.11 *Wireless LAN, Fundamentals*, Moscow, 2004, p. 304 (in Russian).

[4] V. Chernega, B. Plattner, *Wireless Local Area Computer Networks*, Kyiv, 2013, p.238 (in Ukrainian).

[5] IEEE Std 802.11g – 2003. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band.*

***Чернега Віктор Степанович**, 66 років, к.т.н., доцент кафедри інформаційних систем Севастопольського національного технічного університету. З 1964 по 1969 р. навчався у Севастопольському приприладобудівному інституті за фахом "Автоматика і телемеханіка". У 1975 році закінчив аспірантуру при кафедрі автоматики і телемеханіки КПІ. Декілька років стажувався і працював у Федеральній вищій технічній школі м. Цюріха (Швейцарія).*

Автор 25 авторських свідоцтв на винаходи СРСР і патентів України, 5-ти учбових посібників, рекомендованих МОН України для студентів ВНЗ, та біля 60 науково-технічних публікацій. Наукові інтереси: проектування і дослідження комп'ютерних мереж, систем передачі даних, систем автоматизованого контролю.

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

PERFORMANCE OF A TRANSPORT LEVEL OF WLANS IEEE 802.11g FUNCTIONING IN INFRASTRUCTURAL MODE

Victor Chernega

Sevastopol National Technical University
University Str. 33, Sevastopol-99053, Ukraine
v_chernega@rambler.ru

Abstract: *The procedure of frames exchange between client computers and a wireless access point based on 802.11g standard is analyzed in details. We have obtained the expressions which allow calculate the potential bandwidth of such wireless network.*

Keywords: *WLANS of 802.11g, TCP, UDP, infrastructural mode, carrying capacity.*

INTRODUCTION

The purpose of work is the detailed revealing of the procedure of frame exchange between client computers and base station in WiFi WLANS during the data transfer over protocols of a transport level of TCP and UDP for receipt of expressions, allowing to estimate maximally possible speed of payload transfer between the users of the network of IEEE 802.11g in the base mode in default of collisions and hindrances in a communication channel. It is thus assumed that the network works in the infrastructural mode while the frame exchange in the network is carried out over the access point (AP).

The throughput of virtual channel between two client stations of the network, created during the connection establishment on a transport level, is determined by payload content in the bits of N_B, given to the recipient during the session of connection in seconds T_S, i.e. is a transfer of effective data rate $V_{eff} = N_B / T_S$.

Procedure of connection session on protocol of TCP between two client stations of STA-1 and STA-2 through the access point AP includes the phase of establishment of connection between a source and recipient, phase of data transfer, phase of closing and disconnection. For determination of duration of connection phases, data transfer and disconnection by the author temporal diagrams of connection sessions between two client computers were built and the detailed analysis of exchange process of frames and packages in WLAN, working in infrastructural mode with the help of program of monitoring and analysis of network packages of CommView for WiFi was conducted.

On receipt of calculation formulas it was taken into account, that the client station of STA-1 during the transmission of TCP-packages before consuming the channel analyses its state. On discovering that the communication channel is free, the station maintains the obligatory pause T_{DIFS}, after expiration of which the random numbers generator starts and time of the reverse counting (*Backoff*) T_{BO} is calculated. If a communication channel remains free upon termination of interval of T_{BO}, the station of STA-1 forms a TCP-segment with the set flag of synchronization of SYN1:1 and zero data field. This segment on network-level is supplemented by an IP-header, in which the IP-address of the destination station of STA-2 and sender stations STA-1 is specified. The formed package is encapsulated in the frame of data link layer, in which MAC-address of access point (BSSID), recipient STA-2 and source of STA-1 are contained. The point of access accepts a TCP-packet with the bit of synchronization and, in default of errors in it, after self-control of obligatory short interframed pause of SIFS, the frame of confirmation sends the stations of STA-1 on the channel level (a frame MAC-ACK).

On the next stage the access point, along with the client stations of network, joins competition for the receipt of access to the environment. For this purpose it counts off the obligatory intervals of T_{DIFS} and T_{BO} and when there is no supportive in a channel it sends the package of request for connection establishment (package of TCP: SYN1:1) to the destination stations. STA-2 confirms

the correctness of frame reception on channel level and after competition for a channel a package with flags of confirmation of connection establishment set in the header of TCP (flag ACK is set in 1) and request for connection establishment from the station 2 (flag of SYN2:1) are sent. Then in a similar way a package of TCP:SYN2:1; ACK:1 is delivered to the station of STA-1. Confirmation of connection establishment from STA-2 is sent similarly in a separate package with set in the header of TCP by a bit ACK or in composition of the next package of information.

In the article the temporal diagram of exchange of frames and packages between the client stations and access point on protocol of TCP in the phase of data transfer is presented. The case of continuous transmission of two data packages is shown in the diagram. During the construction of the diagram it was taken into account that before the beginning of packages transmission the station of STA-1 must win the competition for access to the environment. It happens only in case when time of the reverse counting T_{BO} in this station is the least compared with an analogical parameter in other stations of local network, trying to get access to the channel, including the access point. On a diagram, in order to diminish of its sizes, the interval T_{BO}, maintained by the stations before the sending of package of confirmation of TCP:ACK, is not shown.

The process of closing of TCP-connection is that one of the stations must send a package with the flag FIN set in the single state. Confirmation of transfer of such package is carried out separately both on a channel and on transport level. The sending of package by one of the stations with the set flag RST results in the break of connection. The temporal diagram of this procedure is analogical to the diagram of connection establishment.

On the basis of the temporal diagrams built by an author the formulas are received, allowing counting temporal expenses on connection establishment T_C, data transfer of $T_{DT\text{-}TCP}$ and completion of connection T_{FIN} without considering the losses of frames and their repeated transmission (5-9). The expenses of time on the transmission of packages on data link and transport levels were counted considering that confirmation packages, and also data packages are transferred in composition of of theoretically most effective data rate $V_{TCP\text{-}W3}$c by a window algorithm with the transmission of one confirmation on three sent packages and with confirmation of every package $V_{TCP\text{-}W1}$are shown. On the same picture the graph of effective data rate V_{UDP} on protocol UDP is represented. The average data rate on protocol TCP, which is determined experimentally with the use of package of Chariot, is shown with the symbols ×. the frames of physical level. In the article the formula (10) of time expense for data transfer is shown also on transport level on protocol of UDP $T_{TD\text{-}UDP}$ considering retransmitting of package by the access point. In the article the graphs of dependence

Measuring of effective rate was made at technical transmission speed of 54 Mbit/s. For every connection session 10 MByte of text messages were transmitted. During measuring the length of informative part of package of information was changed. The transmission was carried out between two stationary computers, located in the distance of 3 meter from each other, supplied with radioadapters of D-Link type, working on an algorithm, determined by the standard of IEEE 802.11g in the mode of "only g". As the base station the access point of LinkSys WRT-54GL was used.

Researches showed that the effective data rate calculated by the obtained formulas coincides well enough with data, received during the experimental measuring of throughput on the real networks. Deviation of the real throughput from the theoretical is connected with the loss of packages due to spikes and buffer overflow, and also with the transmission of doubled packages. In the process of experiment there were cases, when high experimental rate exceeded theoretical for the case of transmission of confirmation frame on every package ($V_{TCP\text{-}W1}$) sent by a source, that is obviously connected with the transmission of one confirmation frame for a few data packages. However there were no cases of exceeding of maximum rate of the value $V_{TCP\text{-}W3}$.

REFERENCES

[1] A.V. Barbosa, M.F. Caetano, J.I. Bordim, The Theoretical Maximum Throughput Calculation for the IEEE 802.11g Standard, *International Journal of Computer Science and Network Security*, (11) 4 (2011), pp.136-142.

[2] V. Chernega, E. Glatz, S. Vinichenco, Estimation of effective data rate at transport level in networks of 802.11b, *SevNTU Journal, Informatics, Electronics, Communications*, Sevastopol, (131) (2012), pp.32-36 (in Russian).

[3] P. Roshan, J. Leary, *802.11 Wireless LAN*, Fundamentals, Moscow, 2004, p. 304 (in Russian).

[4] V. Chernega, B. Plattner, *Wireless Local Area Computer Networks*, Kyiv, 2013, p.238 (in Ukrainian).

[5] IEEE Std 802.11g – 2003, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*.

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

POST-CORRECTION OF ADC NON-LINEARITY USING INTEGRAL NON-LINEATITY CURVE

Vladimir Haasz [1)], David Slepicka [1)], Petr Suchanek [2)]

[1)] Czech Technical University in Prague, Faculty of electrical Engineering, Technicka 2, 16627 Praha 6, Czech Republic, haasz@fel.cvut.cz, slepicd@fel.cvut.cz, http://measure.feld.cvut.cz
[2)] Evolving systems consulting s.r.o., Čs. armády 14, 160 00 Praha 6, Czech Republic, petr.suchanek@evolvsys.cz

Abstract: *The accuracy of AD conversion can be improved using the post-correction of digitizer non-linearity. In principle two methods could be applied – look-up table or an analytical inverse function of integral non-linearity curve (INL(n)). Look-up table can be easily implemented but it demands huge memory space particularly for high resolution ADCs. Inverse function offers flexible solution for parameterization (e.g. frequency dependence) but it also requires fast DSP for real-time correction. The data or coefficients for both methods are frequently determined from a histogram of acquired pure sinusoidal signal. Non-linearity curve can also be gained by another procedure demanding significantly less samples – approximation from a frequency spectrum. The correction of ADC nonlinearity by means of inverse function of INL(n) curve is analyzed in this paper and the results are presented.*

Keywords: *analog-to-digital converter, ADC non-linearity, INL, transfer function, approximation, non-linearity correction, simulations, experimental verification.*

1. INTRODUCTION

The ADC non-linearity is inherently described by the Integral Non-linearity curve $INL(n)$ which is defined as the difference of ADC output and input as the function of the input level. $INL(n)$ can be directly determined using histogram method [1], but this method demands a huge number of samples in a record, thus it is time consuming. However, non-linearity causes also a distortion in the digitized signal and the frequency spectrum can provide similar information as the $INL(n)$ in the code domain.

The $INL(n)$ curve can be split into its low code frequency component (LCF) and the high code frequency component (HCF). The LCF (the rough curve of the $INL(n)$) – see Fig. 1, dotted curve) is responsible for harmonic distortion at lower harmonic components [2, 3, 4], usually the strongest are 2nd and the 3rd ones. If an approximation of the $INL(n)$ curve using polynomials is applied, the third order polynomial is mostly sufficient for the following integral non-linearity correction.

2. APPROXIMATION OF *INL(N)* CURVE

Using polynomials the $INL(n)$ is approximated by

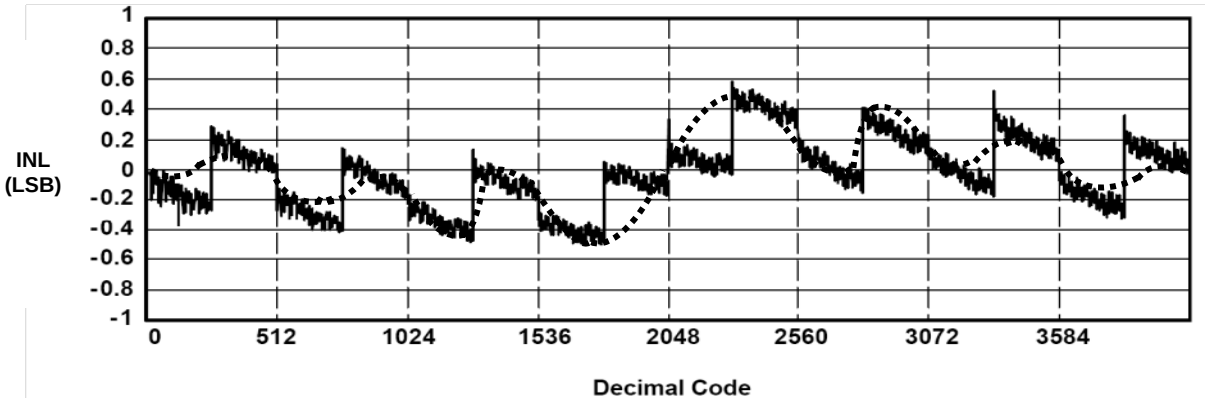


Fig. 1 – An example of *INL(n)* curve and its low code frequency component (dotted curve)

$$INL(n) = \sum_{h=1}^{H_{\max}} a_h x^h(n) \tag{1}$$

where a_h are the nonlinearity coefficients up to the maximum order $H_{\max}$, which is the highest harmonic component considered, n is the normalized ADC code with a bipolar range, and x the ADC input. Having the coefficients a_h and consequently the approximation of $INL(n)$ curve, the non-linearity of digitizer can be corrected. The approximated transfer function TF has to be calculated by adding a straight line to the $INL(n)$, such that

$$TF(n) = n + INL(n) \tag{2}$$

where n is the ADC code and after the substitution it can be expressed as

$$TF(n) = n + \sum_{h=1}^{H_{\max}} a_h x^h(n) \tag{3}$$

If the transfer function TF is monotonical, its inverse exists. For this case, let's propose that the approximation of the inverted transfer function TF^{-1} will also be a polynomial of the same order ($K_{max} = H_{max}$) defined as

$$TF^{-1}(y) = \sum_{k=1}^{K_{\max}} b_k y^k \tag{4}$$

where y is the ADC output. Substituting $y = TF(n)$ from (3) into (4)

$$\begin{aligned} TF^{-1}(TF(n)) &= \sum_{k=1}^{K_{\max}} b_k [TF(n)]^k = \\ &= \sum_{k=1}^{K_{\max}} b_k \left[\sum_{h=1}^{H_{\max}} a_h n^h \right]^k \end{aligned} \tag{5}$$

The distortion of the 2nd and the 3rd harmonic component is usually the most important for majority of digitizers and the higher components are usually negligible. Therefore the $K_{max} = H_{max} = 3$ will be taken into account for the following solution. In this case the general expression (5) changes to

$$\begin{aligned} TF^{-1}(TF(n)) &= \sum_{k=1}^{3} b_k \left[\sum_{h=1}^{3} a_h n^h \right]^k = \\ &= b_1(a_1 n + a_2 n^2 + a_3 n^3) + \\ &+ b_2(a_1 n + a_2 n^2 + a_3 n^3)^2 + \\ &+ b_3(a_1 n + a_2 n^2 + a_3 n^3)^3 \end{aligned} \tag{6}$$

Considering

$$TF^{-1}(TF(n)) = n \tag{7}$$

and comparing the coefficients of the same powers of n an over determined equation system arises (3 unknowns variables b_1, b_2, b_3, 9 equations).

Two methods for the determination of coefficient b_k are presented in this paper. In the first method the coefficients b_1, b_2, b_3 are determined from 3 low-order equations, the equations with polynomials n^l, $l > 3$ are neglected. The coefficients are given by

$$b_1 = \frac{1}{a_1}, \quad b_2 = \frac{a_2}{a_1^3}, \quad b_3 = \frac{2a_2^2}{a_1^5} - \frac{a_3}{a_1^4} \tag{8}$$

The second method is more sophisticated. Since equation (7) can hardly be fulfilled completely the error function

$$e = \left(TF^{-1}\left(TF(n)\right) - n \right)^2 \tag{9}$$

is minimized (least square).

Let's integrate the error function over the full-scale of the ADC to obtain the area below the error function. The full-scale range of the ADC spans over $\langle -1; +1 \rangle$ interval because of normalization.

$$I(b_1, b_2, b_3) = \int_{-1}^{+1} \left(TF^{-1}\left(TF(n)\right) - n \right)^2 dn \tag{10}$$

and after the substitution from (6)

$$\begin{aligned} I(b_1, b_2, b_3) = \int_{-1}^{+1} & (b_1(a_1 n + a_2 n^2 + a_3 n^3)^3 + \\ & + b_2(a_1 n + a_2 n^2 + a_3 n^3)^2 + \\ & + b_3(a_1 n + a_2 n^2 + a_3 n^3) - n)^2 \, dn \end{aligned} \tag{11}$$

Integration (11) eliminates the variable n and only variables b_k remain. Let's take the partial derivatives of the $I(b_1,b_2,b_3)$ function with respect to three variables and equal them to zero.

$$\frac{\delta I}{\delta b_1} = 0, \quad \frac{\delta I}{\delta b_2} = 0, \quad \frac{\delta I}{\delta b_3} = 0 \tag{12}$$

The system with three equations of three variables is obtained

$$\begin{aligned} c_1 b_1 + c_2 b_2 + c_3 b_3 &= d_1 \\ c_4 b_1 + c_5 b_2 + c_6 b_3 &= d_2 \\ c_7 b_1 + c_8 b_2 + c_9 b_3 &= d_3 \end{aligned} \tag{13}$$

where c_i and d_j are collected terms resulting from the partial derivations (12), e.g.

$$c_1 = \frac{4}{3}a_1^2 + \frac{4}{5}a_2^2 + \frac{8}{5}a_1 a_3 + \frac{4}{7}a_3^2 \tag{14}$$

and

$$d_1 = \frac{4}{3}a_1 + \frac{4}{5}a_3 \tag{15}$$

The equations (13) can be rewritten into a matrix form

$$\mathbf{C\,b} = \mathbf{d} \tag{16}$$

and the coefficients b_1, b_2, b_3 can be calculated by applying the Cramer's rule based on determinants

$$b_1 = \frac{\det(C_1)}{\det(C)}, \quad b_2 = \frac{\det(C_2)}{\det(C)}, \quad b_3 = \frac{\det(C_3)}{\det(C)} \tag{17}$$

The solution is

$$\begin{aligned} b_1 &= \frac{d_1c_5c_9 + d_2c_8c_3 + c_2c_6d_3 - c_3c_5d_3 - c_2d_2c_9 - d_1c_6c_8}{c_1c_5c_9 + c_4c_8c_3 + c_2c_6c_7 - c_3c_5c_7 - c_2c_4c_9 - c_1c_6c_8} \\ b_2 &= \frac{c_1d_2c_9 + c_4d_3c_3 + d_1c_6c_7 - c_3d_2c_7 - d_1c_4c_9 - c_1c_6d_3}{c_1c_5c_9 + c_4c_8c_3 + c_2c_6c_7 - c_3c_5c_7 - c_2c_4c_9 - c_1c_6c_8} \\ b_3 &= \frac{c_1c_5d_3 + c_4c_8d_1 + c_2d_2c_7 - d_1c_5c_7 - c_2c_4d_3 - c_1d_2c_8}{c_1c_5c_9 + c_4c_8c_3 + c_2c_6c_7 - c_3c_5c_7 - c_2c_4c_9 - c_1c_6c_8} \end{aligned} \tag{18}$$

3. SIMULATION OF NON-LINEARITY CORRECTION

The correction of the simulated ADC nonlinearity was performed in the second step. The coefficients of the polynomials of the approximated non-linearity *INL*(*n*) curve (1) were computed from the histogram method [1] measured by the digitizer NI PXI 5122. Spectrally pure (filtered) testing signal (THD < -130 dB) was used for this purpose [5]. The approximation of the inverted transfer function was found applying the polynomials. Only the most dominant coefficients a_i (the 2nd and the 3rd order) were considered.

The levels of harmonic components of the simulated output signal and of the same signal after the correction are presented in Table 1. The performance of both methods mentioned above is shown.

Table 1. Results of Correction

Harmonic component	Digital output before correction	Digital output after correction	
		Direct inversion	LSE minimization
2	–77dB	–131dB	–142dB
3	–80dB	–136dB	–155dB

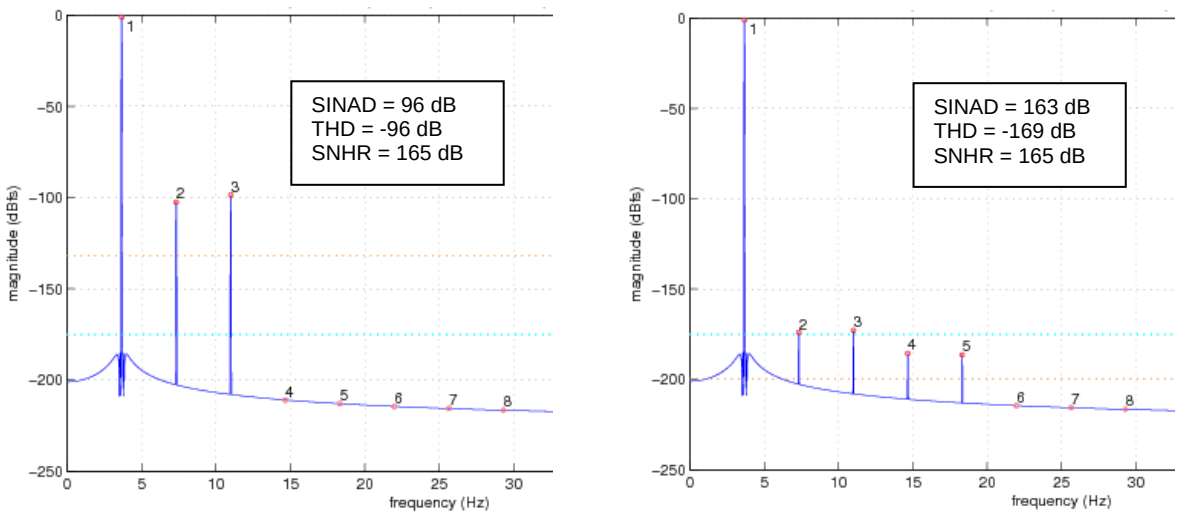
The modeled input and corresponding output signal were in a very good agreement with the real signals. The correction applied on this signal showed to be very effective. However, the correction on real output data did not improve the signal as expected. The reason seemed to be in other ADC imperfections (additive noise, jitter in sampling, non-zero sampled signal phase and hysteretic behavior) which were not taken into account in the simulation. To find the source of the worse correction results in the case of real data further simulations were executed. The influence of the incoherently sampled signal (with non-zero α) was suppressed by applying the Blackman-Harris window of the 7th order to the recorded data. The simulated distorted ADC output was generated as

$$\mathrm{ADC}_{\mathrm{output}} = y = \sum_{h=1}^{3} \mathrm{sign}\left(\mathbf{a}(h)\right) \left[\mathrm{abs}\left(\frac{\mathbf{a}(h)}{\mathbf{a}(1)} \right) \mathrm{ADC}_{\mathrm{input}} \right]^h \tag{19}$$

where $\mathbf{a} = [adc\_full\_scale, -18, -13]$ ($adc\_full\_scale = 2^{23}$ for the simulated 23-bit ADC and the numbers –18 and –13 are the coefficients of the 2nd and the 3rd non-linearity order). No rounding (quantization in amplitude) of the ADC output was used in order to better observe the performance of the correction. The following ADC imperfections were added to the modeled output signal:

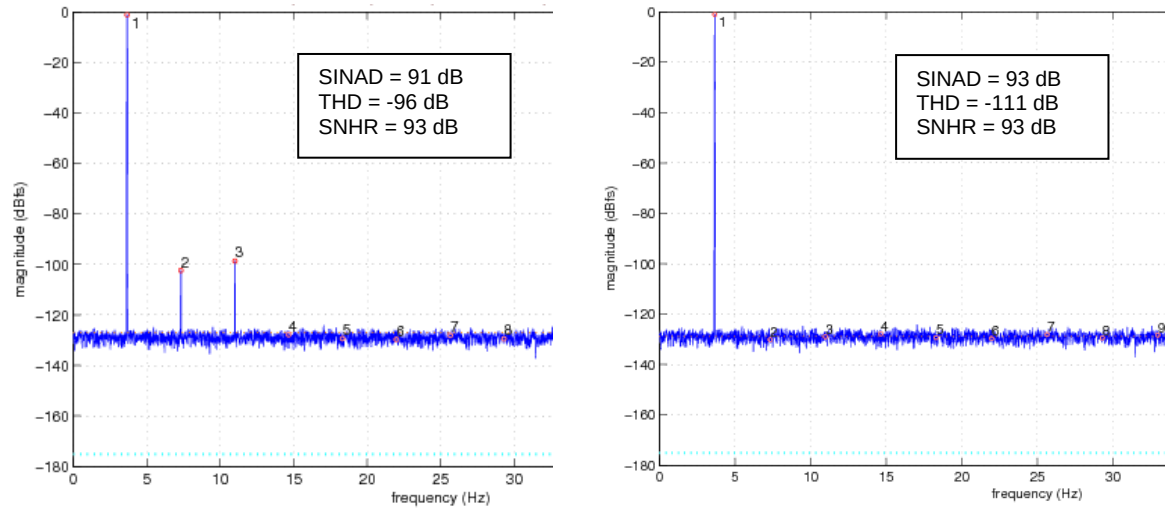
a) additive white noise,
b) sampling jitter,
c) influence of non-zero sampled signal phase,
d) hysteresis.

The result corresponding to harmonic distortion only was used as the reference one (Fig. 2). The influence of the additive noise is presented in Fig. 3.



a) Frequency spectrum before correction **b) Frequency spectrum after correction**

Fig. 2 – Non-linearity correction of simulated signal – sine-wave signal with harmonic distortion



a) Frequency spectrum before correction **b) Frequency spectrum after correction**

Fig. 3 – Non-linearity correction of simulated signal – sine-wave signal with white noise (σ^2 = 60 LSB) and harmonic distortion

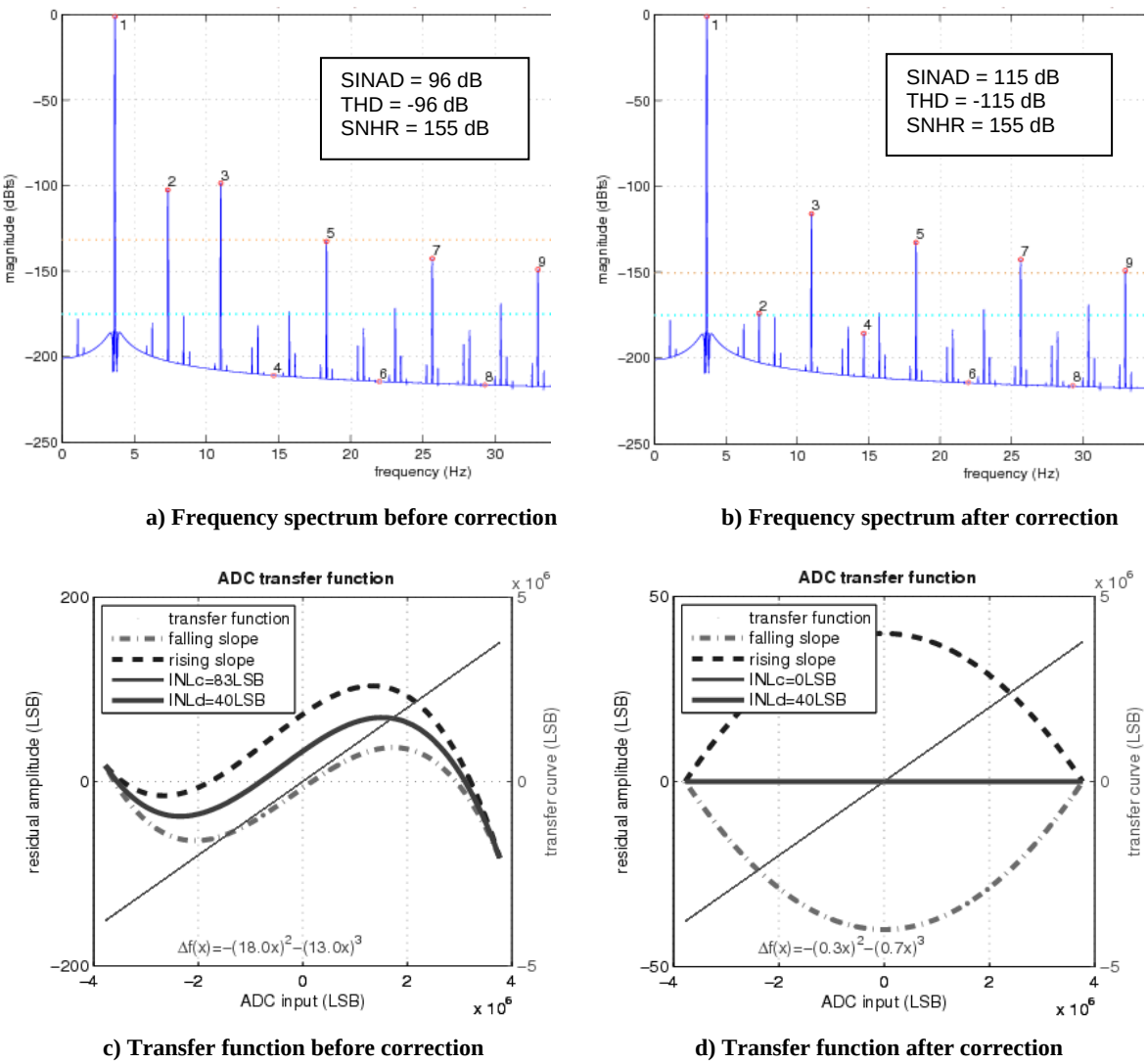
This simulation proved that the presence of an additive white noise does not cause noticeably influence the results of correction – higher harmonic components were suppressed to negligible level below noise. The same result was found for sampling jitter. The simulation with variable non-zero sampled signal phase showed also no influence.

The last imperfection of a real ADC, which was investigated, was the hysteresis. For better observation the resulting frequency spectrum as well as the integral non-linearity curve (the deviation from the ideal transfer function – residual amplitude) were calculated (see Fig. 4).

The residual amplitude for falling and rising slopes was plotted individually. The dashed and dot-dash lines were reconstructed from the falling and rising slopes of the signal. *INLd* is so called "differential-mode" component which corresponds to ADC hysteresis behavior. The full line represents the "common-mode" non-linearity component *INLc* [6]. The hysteresis was modeled using equation

$$y^{\mathrm{hyst}}(x) = \beta\left[\left(\frac{x}{X_1}\right)^2 - 1\right]\mathrm{sign}(x'), \qquad (20)$$

where $y^{\mathrm{hyst}}(x)$ is the additive contribution of the ADC hysteresis to its output, β is a the proportion factor, X_1 is the amplitude of the input signal, and the sign(x') is a binary function with +1 and –1 output values depending on the slope of the input signal x. Fig. 4c shows "typical" curves of residual amplitude, which is significantly different for failing and rising slopes.



a) Frequency spectrum before correction **b) Frequency spectrum after correction**

c) Transfer function before correction **d) Transfer function after correction**

Fig. 4 – Non-linearity correction of simulated signal – sine-wave signal with hysteresis α = 40 LSB and harmonic distortion

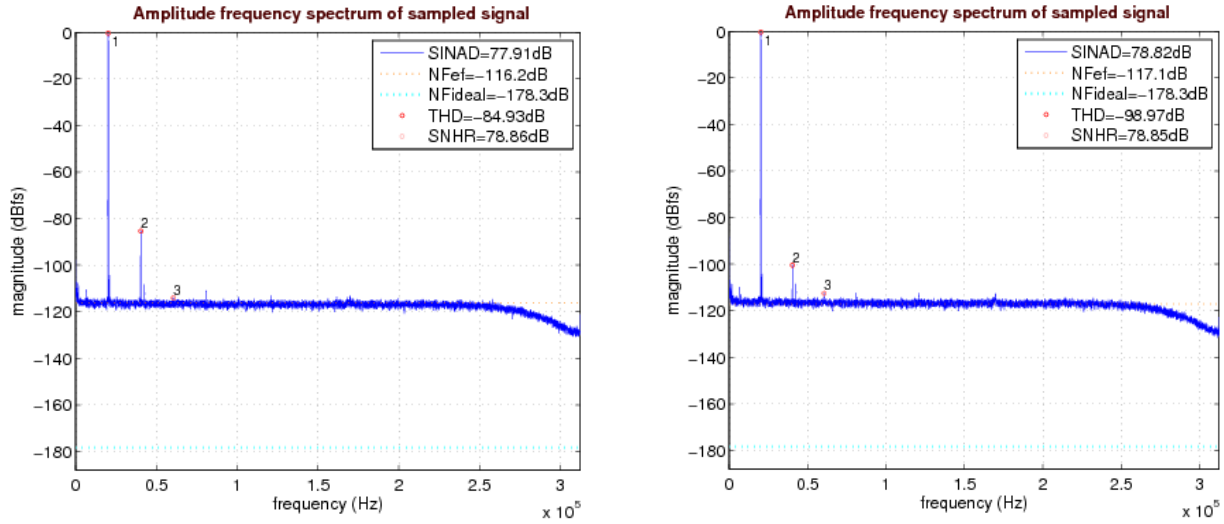
The result of the correction is presented in Fig. 4d. The "common mode" non-linearity was correctly estimated as $\Delta f(x) = -(18.0x)^2 - (13x)^3$ LSB and removed. Only the hysteresis remained after the correction. It corresponds to the premise that the proposed method can correct only "pure" non-linearity, but not the hysteresis.

4. EXPERIMENTAL VERIFICATION

To verify the simulation results experimental measurements using two high quality digitizers (23-bit Digitizer VXI HP E1430A and 24-bit Digitizer NI PXI-5922) were performed. High-quality ADC testing system at the CTU in Prague [5] was applied for this purpose. The input signal was generated by ultra-low distortion Stanford Research DS360 generator and it was subsequently filtered by band-pass filter to achieve high spectral purity of the signal. The record of 2 MSa was divided into four segments. One of them was selected as the reference for calculating coefficients of the inverted polynomial. The other three data segments were then corrected. In the case of VXI HP E1430A digitizer input signal frequency of 20.19 kHz was used. An example of the frequency spectra of the output signal before and after the correction is shown in Fig. 5. The *THD* was improved by about 15 dB by means of the correction.

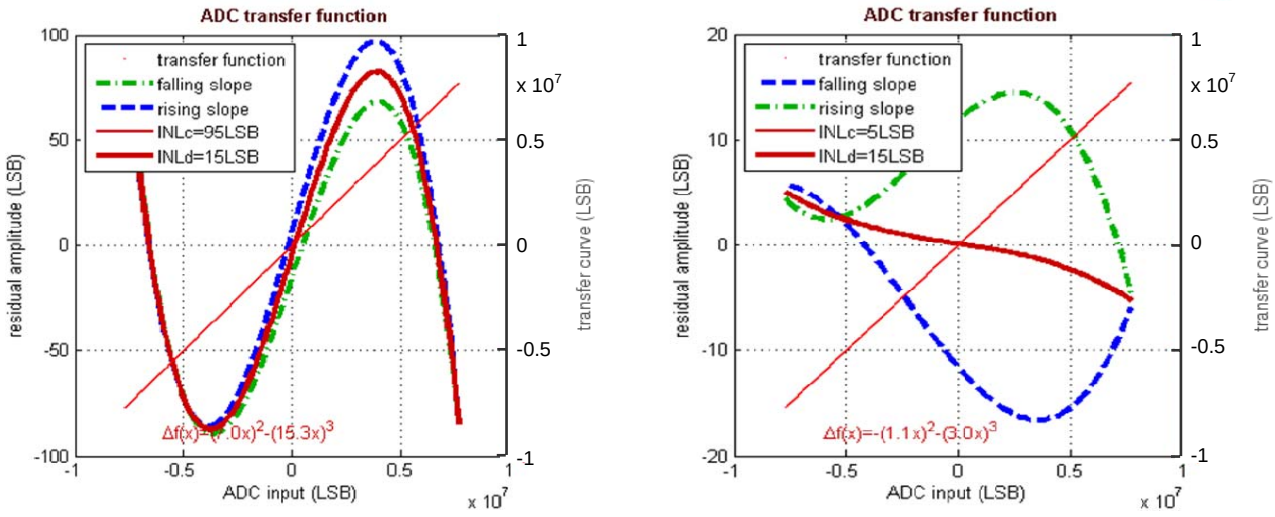
Secondly, the 24-bit Digitizer NI PXI-5922 was tested. In this case two frequencies of input signal were used: 20.19 kHz and 1.053 MHz. The other conditions remained the same. The residuals before and after the correction show more details than the frequency spectra in this case (see Fig. 6 and 7).

In case of 20 kHz input signal the hysteresis slightly influences the result. The "common mode" non-linearity is well suppressed by the correction but the residual non-linearity of about 15 LSB caused by hysteresis (different for falling and rising slopes) remains. The resulting "common-mode" non-linearity decreased about 20 times.



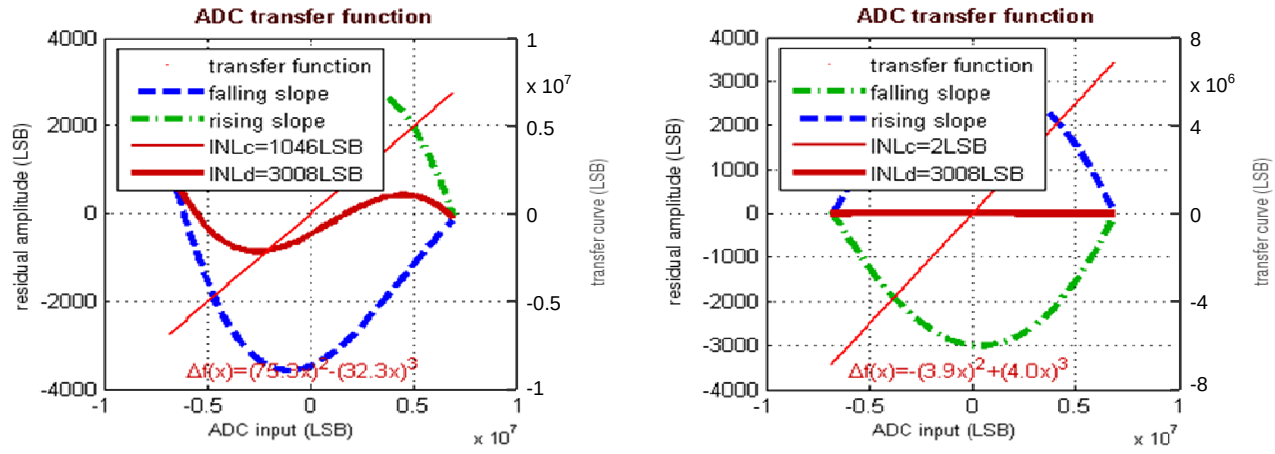
a) Before correction **b) After correction**

Fig. 5 – Frequency spectra of digitized output signal (VXI HP E1430A)



a) Before correction **b) After correction**

Fig. 6 – Integral non-linearity (NI PXI-5922, f_{inp} = 20.19 kHz)



a) Before correction **b) After correction**

Fig. 7 – Integral non-linearity (NI PXI-5922, f_{inp} = 1.053 MHz)

However, for 1 MHz input signal the hysteresis is about two hundreds times higher then for 20 kHz and the "differential-mode" non-linearity caused by the hysteresis is dominant. Common-mode integral non-linearity is suppressed well by the correction indeed but the residual "differential-mode" non-linearity remains and the remaining non-linearity after the correction is practically the same as before.

5. CONCLUSION

Two methods for calculation of coefficients of the inverted polynomial used for ADC non-linearity correction were introduced, derived and compared. The first one uses more straightforward derivation of the coefficients; the second one minimizes the least square error.

The first simulation did not take ADC imperfections (additive noise, jitter in sampling, non-zero sampled signal phase and hysteretic behavior) into account. They showed that the both methods used for calculation coefficients are applicable. Simulations verified that the correction is useable also in the cases when other ADC imperfections are not negligible. This statement was also confirmed experimentally.

The proposed method of ADC non-linearity correction using polynomial approximation of the integral non-linearity $INL(n)$ and its inverse function gives mostly good results but not always. It concerns e.g. digitizers with noticeable hysteresis which is particularly common for signals with frequency near the maximum input frequency of digitizers. Generally, inverse function used for post-correction of INL is frequency dependent. For this reason it cannot be directly used for wide-band signals. However, this issue also concerns other post-correction methods, e.g. look-up table.

6. REFERENCES

[1] *IEEE 1241-2000 Standard for Analog to Digital Converters,* The Institute of Electrical and Electronics Engineers, 2001.

[2] L. Michaeli, P. Michalko, J. Saliga, Fast Gesting of ADC Using Unified Error Model, *The 17th IMEKO world congress*, pp. 534–537, Dubrovnik, Croatia, 2003.

[3] A.C. Serra, M. F. da Silva, P. Ramos, R. C. Martins, L. Michaeli, J. Saliga, Combined Spectral and Histogram Analysis for Fast ADC Testing, *IEEE Transactions on Instrumentation and Measurement*, (54) 4 (2005).

[4] N. Björsell, P. Händel, Achievable ADC Performance by Postcorrection Utilizing Dynamic Modeling of the Integral Nonlinearity", *EURASIP Journal on Advances in Signal Processing*, vol. 2008.

[5] V. Haasz, M. Komarek, J. Roztocil, D. Slepicka, P. Suchanek, System for Testing Middle-Resolution Digitizers Using Test Signal up to 20 MHz, *IMTC/06 Proceedings of the 23th IEEE Instrumentation and Measurement Technology Conference*.

[6] C.L. Monteiro, P. Arpaia, A.C. Serra, A comprehensive phase-spectrum approach to metrological characterization of hysteretic ADCs, *IEEE Transactions on Instrumentation and Measurement*, (51) 4 (2002) pp. 756-763.

[7] P. Suchanek, V. Haasz, D. Slepicka, ADC Nonlinearity Correction Based on INL(n) Approximations. *IEEE International Workshop IDAACS 2009*. Rende, Italy 2009, pp. 137–140.

[8] P. Suchanek, V. Haasz, D. Slepicka, Evaluation of ADC Non-linearity Correction Based on *INL*(*n*) Approximation, *BEC 2012 – 13th Biennial Baltic Electronics Conference* [CD-ROM]. Tallinn, Estonia 2012, pp. 101-104.

***Vladimír Haasz**, Head of the Dept. of Measurement of the Czech Technical University in Prague, member of the IMEKO TC-4 – Measurement of Electrical Quantities, member of the IEEE Instrumentation & Measurement Society. He is interested in measurement of dynamic parameters of AD modules including EMC.*

***David Slepička**, received the PhD degree in Electrical Engineering in 2005 from the Czech Technical University in Prague. He is currently doing a research on high-resolution ADC testing. The research involves the development of methodologies for analog and digital signal processing and the design of DAQ systems.*

***Petr Suchánek** received the PhD degree in Electrical Engineering in 2012 from the Czech Technical University in Prague. The main area of his research concerns the AD converter modelling, results presented here followes from his PhD thesis. Currently, he works for a private company in Czech Aerospace industry.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ ДЛЯ ВЫЯВЛЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

Дмитрий Комашинский, Игорь Котенко

Лаборатория проблем компьютерной безопасности СПИИРАН
14 линия, 39, Санкт-Петербург, 199178, Россия
komashinskiy@comsec.spb.ru, ivkote@comsec.spb.ru,
http://comsec.spb.ru

Резюме: *в статье проводится обзор наиболее значимых работ в области создания систем обнаружения и идентификации вредоносных программ на основе методов интеллектуального анализа данных. Для формализации этого процесса используются элементы методологии SADT, обобщающие основные процедурные аспекты существующих работ, посвященных данной предметной области. Выделяются основные группы сущностей, используемых для формирования типовых методик обнаружения вредоносных программ на основе данной группы методов*

Ключевые слова: *интеллектуальная обработка данных, вредоносные программы, обнаружение.*

INTELLIGENT DATA ANALYSIS FOR MALWARE DETECTION

Dmitry V. Komashinskiy, Igor V. Kotenko

Laboratory of computer security problems SPIIRAS
39, 14th Liniya, St. Petersburg, 199178, Russia
komashinskiy@comsec.spb.ru, ivkote@comsec.spb.ru,
http://comsec.spb.ru/en

Abstract: *The paper considers a state-of-the-art survey of systems for malware detection and identification based on intelligent data analysis. The SADT methodology was adopted to formalize this process in order to generalize common procedural aspects described in the analyzed papers within the area. The set of basic abstract items specifying the essence of each concrete approach to detect malware is emphasized.*

Keywords: *Intelligent data analysis, malware, detection.*

1. ВВЕДЕНИЕ

В настоящее время одной из актуальных задач в области выявления вредоносных программных объектов (далее используется аббревиатура ВП от «вредоносная программа», англ. malware) является применение методов интеллектуального анализа данных (ИАД, англ. Intelligent Data Analysis, IDA).

Концепция применения ИАД для обнаружения вредоносных программ была сформулирована Кефартом [3] и др. в 90-х годах прошлого века и получила практическое продолжение в работе Столфо, Шульца и др. [20] в 2001 году. К настоящему времени очевидно, что теоретические изыскания в данной предметной области успешно стимулируют разработку и интеграцию практических методик противодействия ВП [4, 16, 25]. Более того, наблюдается тенденция усложнения задач в данной предметной области в соответствии с текущими вызовами, определяемыми эволюцией ВП. К примеру, Масуд и др. [26] рассматривают проблему выявления ВП как задачу анализа непрерывного и изменчивого во времени потока данных, что дает понимание современных тенденций в данной предметной области.

Несмотря на наличие ряда ценных результатов, перед исследователями возникают новые, все более сложные задачи, направленные

на формирование систем обнаружения и идентификации вредоносных программ, оптимальных с точки зрении учета требований к характеристикам точности, производительности и ресурсопотребления.

Авторам представляется, что в свете наличия большого количества работ в данной области достаточно актуальной задачей является проведение их анализа, который позволит исследователям принимать более эффективные решения.

Структура статьи представляется следующим образом.

В первом разделе предлагается формальное представление процессов использования методов ИАД для построения систем обнаружения и идентификации вредоносных программ.

Во втором разделе раскрываются основы существующих подходов к выявлению вредоносных программ на примере файловых объектов формата Portable Executable (PE32), являющихся основным средством распространения вредоносных программ для операционных систем Ms Windows.

В третьем разделе рассматриваются отдельные статические и динамические подходы к выявлению вредоносных программ на основе ИАД, а также методы их комбинирования.

Итоги обзора подводятся в заключении.

2. ФОРМАЛЬНОЕ ОПИСАНИЕ ПРОЦЕССОВ ИСПОЛЬЗОВАНИЯ МЕТОДОВ ИАД

Процесс обучения системы обнаружения ВП (рис. 1) может быть представлен как последовательность действий (цепочка подпроцессов)

$$M_{Learn} = [P_{Ext}, P_{Sel}, P_{Learn}, P_{Val}], \qquad (1)$$

обеспечивающая получение из обучающего набора данных D_{Train} целевой модели M_{Dec}, построенной на основе пространства атрибутов $A_{Dec} = \{a_1, a_2, \dots, a_n\}$.

В дальнейшем пространство атрибутов A_{Dec} и работающая в его рамках целевая модель M_{Dec} используются в процессе эксплуатации системы (рис. 2).

Выделенные элементы процесса обучения данных систем могут быть охарактеризованы следующим образом.

Подпроцесс извлечения признаков P_{Ext} обеспечивает формирование начального пространства атрибутов $A_{Raw} = \{a_1, a_2, \dots, a_r\}$, в рамках которого формируется описание $D_{Raw} = \{d_1^R, d_2^R, \dots, d_m^R\}$ каждого элемента входного набора данных $D_{Train} = \{d_1^T, d_2^T, \dots, d_m^T\}$.

Формирование A_{Raw} осуществляется за счет использования предварительно выбранной исследователем модели представления объектов M_{View}, целью которой является обеспечение единого подхода к рассмотрению элементов множества D_{Train} в рамках выделенного аспекта, обеспечивая тем самым преобразование M_{View} : $D_{Train} \rightarrow A_{Raw}$.

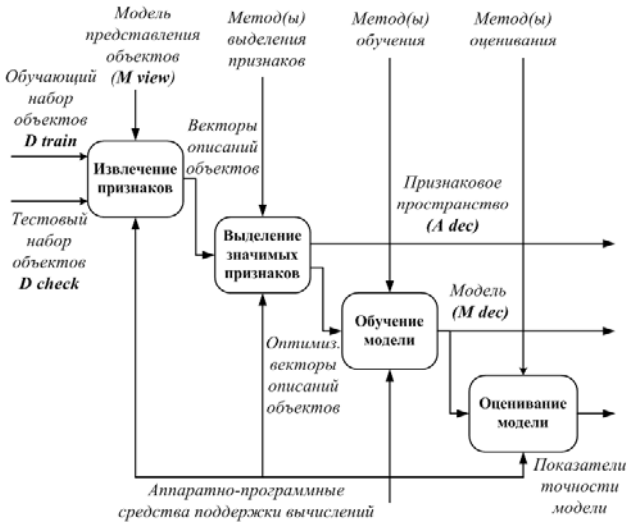


Рис. 1 – Модель процесса обучения систем обнаружения ВП

Задачей подпроцесса выделения значимых признаков P_{Sel} является оптимизация размерности и эффективности использования начального пространства атрибутов A_{Raw}, на основе которого построено множество описаний элементов входного набора данных D_{Raw} с получением новых пространства атрибутов A_{Dec} и описания входного набора данных D_{Dec}.

Рис. 2 – Модель процесса эксплуатации систем обнаружения ВП

На практике задача оптимизации может быть связана как с выделением A_{Dec} из A_{Raw}, так и с полным или частичным формированием множества A_{Dec} на основе атрибутов, не

входящих в начальное пространство A_{Raw}. Часто в интересах минимизации временных и ресурсных затрат задача конструирования признаков по умолчанию сводится к формированию модели представления объектов M_{View} таким образом, чтобы оптимизация начального пространства атрибутов не требовала формирования новых.

Это объясняет фокус данного процесса на использовании определенной при подготовке эксперимента процедуры $M_{Sel}: A_{Raw} \rightarrow A_{Dec}$, $A_{Dec} \subset A_{Raw}$ выделения значимых признаков A_{Dec} из числа существующих A_{Raw}. Подпроцесс обучения модели P_{Learn} является основополагающим во всем процессе обучения системы и обеспечивает формирование целевой модели M_{Dec} на основе использования оптимизированного набора данных D_{Dec} применительно к выбранному исследователем методу обучения. Подпроцесс оценивания модели P_{Val} предоставляет возможность получить количественную оценку качества предиктивной способности полученной на предыдущем шаге целевой модели M_{Dec} на основе тестового набора данных $D_{Check} = \{d_1^C, d_2^C, \dots, d_m^C\}$.

Процесс эксплуатации эвристической системы обнаружения РПВ может быть представлен как цепочка подпроцессов)

$$M_{Func} = [P_{Ext},\ P_{Func}]\,. \qquad (2)$$

Он использует полученные на этапе обучения пространство атрибутов A_{Dec} для формирования оптимизированного представления объектов множества и целевую модель M_{Dec}. Подпроцесс извлечения признаков P_{Ext} по своей сути идентичен аналогичному подпроцессу фазы обучения систем обнаружения РПВ. Его основным отличием является то, что извлекаются признаки, относящиеся только к оптимизированному пространству атрибутов, $M_{View}: D_{Load} \rightarrow A_{Dec}$. Размер множества входящих объектов D_{Load} в данном случае не имеет значения и в общем случае $|D_{Load}| = 1$. Целью подпроцесса эксплуатации целевой модели P_{Func} является формирование метки класса объектов множества D_{Load}.

В сокращенном виде, идея формального представления системы выявления ВП на основе ИАД может быть выражена в виде кортежа

$$S = \langle M_{view}, M_{fs}, M_l, D_{tr}, E \rangle, \qquad (3)$$

где M_{view} – используемые модели представления анализируемых объектов,

M_{fs} – метод(ы) выделения значимых признаков,

M_l – метод(ы) обучения,

D_{tr} – используемые для обучения данные и

E – применяемые программные средства поддержки вычислений.

Полученное представление используется для структурирования настоящей работы.

Модель представления объектов устанавливает формальное представление объектов конкретного типа, которое будет обрабатывать система. Как правило, эта модель включает обобщенную модель объекта и на ее основе вводится понятие признаков, совокупность которых тем или иным образом характеризует каждый уникальный экземпляр объекта;

Методы выделения значимых признаков позволяют произвести минимизацию количества признаков, используемых для описания объектов.

Методы классификации (кластеризации) устанавливают конкретный подход к обучению, используемый в системе.

Используемые *средства поддержки вычислений* позволяют организовать практические аспекты реализации системы;

Источники данных используются при обучении системы и начальной оценке показателей точности решений.

3. ОСНОВНЫЕ ПОДХОДЫ К АНАЛИЗУ ИСПОЛНЯЕМЫХ ФАЙЛОВЫХ ОБЪЕКТОВ

На рис.3 представлена обобщенная последовательность действий, необходимая для получения относительно полного представления о внутренних структурных и функциональных особенностях бинарных исполняемых файлов формата Portable Executable (PE32), до настоящего времени являющегося основным источником угроз для персональных компьютеров.

Как показано на рис. 3, существует две основные группы подходов к анализу исполняемых файловых объектов.

Это подходы, реализующие (1) статические и (2) динамические методики их обработки и принятия решения о степени их вредоносности.

Статические подходы предоставляют более быстрые и менее затратные способы анализа объектов, не требующие их выполнения (или моделирования процесса выполнения) в интерпретирующей их среде.

В зависимости от типа файлов, интерпретирующей средой может операционная система или

какое-либо специализированное программное приложение, например Интернет-браузер.

Основным недостатком данной группы подходов является их неспособность эффективно решать проблему множественности статических представлений объекта, обладающего уникальным поведенческим паттерном (поведением).

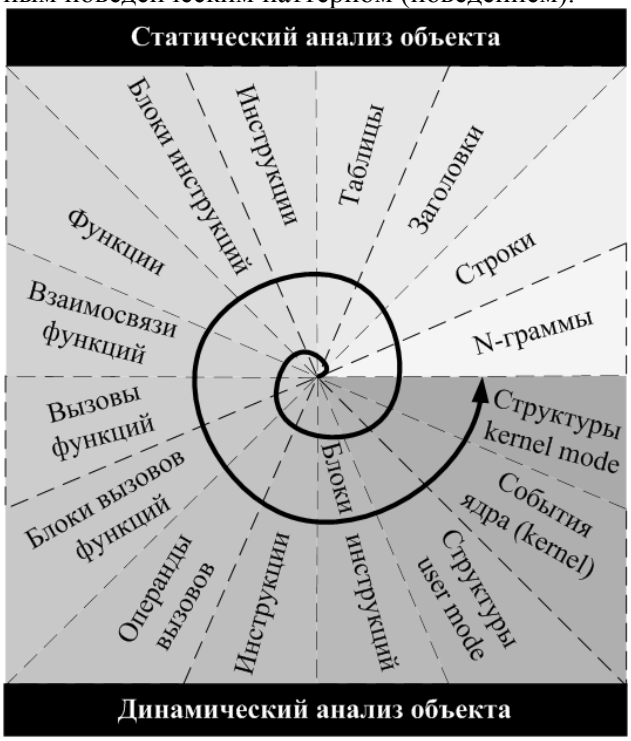


Рис. 3 – Анализ объектов формата Portable Executable

Это ограничивает применимость статических методов и, в итоге, объясняет недостаточную точность при использовании их в отрыве от других подходов. В качестве простейшего примера можно показать, что построение конечного исполняемого файла на основе одного и того же набора исходных файлов (программного проекта) с использованием разного инструментария (разных опций компиляции) позволяет сформировать множество его статически различных функциональных копий. Это, по сути, и приводит к необходимости использовать все более и более сложные методы статического анализа. Данная особенность широко используется злоумышленниками за счет использования различных инструментальных средств компиляции, обфускации, упаковки и программной защиты. При комбинировании данных средств можно с уверенностью утверждать о возможности формирования нескольких десятков тысяч структурно различных копий объекта, обладающего одними и теми же функциональными свойствами.

Группа динамических подходов объединяет более медленные и дорогие с точки зрения используемых вычислительных ресурсов способы получения достоверной информации о функциональности анализируемого объекта. Однако, данные способы позволяют нивелировать проблемы, присущие статическим подходам. Вместе с тем, динамический анализ тоже обладает рядом недостатков.

В первую очередь, как было показано выше, это относится к проблеме соответствия моделируемой при анализе внешней среды (окружения) и ожидаемой (реальной) среды. Например, некоторые вредоносные приложения выполняют явные деструктивные действия только при наличии условий, определенных внешней средой. Их отсутствие не позволяет выявить в процессе анализа деструктивный функционал, и, тем самым, может привести к ложному решению аналитика.

Во-вторых, цена разработки и поддержки корректных моделей оказывается иногда непомерно высокой за счет трудоемкости и сложности подобной работы. Как правило, любой программный инструментарий, поддерживающий процесс динамического анализа, имеет определенные недостатки, позволяющие вредоносному коду успешно выявлять их наличие и противостоять им. В качестве примера, можно привести понятия так называемых подходов противодействия отладочным средствам и программным эмуляторам (anti-debugging and anti-emulation tricks).

Наличие проблем в указанных группах подходов и обуславливает природу показанного на рис. 3 итеративного процесса анализа неизвестных объектов от простого к сложному, а также объясняет причину, по которой в настоящее время используются и те, и другие подходы, несмотря на все их недостатки.

4. АНАЛИЗ ИССЛЕДОВАНИЙ

На рис. 4 показано относительное место каждой используемой группы признаков в рамках базиса, определяемого степенью сложности разработки технологии, обеспечивающей их сбор, и средним временем самого процесса сбора.

Например, достаточно очевидно то, что процесс программной реализации простейшего средства сбора N-грамм значительно проще, а получаемое средство универсальнее и быстрее по сравнению с процессом разработки системы дизассемблирования, требующей учета, по крайней мере, поддерживаемых файлового формата и набора команд CPU.Обзор наиболее

Рис. 4 - Показатели процессов сбора признаков

значимых работ в области применения методов ИАД для задачи выявления вредоносных программных объектов проведем в соответствии с основными используемыми группами признаков.

N-граммы. Первой работой, в которой было представлено применение N-грамм для извлечения признаков вредоносного кода, является работа Кепхарта и др. [3], опубликованная в 1995 году и посвященная актуальной на тот момент проблеме выявления зараженных загрузочных секторов. Одной из основополагающих работ, в которых исследовалась эффективность N-грамм для построения системы обнаружения ВП формата PE32, является работа Шульца, Столфо и др. [20]. Они применили данную группу признаков с установленной длиной, равной двум байтам.

В дальнейшем эффективность N-грамм неоднократно обсуждалась другими группами исследователей. Колтер и Малуф [5] в 2004 году продолжили исследование вопроса обнаружения вредоносных PE32-файлов с помощью N-грамм с длиной в диапазоне от 1 до 10 байт, извлеченных из областей, содержащих машинный код. Авторы сообщили, что при выполнении экспериментов они выявили оптимальное для задачи соотношение длинb N-граммы и количества N-грамм в виде 4:500. В 2007 - 2008 годах Масуд, Кхан и др. [12-14] активно использовали N-граммы для построения общего подхода к обнаружению ВП за счет комбинирования групп признаков. В 2009 году Менахем и др. [15] применили данную подгруппу признаков для обоснования применимости иерархических методов комбинирования средств классификации для выявления ВП. В 2010 году в ряде работ [7, 19] N-граммы использовались для продолжения экспериментов. Сантос и др. [19] выполнили исследование применимости метода k ближайших соседей (k Nearest Neighbours, kNN) и выявления оптимальных значений N и k для задачи выявления ВП. Другая работа [7] была посвящена актуальной проблеме минимизации начального количества N-грамм за счет введения понятия позиции N-граммы в рамках значимого региона анализируемого объекта.

Одним из важнейших преимуществ N-грамм, как признаков ВП, является простота процесса их извлечения и наглядность. Вместе с тем, данная группа признаков обладает и существенными недостатками, определяемыми в первую очередь их потенциально большим количеством [7], что усложняет проведение дальнейших процедур выделения значимых признаков. Кроме того, системы, использующие только данную группу признаков, уязвимы по отношению к использованию разнообразных методов обфускации контента (например, упаковка, установка дополнительных защитных программных слоев для усложнения реверсинга и т.д.).

Строки. Строковые данные, также часто упоминаемые как интерпретируемые (читаемые) строки, можно рассматривать в качестве группы признаков, семантически близкой к рассмотренным выше байтовым

последовательностям (N-граммам). Их основным отличиями следует считать (1) переменную длину, (2) принадлежность значений символов строки к определенному диапазону значений, определяющему алфавит, и (3) наличие терминирующего символа, завершающего строку. Как правило, значение последнего зависит от контекста процедуры извлечения признаков. Так, например, для процедуры анализа файлов формата PE32 терминирующий символ исторически связан с понятием нулевого символа, определяющего завершение строки (C-строки).

Уже упомянутую работу Шульца [20] можно считать первой работой, официально использовавшей строковые данные для построения систем обнаружения ВП на основе ИАД. Несмотря на это, следует отметить, что подобные признаки активно использовались в антивирусной индустрии и ранее для формирования сигнатур. Колтер и Малуф [5] на практике показали, что очень часто наиболее значимые для процесса детектирования ВП N-граммы соответствуют определенным последовательностям символом, относящимся к строкам. В настоящее время в силу определенных недостатков статические строковые признаки в меньшей степени значимы при проведении исследований, однако все равно принимаются во внимание. Например, Лу и другие [11] используют строковые признаки в качестве одной из групп признаков при построении комбинированной схемы методов для обнаружения ВП.

К достоинствам строк, как признаков, следует отнести указанные выше преимущества использования N-грамм и интерпретируемость. Основными недостатками строк являются проблемы, присущие N-граммам.

Заголовки и таблицы. Данные заголовков исполняемых файлов являются важным источником информации для систем обсуждаемого класса. Шульц и др. [20] применяют данные таблиц импорта для формирования подмножества признаков, несущих информацию об используемых библиотечных вызовах функций операционной системы. Менахем и др. [15] используют значения заголовков PE32 и информацию директорий импорта, экспорта и ресурсов для построения многослойной иерархической системы классификации, основанной на применении признаков различных групп. Пердиски и др. [17] используют данные заголовков для формирования набора производных признаков (таких, как энтропия секций). Маттик [16] использует числовые данные заголовков и некоторые производные признаки для анализа применимости статического анализа. Шахзад и др. [22] производит анализ заголовочных структур исполняемых файлов формата ELF (используемых в Unix-подобных ОС) и тем самым обосновывает применимость подходов, традиционно используемых для структурного анализа исполняемых файлов ОС Windows (PE32) для статического обнаружения вредоносных файлов на других платформах. Основным достоинством данной группы признаков является относительная простота их извлечения и интерпретации. Использование данных этого уровня, в конечном счете, позволяет получить значительную долю информации о начальном состоянии объекта непосредственно перед началом его интерпретации исполняющей средой. Например, заголовочные и табличные данные файлового формата PE32 позволяют с достаточной точностью судить о начальном состоянии адресного пространства процессов, его выполняющих. Вместе с тем, наличие широкого спектра методов противодействия статическому анализу на практике не всегда позволяет в полной мере воспользоваться преимуществами заголовочных и табличных данных.

Инструкции и их совокупности. Группы признаков, извлекаемых из анализируемых объектов на уровне анализа вложенного программного кода (для исполняемых форматов это, как правило, достигается дизассемблированием), позволяют получать низкоуровневую статическую информацию, описывающую поведенческие аспекты, дающие представление об отдельных управляющих командах и их совокупностях (последовательностях команд и их логических объединений). Йе и др. [24] используют производные частотные характеристики наличия машинных команд и их последовательностей в рамках отдельных функциональных блоков. Масуд и др. [12-14] применяют последовательности описаний машинных команд (семантический тип операции и типы аргументов). Сиддики и др. [23] предлагают подход к формированию описаний последовательностей машинных команд на основе значений их опкодов. Алазаб [1] и др. используют статические данные о вызовах функций для улучшения показателей системы выявления ВП на основе N-грамм. Кинэйбл и Костакис [4] предлагают методику, использующую статические данные о связях совокупностей инструкций (функций) для подготовки статического графа вызовов функций, используемого для оценки подобия ранее неизвестных объектов существующим описаниям вредоносных. Данные группы признаков

являются наиболее информативными по сравнению с описанными выше и позволяют достигнуть максимальной (по сравнению с другими статическими типами данных) точности принятия решения. Вместе с тем, их очевидным недостатком является достаточная сложность обработки и невозможность их полноценного использования в условиях широкого применения злоумышленниками средств программной защиты и упаковки.

Динамические признаки. Авторы данного обзора представили подход [8] к выявлению вредоносных программ на основании использования ими в процессе работы системных сервисов ядра для ОС Windows XP. Использовались числовые и категориальные производные признаки, характеризующие, например, количество вызовов и типы доступа к отдельным объектам контролируемой системы. В отличие от этого, Йе и др. [25] предложили использовать в качестве признаков последовательность перехватываемых вызовов.

Ланци и др. [10] для построения формализованного описания процесса выполнения контролируемого приложения используют так называемые «поведенческие N-граммы», формируемые методом прохода «плавающего окна» заданной длины N по наблюдаемой последовательности вызовов системных сервисов. Они показали, что наиболее эффективные значения N лежат в интервале от 2 до 4 включительно. Примечательно то, что логика сбора последовательности вызовов опирается на понятие контроля доступа к критическим объектам системы (отдельные файлы, ключи и значения системного реестра).

Рик и др. [18] для описания выполняемых приложений применили символьные цепочки, характеризующие семантику выполняемых системных вызовов и их аргументы. Выделение отдельных признаков тоже производилось с помощью метода «плавающего окна».

Шахзад и др. [21] представили подход к выявлению ВП, опирающийся на контроль состояния определенных полей структур ядра, ассоциированных с анализируемыми процессами. Оценка практической применимости подхода была проведена по отношению к ВП, функционирующим на ОС семейства Linux. Было показано, что для успешного выявления некоторых классов ВП достаточно проведения мониторинга структур ядра на начальных стадиях жизненного цикла процессов. С теоретической точки зрения, подход применим и для ОС Windows в рамках систем класса HIPS (Host Intrusion Prevention System).

Исследование применимости формальных представлений выполнения анализируемых объектов на основе мониторинга выполняемых машинных команд проводилось в работах [2] и [6]. Первый подход ориентирован на обнаружение ВП на основе поиска непрерывных цепочек исполняемых инструкций, свойственных вредоносным программ. Вторая работа расширяет данный подход за счет ориентации его на идентификацию отдельных семейств ВП, вводит формальное определение модели анализируемого объекта и использует собственные критерии, задающие понятие цепочки машинных команд и необходимое время их сбора.

Комбинирование классификаторов. Вопрос применения методов комбинирования методов ИАД для обнаружения ВП исследуется, как правило, в контексте задачи повышения точности принятия решения и улучшения показателей эксплуатационных характеристик. В качестве примера работы, освещающей прикладные аспекты комбинирования, можно рассматривать монографию Кунчевой [9].

Работа Менахема и др. [15] посвящена задаче улучшения показателей точности и времени обнаружения ВП за счет использования ансамбля классификаторов. Для формирования пространства признаков применялись статические признаки, полученные на основе извлечения N-грамм в диапазоне от 3-х до 6-ти. Выделение значимых признаков основано на методах Fisher Score и Information Gain, что позволило изначально сократить количество использованных признаков этого класса до 300 5-грамм и 600 6-грамм. Кроме того, в качестве источника информации применялись структурные характеристики файлов. Использовались данные общих заголовков, секций импорта и экспорта, директории ресурсов. Предложена группа признаков, отображающих количественные и качественные характеристики программных функций, содержащихся в анализируемых файлах и отношения объема кода к общему размеру файла и его отдельных частей. После слияния данных групп признаков и проведения повторной оценки их значимости с помощью метода Information Gain был выделен окончательный набор из 140 признаков, использованных в дальнейшей части работы. В качестве основных методов обучения были реализованы методы Majority Voting, Performance Weighting, Distribution Summation, Bayesian Combination, Naive Bayes Combination, Stacking и предложенный авторами метод комбинирования Troika. Метод Troika основан на четырех-уровневой иерархической системе классификаторов, где в качестве базовых классификаторов

используются методы деревьев решений (Decision Tree, DT), kNN, Voting Features Interval (VFI), One Rule (OneR) и наивный Байесовский классификатор (Naive Bayes, NB), а в качестве классификаторов промежуточного слоя - DT и kNN. Оценивание точности рассмотренных методов комбинирования проводилась как для задачи классификации с двума метками, так и для задачи классификации с количеством классов, равным 8 (7 типов ВП и безопасные программы). Оценка результатов на основе 10-кратной кросс-проверки показала высокую точность метода Troika. Однако, этот метод наряду с методом Stacking, требовал существенного количества времени для обучения. Поддержка вычислений осуществлялась с помощью программного пакета Weka.

Йе, Ли и др. [24] исследовали вопрос использования ансамбля методов кластеризации для построения системы автоматической категоризации новых, ранее неизвестных файлов формата PE32.

Основным элементом мотивации работы являлась необходимость снижения времени принятия решения о степени опасности каждого поступающего в систему объекта и его семействе. Обучение решающей модели и ее валидация проводились на данных корпорации KingSoft, представляющих наборы ВП, зарегистрированных на протяжении нескольких суток. В качестве основных признаков использовались статические атрибуты, характеризующие частотные характеристики появления машинных инструкций и наличие цепочек инструкций, входящих в функции, хранимые исследуемыми файлами. Идея построения ансамбля методов кластеризации была основана на использовании двух базовых методов. Первым являлся так называемый метод гибридной иерархической кластеризации, объединяющий преимущества метода иерархической кластеризации, как общего подхода к последовательному объединению кластеров на каждом шаге, и метода k-medoids, как метода разделения полученной совокупности кластеров. Данный метод использовался для обработки статических частных показателей наличия инструкций. Вторым являлся метод Weighted Subspace K-Medoids, помимо прочего решающий задачу минимизации пространства значимых признаков наличия тех или иных последовательностей машинных инструкций в рамках отдельного кластера. Работа выполнялась с использованием собственного набора программного инструментария.

Лу и др. [11] предложили подход к решению проблемы точности обнаружения ВП за счет комбинирования как признаков, так и методов. Идея комбинирования признаков заключалась в использовании объединенного множества статических и динамических признаков, где в качестве статических выступали признаки наличия вызовов системных библиотек вида «Имя библиотеки - Имя функции», а динамические признаки представляли набор потенциально подозрительных действий, выполняемых приложениями. Всего для проведения экспериментов использовались данные из ряда открытых источников. Для выделения набора значимых признаков использовался метод Information Gain. Комбинирование методов основано на применении двухуровневой иерархической модели, объединяющей методы опорных векторов (Support Vector Machine, SVM) и Association Rules. Показано, что предложенный авторами подход способен успешно конкурировать с известными методами комбинирования (в плане повышения точности обнаружения ВП) и имеет сравнимое с другими методами время обучения. Для сравнения использовались базовые методы классификации NB, SVM, kNN, DT, OneR и методы комбинирования Voting, Bagging DT, Boosting DT, Stacking и Grading.

Используемые программные средства поддержки вычислений и источники данных. Анализ работ, представленных в данном разделе, показывает следующие особенности при организации экспериментов.

Как правило, исследователи предпочитают использовать существующие свободные программные средства, реализующие методы ИАД. Наиболее популярным является программный пакет WEKA (Waikato Environment for Knowledge Analysis), используемый во многих упомянутых публикациях [5, 7, 8, 15 – 17, 21]. Помимо этого, в ряде работ [21, 23-25] WEKA используется совместно с библиотекой libSVM, реализующей алгоритмы, реализуемые методом опорных векторов (эта библиотека используется сама по себе в работе Алазаба и др. [1]). Сидикки и др. [23] в своих экспериментах используют возможности языка программирования R. Авторы данного обзора в одной из своих последних работ, посвященной данной теме [6], используют программный пакет RapidMiner, позволяющий объединять собственные возможности со всеми указанными выше программными средствами в рамках единой графической независимой от платформы среды. Похожими возможностями обладает программный пакет KEEL (Knowledge Extraction based on Evolutionary Learning), использованный Шахзадом и др. [22] для дополнения возможностей программного пакета WEKA.

Формирование наборов данных, в частности, вредоносных, для проведения экспериментов является сложной задачей в силу достаточно очевидных причин, главной из которых является то, что источник данных должен заслуживать доверия. Предпочтительным источником являются внутренние хранилища данных компаний, специализирующихся на обнаружении ВП.

В качестве примера подобных работ, выполненных с использованием данных антивирусных компаний, можно привести публикации Кинэйбла и Костакиса (F-Secure) [4], Маттика (McAffee) [16], Йе и др. (KingSoft) [25].

Вместе с тем, используются и другие, более доступные, коллективно поддерживаемые источники данных, например набор коллекций VXHeavens (используются в [2, 5-8, 12-14, 21-23]), данные OffensiveComputing (используется Шахзадом [22]) и Malfease (используется Пердиски [17]).

4. ЗАКЛЮЧЕНИЕ

В статье на основании показанного формализованного представления выделены базовые составляющие, используемые для формирования структуры основной части работы. Процесс группировки использованных в работе публикаций осуществлен в соответствии с введенным понятием модели представления объекта.

Показано, что методы интеллектуального анализа данных могут быть использованы не только в качестве средств формирования систем автоматического выявления вредоносных объектов, но и для оценки эффективности применения новых моделей. Дальнейшая работа по этой теме будет связана с исследованиями систем обнаружения современных Интернет–угроз.

Работа выполняется при финансовой поддержке Министерства образования и науки РФ (контракт 11.519.11.4008), РФФИ (проект №13-01-00843-а), программы фундаментальных исследований ОНИТ РАН (проект №2.2), проектов Евросоюза SecFutur и MASSIF.

5. СПИСОК ЛИТЕРАТУРЫ

[1] M. Alazab, R. Layton, S. Venkataraman, P. Watters, Malware Detection Based on Structural and Behavioural Features of API Calls, *2010 International Cyber Resilience Conference*, Perth, Australia, (23-24 August), pp. 1-10.

[2] J. Dai, R. Guha, J. Lee, Efficient Virus Detection Using Dynamic Instruction Sequences, *Journal of Computers*, (4) 5 (2009), pp. 405-414.

[3] J.O. Kephart, G.B. Sorkin, W.C. Arnold, D.M. Chess, G.J. Tesauro, S.R. White, Biologically inspired defenses against computer viruses, *International Joint Conference on Artificial Intelligence*, Montreal, Canada, (20-25 August 1995), pp.985-996.

[4] J. Kinable, O. Kostakis, Malware Classification Based onCall Graph Clustering, *Journal In Computer Virology*, (7) 4 (2011), pp. 233-245.

[5] J.Z. Kolter, M.A. Maloof, Learning to Detect Malicious Executables in the Wild, *2004 International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA (22-25 August), pp.470-478.

[6] D.V. Komashinskiy, I.V. Kotenko, Using Low-Level Dynamic Attributes for Malware Detection Based on Data Mining Methods, Lecture Notes in Computer Science, Springer-Verlag, Vol. 7531. *2012 International Conference on Mathematical Methods, Models and Architectures for Computer Network Security*, St.Petersburg, Russia (17-20 October), pp.254-269.

[7] D.V. Komashinskiy, I.V. Kotenko, Malware Detection by Data Mining Techniques Based on Positionally Dependent Features, *2010 Euromicro International Conference on Parallel, Distributed and network-based Processing*. Piza, Italy (17-19 February), pp.617-623.

[8] D.V. Komashinskiy, I.V. Kotenko, Integrated Usage of Data Mining Methods for Malware Detection, *2009 International Workshop "Information Fusion and Geographical Information Systems"*. St.Petersburg, Russia (17-20 May). Lecture Notes in Geoinformation and Cartography, Springer, pp.343-357.

[9] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004, 350 p.

[10] A. Lanzi, D. Balzarotti, C. Kruegel, M. Christodorescu, E. Kirda, AccessMiner: Using System-Centric Models for Malware Protection, *2010 ACM conference on Computer and Communication Security*, Chicago, IL, USA (4-8 October), pp.399-412.

[11] Y.-B. Lu, S.-C. Din, C.-F. Zheng, B.-J. Gao, Using Multi-Feature and Classifier Ensembles to Improve Malware Detection, *Journal of Chung Cheng Institute of Technology*, (39) 2 (2010), pp.57-72.

[12] M.M. Masud, L. Khan, B. Thuraisingham, Feature-Based Techniques for Auto-Detection of Novel Email Worms, *2007 Pacific-Asia Conference on Knowledge Discovery and Data*

Mining, Nanjing, China (22-25 May), pp.205-216.

[13] M.M. Masud, L. Khan, B. Thuraisingham, A Hybrid Model to Detect Malicious Executables, *2007 IEEE International Conference on Communications,* Glasgow, Scotland (24-28 June), pp.1443-1448.

[14] M.M. Masud, L. Khan, B. Thuraisingham, *Data Mining Tools for Malware Detection*. CRC Press Taylor & Francis Group, 2012. 450 p.

[15] E. Menahem, A. Shabtai, L. Rokach, Y. Elovici, Improving Malware Detection by Applying Multi-Inducer Ensemble, *Journal Computational Statistics & Data Analysis* (53) 4 (2009), pp.1483-1494.

[16] I. Muttik, Malware Mining, *2011 Virus Bulletin Conference*, Barcelona, Spain, (5-7 October), pp.46-51.

[17] R. Perdisci, A. Lanzi, W. Lee, McBoost: Boosting scalability in malware collection and analysis using statistical classification of executables, *2008 Computer Security Applications Conference*, Anahem, CA, USA, (8-12 December), pp.301-310.

[18] K. Rieck, T. Holz, C. Willems, P. Dussel, P. Laskov, Learning and Classification of Malware Behavior, *2008 International conference on Detection of Intrusions and Malware and Vulnerability Assessment*, Paris, France (10-11 July), pp.108-125.

[19] I. Santos, Y. Penya, J. Devesa, P. Bringas, N-grams-based File Signatures for Malware Detection, *2009 International Conference on Enterprise Information Systems*, Milan, Italy (6-10 May), pp.317-320.

[20] M. Schultz, E. Eskin, E. Zadok, S. Stolfo, Data Mining Methods for Detection of New Malicious Executables, *2001 IEEE Symposium on Security and Privacy*, Oakland, CA, USA (13-16 May), pp. 38-49.

[21] F. Shahzad, S. Bhatti, M. Shahzad, M. Farooq, In-Execution Malware Detection using Task Structures of Linux Processes, *2011 IEEE International Conference on Communications*, Kyoto, Japan (5-9 June), pp.1-6.

[22] F. Shahzad, M. Farooq, ELF-Miner: Using Structural Knowledge and Data Mining Methods to Detect New (Linux) Malicious Executables, *Journal of Knowledge and Information Systems*, (30) 3 (2012) pp.589-612.

[23] M. Siddiqui, M. Wang, J. Lee, Detecting Internet Worms Using Data Mining Techniques, *Journal of Systemics, Cybernetics and Informatics*, (6) 6 (2008), pp. 48-53.

[24] Y. Ye, T. Li, Y. Chen, Q. Jiang, Automatic Malware Categorization Using Cluster Ensemble, *2010 ACM International Conference on Knowledge discovery and data mining*, Washington, USA (25-28 July), pp.95-104.

[25] Y. Ye, T. Li, K. Huang, Q. Jiang, Y. Chen, Hierarchical associative classifier (HAC) for malware detection from the large and imbalanced gray list, *Journal of Intelligent Information Systems* (35) 1 (2010), pp.1-20.

[26] M. Masud, T. Al - Khateeb, K. Hamlen, L. Khan, J. Han, B. Thuraisingham, Cloud-Based Malware Detection for Evolving Data Streams, *In Journal ACM Transactions on Management Information Systems*, (2) 3 (2011), article 16, 27 p.

***Дмитрий Владимирович Комашинский**, аспирант лаборатории проблем компьютерной безопасности СПИИРАН. В 2000 году окончил Санкт-Петербургское Высшее Военное инженерное училище связи им. Ленсовета. В область научных интересов входят безопасность компьютерных сетей, защита программного обеспечения, обнаружение компьютерных атак, защита от вирусов и сетевых червей, интеллектуальный анализ данных, разработка программного обеспечения.*

***Игорь Витальевич Котенко**, доктор технических наук (1999), профессор. Заведующий лабораторией проблем компьютерной безопасности СПИИРАН. Автор более 450 научных работ. Область научных интересов – информационная безопасность, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, ложные информационные системы, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму; искусственный интеллект, в том числе многоагентные системы, мягкие и эволюционные вычисления, машинное обучение.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

INTELLIGENT DATA ANALYSIS FOR MALWARE DETECTION

Dmitry V. Komashinskiy, Igor V. Kotenko

Laboratory of computer security problems SPIIRAS
39, 14th Liniya, St. Petersburg, 199178, Russia
komashinskiy@comsec.spb.ru, ivkote@comsec.spb.ru,
http://comsec.spb.ru/en

Abstract: *The paper considers a state-of-the-art survey of systems for malware detection and identification based on intelligent data analysis. The SADT methodology was adopted to formalize this process in order to generalize common procedural aspects described in the analyzed papers within the area. The set of basic abstract items specifying the essence of each concrete approach to detect malware is emphasized.*

Keywords: *Intelligent data analysis, malware, detection.*

1. INTRODUCTION

The research is dedicated to issues of using Intelligent Data Analysis (IDA) techniques for designing systems able to detect and identify malicious executable binaries (malware) automatically.

The paper comprises three main structural parts named as follows: “The formal description of using Intelligent Data Analysis techniques”, “Primary ways on analyzing executable binaries” and “State of the art”.

2. FORMAL DESCRIPTION OF USING DATA MINING TECHNIQUES

The first part introduces main results for SADT – based modelling of the learning and functioning processes of systems aimed at automatic malware counteraction (Fig.1).

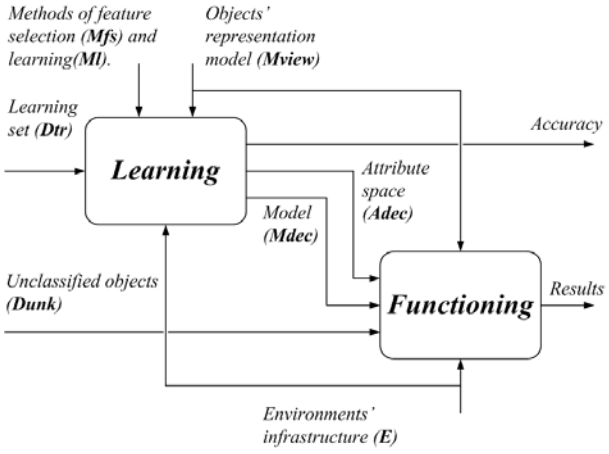


Fig.1 – High level representation of the model

The formal description of such systems is expressed by following statement:

$$S = \langle M_{view}, M_{fs}, M_l, D_{tr}, E \rangle, \qquad (1)$$

where M_{view} – used representation model(s) of analyzed objects,

M_{fs} – method(s) to select valuable features,

M_l – used learning method(s),

D_{tr} – used data for learning and validation and

E – adopted software for preparing necessary environment.

3. ANALYZING EXECUTABLE BINARIES

The second part generalizes essential aspects of modern approaches to executable malware’s detection and identification.

The main static and dynamic groups of malware attributes (features) are distinguished and described.

The main purpose of the section is to emphasize the interconnection of the introduced above representation model’s concept M_{view} with listed in this part groups of attributes which are most valuable for malware detection process.

4. RELATED WORK

The third part of the research is organized as a survey of a number of existing publications [1-14] devoted to the topic of using Intelligent Data

Analysis methods for detecting and identifying executable malware samples.

The survey's structure is based on the formal description posed above (statement 1) and considers the following issues:

(1) static [2-10, 12, 13] and dynamic [1, 11, 14] detection of malicious software; in particular, the practices of combining representation models and learning methods [5, 8, 13],

(2) adopted Intelligent Data Analysis software kits and

(3) used sources of data for performing learning and validation procedures.

The survey discusses different types of attributes and their validity for designing IDA – based automatic malware detection systems.

It goes through following static attributes types:

- n-grams [2, 10],
- strings [4, 5, 10],
- headers and tables [8-10],
- instructions and their sequences [6, 7, 12, 13],
- static call dependency graphs [3].

The similar approach is used for considering dynamic attributes' feasibility for preparing such automatic decision making systems at levels of machine instructions [1], operating system's calls [14] and structures [11].

The paper also encompasses main sets of Intelligent Data Analysis tools used nowadays by researchers for arranging proper experimental job (WEKA, libSVM, RapidMiner) and touches on the issu of data sources.

This research is being supported by grant of the Russian Foundation of Basic Research (project #13-01-00843-a), Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (contract #2.2), State contract #11.519.11.4008 and partly funded by the EU as part of the SecFutur and MASSIF projects.

5. REFERENCES

[1] J. Dai, R. Guha, J. Lee, Efficient Virus Detection Using Dynamic Instruction Sequences, *Journal Of Computers*, (4) 5 (2009), pp. 405-414.

[2] J.O. Kephart, G.B. Sorkin, W.C. Arnold, D.M. Chess, G.J. Tesauro, S.R. White, Biologically inspired defenses against computer viruses, *1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, (20-25 August), pp.985-996.

[3] J. Kinable, O. Kostakis, Malware Classification Based onCall Graph Clustering, *Journal In Computer Virology*, (7) 4 (2011), pp. 233-245.

[4] J.Z. Kolter, M.A. Maloof, Learning to Detect Malicious Executables in the Wild, *2004 International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA (22-25 August), pp.470-478.

[5] Y.-B. Lu, S.-C. Din, C.-F. Zheng, B.-J. Gao, Using Multi-Feature and Classifier Ensembles to Improve Malware Detection, *Journal of Chung Cheng Institute of Technology*, (39) 2 (2010), pp.57-72.

[6] M.M. Masud, L. Khan, B. Thuraisingham, Feature-Based Techniques for Auto-Detection of Novel Email Worms, *2007 Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Nanjing, China (22-25 May), pp.205-216.

[7] M.M. Masud, L. Khan, B. Thuraisingham, A Hybrid Model to Detect Malicious Executables, *2007 IEEE International Conference on Communications*, Glasgow, Scotland (24-28 June), pp.1443-1448.

[8] E. Menahem, A. Shabtai, L. Rokach, Y. Elovici, Improving Malware Detection by Applying Multi-Inducer Ensemble, *Journal Computational Statistics & Data Analysis* (53) 4 (2009), pp.1483-1494.

[9] I. Muttik, Malware Mining, *2011 Virus Bulletin Conference*, Barcelona, Spain, (5-7 October), pp.46-51.

[10] M. Schultz, E. Eskin, E. Zadok, S. Stolfo, Data Mining Methods for Detection of New Malicious Executables, *2001 IEEE Symposium on Security and Privacy*, Oakland, CA, USA (13-16 May), pp. 38-49.

[11] F. Shahzad, S. Bhatti, M. Shahzad, M. Farooq, In-Execution Malware Detection using Task Structures of Linux Processes, *2011 IEEE International Conference on Communications*, Kyoto, Japan (5-9 June), pp.1-6.

[12] M. Siddiqui, M. Wang, J. Lee, Detecting Internet Worms Using Data Mining Techniques, *Journal of Systemics, Cybernetics and Informatics*, (6) 6 (2008), pp. 48-53.

[13] Y. Ye, T. Li, Y. Chen, Q. Jiang, Automatic Malware Categorization Using Cluster Ensemble, *2010 ACM International Conference on Knowledge discovery and data mining*, Washington, USA (25-28 July), pp.95-104.

[14] Y. Ye, T. Li, K. Huang, Q. Jiang, Y. Chen, Hierarchical associative classifier (HAC) for malware detection from the large and imbalanced gray list, *Journal of Intelligent Information Systems* (35) 1 (2010), pp.1-20.

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

TOWARDS DATA PERSISTENCY IN REAL-TIME ONLINE INTERACTIVE APPLICATIONS

Max Knemeyer, Mohammed Nsaif, Frank Glinka, Alexander Ploss, Sergei Gorlatch

University of Muenster, Germany

Abstract: *The class of distributed Real-time Online Interactive Applications (ROIA) includes such important applications as Massively Multiplayer Online Games (MMOGs), as well as interactive e-Learning and simulation systems. These applications usually work in a persistent environment (also called world) which continues to exist and evolve also while the user is offline and away from the application. The challenge is how to efficiently make the world and the player characters persistent in the system over time. In this paper, we deal with storing persistent data of real-time interactive applications in modern relational databases. We analyze the major requirements to a system for persistency and we describe a preliminary design of the Entity Persistence Module (EPM) middleware which liberates the application developer from writing and maintaining complex and error-prone code for persistent data management. EPM automatically performs the mapping operations to store/retrieve the complex data to/from different types of relational databases, supports the management of persistent data in memory, and integrates it into the main loop of the ROIA client-server architecture.*

Keywords: *Massively multiplayer online games (MMOG); persistency; virtual worlds; object-relational mapping; real-time applications.*

1. INTRODUCTION

Distributed real-time online interactive applications (ROIA) can potentially be used simultaneously by thousands of users. They make high demands on availability, responsiveness and scalability. The probably most demanding applications of this type are Massively Multiplayer Online Games (MMOGs). The number of users in this area increases sharply in recent years: the most successful Massively Multiplayer Online Role Playing Game (MMORPG) is the "World of Warcraft" [1] with more than 11 millions of active individual players. To manage the huge amount of involved data, data in such games are often stored in relational databases which are based on a solid and mature technology. For example, World of Warcraft [1], Guild Wars [2] and the virtual world Second Life [3] employ this technology.

In a MMOG, players stay together in a large virtual world to communicate and interact with each other. The players are being represented by virtual characters called *avatars*. In addition to interacting with other players, an important incentive of a player is to develop his avatar: e.g., the avatar can become equipped with new objects or learn new skills. To make this development persistently, i.e. such that changes are not lost when the game is interrupted, they must be saved (persisted). To increase the reliability of a gaming application, not only the states of avatars should be stored, but also the global state of the game world is usually stored permanently.

Through the actions of the user in the game, his avatar changes or evolves. To avoid losing the recent development of the avatar and the new states of the game world, there is a need to store these data persistently.

To store the changed entities, a system for persistent data storage is required. The persistent data storage is used when the player enters the virtual world at arbitrary time, such that all previous changes become available again. The saving is usually made at so-called *key points* of the game, for example, when the avatar completes a specific mission, acquires new objects or learns new skills. To increase reliability, it must be possible to save the changes of the game world continuously. Persistent data management takes a significant part of the code of a game, up to 40% [4]. Writing and maintaining this code is complex and error-prone, especially if new features are added to the game. The persistence code is usually tailored to a specific application use case, and thus poorly reusable. Therefore, providing generic solution that supports the game developers in this task is desirable.

In this paper, we discuss the problem of efficiently storing the persistent data of real-time interactive applications. We target applications which a) are developed in C++, the programming language used for most ROIA, and b) store their complex data in relational database management systems (RDMS). As the result of our analysis, we present a preliminary design of our persistency system – the Entity Persistence Module (EPM) – which we design as a middleware, i.e. a software layer that connect the application with different types of relational databases. We also describe how EPM provides the application developer with a programming interface (API) in order to simplify the use of the presented persistency system.

In this paper, we discuss the problem of efficiently storing the persistent data of real-time interactive applications. We target applications which a) are developed in C++, the programming language used for most ROIA, and b) store their complex data in relational database management systems (RDMS). As the result of our analysis, we present a preliminary design of our persistency system – the *Entity Persistence Module* (EPM) – which we design as a middleware, i.e. a software layer that connect the application with different types of relational databases. We also describe how EPM provides the application developer with a programming interface (API) in order to simplify the use of the presented persistency system.

The paper is organized as follows. In Section II, we present basic fundamentals about the MMOG architecture. Section III describes how persistent data can be represented in relational database management systems. In Section IV, we describe and analyze the common approaches of persistence layers. Section V describes the preliminary design of EPM and explains how it works as a middleware software layer.

2. PROPERTIES OF MULTIPLAYER ONLINE GAMES

MMOGs are a class of online games in which thousands of players participate simultaneously in a game by communicating and interacting with each other. This game class has been growing in several distinct categories, such as: Role-Playing Games (RPG), First Person Shooters (FPS), Real-Time Strategy Games, and others. Although each category has its specific game logic, they basically have a similar structure as follows:

- The game comprises a virtual world where players reside and operate.
- The actions of players change the state of the game world, including player avatars, according to the rules of the game logic.
- The game logic dictates what actions are possible and how they affect the game world.

In MMOG, a player with his character, called avatar, moves and interacts with other objects in the game world. All changeable world objects are called dynamic objects or *entities*. These include, for example, computer-controlled characters, weapons, and the avatars of other participants. The entities have different *attributes* which describe them or their state. For example, an avatar may has information about its position in the virtual environment, its life force, its name and carried items. In role-playing games a user can, for example, move an avatar, collect items, and trade with other avatars. Through the actions of the user in the game, his avatar may change or evolve. To avoid losing the recent development of the avatar and the new states of the game world, there is an essential need to storing these data persistently.

To store the changed entities, a system for persistent data storage is required. The persistent data storage is used when the player enters the virtual world at arbitrary time, such that all previous changes become available again. The game developers need such a system in order to save, load and delete entities. The saving is usually made at so-called *key points* of the game, for example, when the avatar completes a specific mission, acquires new objects or learns new skills. To increase reliability, it must be possible to save the changes of the game world continuously.

The basic architecture used for MMOGs is the traditional client-server architecture, enriched with multiple servers. A client is responsible for presenting the game world to a player and interacting with that player. The client takes the inputs from the player and initiates changes in the game world. The server is responsible for the simulation of the game world and updating its state; it is usually called *game server*.

Fig. 1 shows the main functions of the game server, which are realized in the following three steps:

- The game server manages all entities of the virtual game world by continually receiving the actions of the players from the clients and analyzing them (Step 1);
- The new game state is computed by applying the actions of the players and the rules of the game logic to the entities (step 2);
- The new state is sent to the players (step 3).

These three steps run within the game in a loop, called *mainloop*. A single iteration is called a *tick* and the number of cycles per second is called the *tick-rate* of the game. For a smooth gaming experience, it is essential that a certain tick-rate is kept. For example, Quake3 Arena [5] is a fast FPS

game which requires a tick-rate of minimum 20 ticks/second.

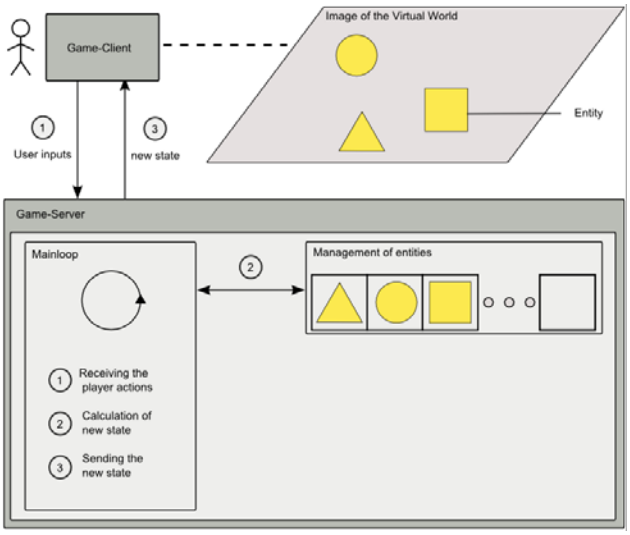


Fig. 1 – Steps of a ROIA mainloop

3. DATABASES FOR GAME PERSISTENCY

Database Management systems (DBMS) are classified according to the way of data representation, i.e., according to the data model of DBMS. The two most popular data models are record-oriented (i.e. relational data model) and object-oriented (i.e. object-oriented data model). The nature of the work environment and the requirements of an application determine which database model is more suitable [6].

An object-oriented DBMS supports complex data stored as objects; it employs a data model with object-oriented features: encapsulation, inheritance, and polymorphism. However, this data model lacks advanced searching facilities, therefore it sometimes called no-query. The underlying model in a Relational DBMS only supports simple data, rather than complex objects, but it strongly supports various advanced searching facilities [7], e.g., the relational SQL. To store the persistent objects of MMOG applications into a database and then retrieve them in an efficient way, we need both these facilities, complex data and query support.

In a relational DBMS, users can query any table in the database and combine related tables using special join functions to include relevant data contained in other tables into the results, and if needed, filter the results. We call this property the *ease of data retrieval*. The relational database model is naturally scalable and extensible, providing a flexible structure to meet changing requirements and increasing amounts of data. The relational model permits changes to the database structure which can be implemented easily without impacting the data or the rest of the database. There is theoretically no limit on the number of rows and columns of tables. In reality, growth and change are limited by the relational database management system (RDBMS) and the hardware used for implementation.

In order to create a relational database, it is necessary to define a *schema*, i.e. its structure described in a formal language supported by the DBMS. It refers to the organization of data and is a blueprint of how a database will be constructed (divided into database tables), i.e. it is a set of formulas (sentences) called *integrity constraints* imposed on a database. These constraints ensure compatibility between the parts of the schema. In relational databases, the schema defines tables, columns or fields, relationships, views, indexes, packages, types, database links, and other elements. In MMOGs, for example, two objects – the avatar and its inventory – are usually presented in two tables (relations), and their properties (attributes) are presented in the columns of these tables. Therefore, the properties of an avatar: AvatarID, Name, PositionX, PositionY, PositionZ, Energy, and InventoryID can be presented as columns in the avatar table, and the properties of inventory: InventoryID, Item1, Item2, and Item3 are presented as columns in the inventory table.

For defining and managing data and data structures in RDBMS, *Structured Query Language* (SQL) is used as a standardized special-purpose programming language. SQL acts as an interface to the RDBMS on the application development side.

Our approach is to develop a middleware for converting the complex data or complex objects of a MMOG application into simple data. Then we can use the relational DBMS as a database for MMOG applications. Relational DBMS are used in most popular MMOG applications, such as Second Life [3], and Guild Wars [2].

Nowadays, most popular multiplayer games, especially MMOGs, are developed using C++, because these modern games have high performance requirements which are best addressed with a relatively low-level, object-oriented programming language. Since the system for persistent data storage, which is presented in this paper, is used in the field of MMOGs, it is also developed in C++.

In order to access a relational database from C++, i.e. use SQL in a C++ program code, there are two main possibilities: native and general database libraries. Database vendors provide native libraries that can be used via a special API to establish a direct connection between the program code and a specific database without any mediation; this is what is called *native connection*. The native libraries that are represented as API wrappers include for example, MySQL++ [8] for MySQL database, and libpqxx [9] for PostgreSql database. A native library

is usually better suited than general database libraries because of its ability to establish a direct connection with database, but the disadvantage is the restriction to a specific database. This may be a problem when using native libraries with MMOGs because the latter need to establish connections to not specified database types that are distributed on multiple servers. The solution is to use a database-independent library (or general database library).

Our approach relies on *Simple Oracle Call Interface*(SOCI) library [10] which allows to access different databases, and at the same time it is a native database library. Fig. 2 shows the modular structure of the SOCI library allowing the integration of different database backends. SOCI makes SQL queries embedded in the regular C++ code, i.e. staying entirely within the standard C++. SOCI is integrated with databases via database backends. The backend forwards the data queries of an application into the appropriate database.

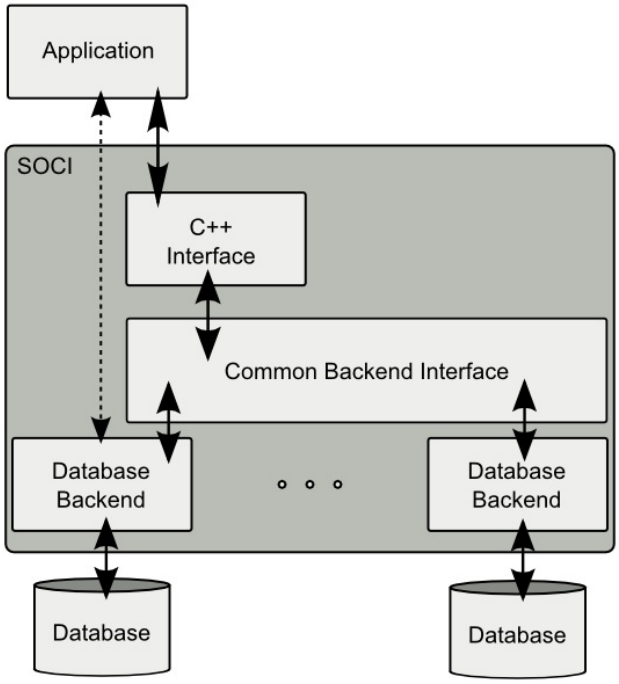


Fig. 2 – SOCI modular structure

The SOCI current version (3.1.0) supports various database types: Oracle, PostgreSQL, MySQL, SQLite3 and Firebird, as well as the generic backend: *Open Database Connectivity* (ODBC). Thus, by using SOCl we can combine the performance advantage of native libraries and the variety database access of ODBC. The SQL commands are passed to the RDBMS without conversion as it happens in ODBC. Additionally, SOCI offers a flexible support for user-defined data types and also an extensive integration with *Boost data types* of many Boost C++ Libraries, i.e. representation of nonstandard data type during storing and retrieving to/from databases. The Boost libraries of C++ are used to store arbitrary information in a variable, e.g. the *Boost.Tuple* library offers the *boost::tuple* class which offers the ability to store a virtually unlimited number of values in one variable in a C++ program.

4. KINDS OF PERSISTENCE SYSTEMS

There are three common approaches to persistence regarding the connection with database:

Database access by means of user classes

In this approach, particular methods for persistence are realized in the classes of users, i.e. the code is used for implementing the access to the database and to SQL commands directly by the classes written by the user. This approach is particularly suitable for small projects.

Database access by data access classes

Here, the code to access the database is placed in additional classes which separate the classes of users from the database. The class instances are called Data Access Objects [11] and are responsible for storing the persistent data of a class. To exchange data between user classes and the Data Access Objects, Data Transfer Objects are used to encapsulate the persistent data which is loaded from/to database. As a result, an additional code is needed in the classes of users to match the persistent data with the Data Transfer Objects. Furthermore, when replacing the employed database by a new one, all the data access classes should be adjusted.

Database access by an abstraction layer

In this approach, the user classes and the communication with database are strictly separated. Fig. 3 shows, as an example, a schematic representation of an abstraction layer to access and store a user class (persistent class) using mapping information. Communication with the database is performed at a central point of the abstraction layer used by all persistent classes of the user. With an abstraction layer, the developer does not write any additional code for database access, but rather defines meta-information which describes the mapping of objects to the database tables. For each user class required to be persistent, such mapping information must be specified. Using this mapping information, the code for database access and the necessary SQL commands are created by the abstraction layer.

This approach of using an abstraction layer (also called *persistency layer*) is used in our system presented in this paper. This approach has significant advantages over the two previous approaches: it is reusable for different projects, and furthermore, it is more easily extendible and customizable. These advantages are especially important for large projects.

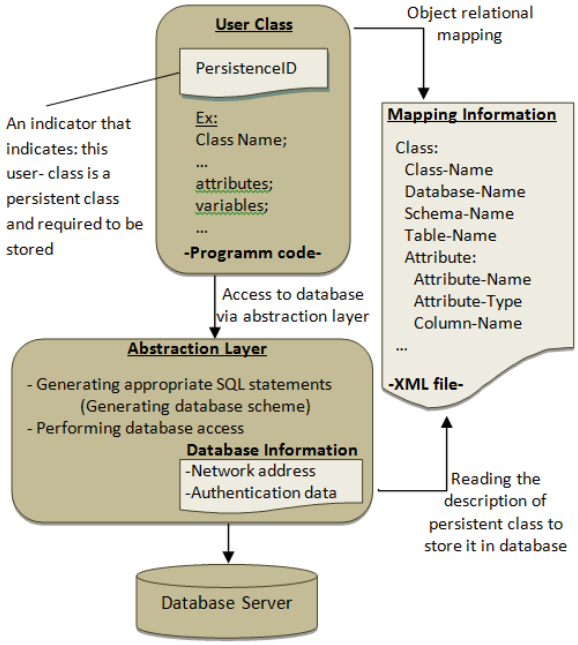


Fig. 3 – Schematic representation of the abstraction layer

The system of persistent data storage is a link between the entities of MMOGs (i.e., the objects of the application) and the RDBMS. Our analysis reveals the following basic requirements towards such a persistency system:

- The system should persist the state of the MMOG game world, i.e. the states of individual objects and entities of an application are continuously stored into a relational DBMS. One of the difficulties here is converting the data between incompatible type systems. The data type of objects is almost always a non-primitive value (composite value), while relational DBMS can only store and manipulate scalar values, (i.e., primitive data types), such as integers and strings that are organized in tables and stored as records. Therefore, the system should be able to convert the object values into groups of simpler values for storing in the database, and then, when the game logic requires it, convert them back upon retrieval from the database without mismatch. This task is usually called *Object-Relational Mapping* [12]. Particularly important is how the attributes of objects and the relationships between objects are stored. The system should support object-oriented concepts, such as inheritance and polymorphism.
- The system should be able to store entities continuously at certain times (e.g., when an avatar gets new objects or learns new skills, or completes a specific task in the game). Not always the entire entity is to be stored; it should be possible to define which particular attributes should be stored. For this purpose, the persistent data management system must provide an appropriate interface for the application developer. This interface will provide an abstraction from the direct interaction with the database, such that the developer does not need to write a database-specific code.
- The system should not be limited to a particular database, but rather be able to work with different RDBMS. Therefore, it must abstract from specific types of databases by providing general supporting interfaces for database connections.

5. THE EPM SYSTEM FOR DATA PERSISTENCY

This section describes the basic concepts and preliminary design of our system for persistent data storage called *Entity Persistence Module* (EPM). EPM serves as an interface between real-time interactive applications and the relational database management systems (RDBMS).

a. Integration of persistence in MMOG

We design our EPM system to work as a software layer between MMOG and RDBMS. To integrate persistence into the complex infrastructure of MMOG applications, EPM resides on two types of servers: login server and game server as shown in Fig 4. The login server checks whether a player is eligible to participate in the game. If necessary, corresponding data of the player, such as name, address, and list of avatars owned by the player is loaded from the database. To play the game, the player retrieves data from the account-database to the game-client; therefore, EPM works between the login server and the account-database. Based on this retrieved data, the game logic determines where the avatar has stopped in the previous game session. When the player exits the game, the game-client is logged out at the login server and the account data is stored in the account-database.

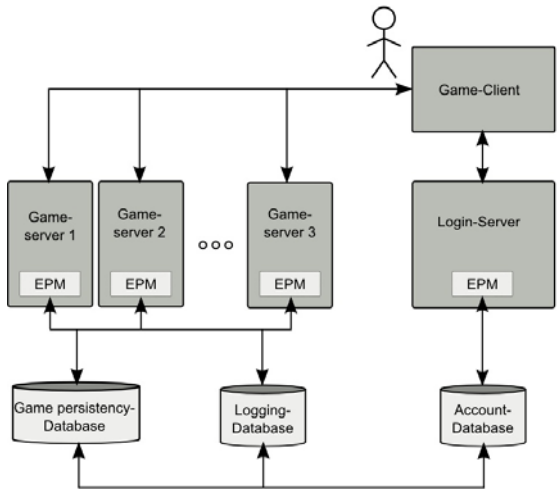


Fig. 4 – Integration of EPM in the architecture of an MMOG

Fig. 5 shows the integration of EPM with the mainloop performing the continuous processing of the game state. The mainloop receives the actions of players from the clients and analyzes them (Fig. 5 step 1), then it calculates the new state (step 2), which is persisted using EPM (step 3), and then sent back to the clients (step 4).

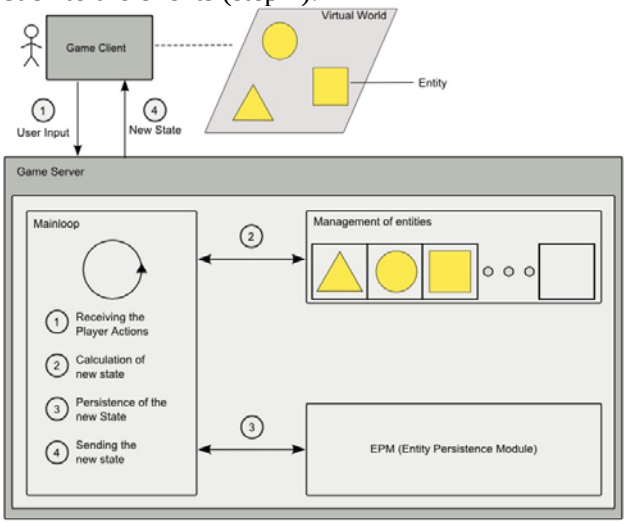


Fig. 5 – Integration of EPM in the application mainloop

b. Architecture of the persistence module

The essential part of the EPM architecture comprises the software components in the gray rectangle in the middle of Fig. 6.The bottom side of the figure represents the database side, while the upper side represents the application side as well as mapping and database information files. These files are the supplementary part of the EPM which allows the essential part of EPM to work in an efficient way. The architecture of EPM follows some ideas of an abstract design in [13].

c. Information about persistency

In order to allow for our persistency layer – the EPM system – to establish a connection between the persistent classes of MMOGs and one or more RDBMS, EPM needs meta-information about: 1) the classes which need persistency, and 2) the desirable database for storing. This meta-information is provided by the application developer and presented to EPM by: *mapping file, PersistenceID, datatype mapping, and database config.*

Mapping File

Mapping files describe where and how persistent objectsare stored. EPM provides its own XML representation of the data structures of mapping information as XML files. The mapping information specifies which classes and which attributes should be persistent and where they should be stored.

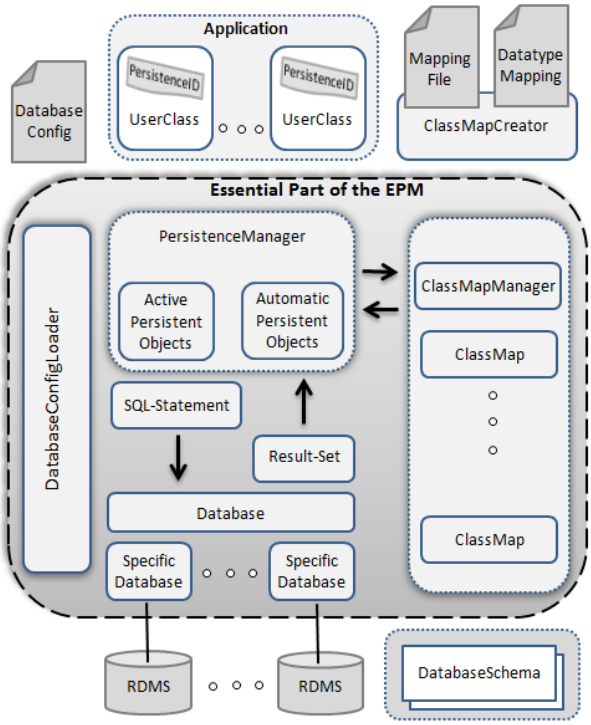


Fig. 6 – Architecture Overview of the Entity Persistence Module (EPM)

Listing 1 shows an example of XML mapping file that contains a part of the mapping information for a base class called Avatar with one of its attributes. This attribute in the example is the name of avatar. The first four lines define the mapping information of the avatar class, the given name for persistent class is specified in line 2, then the name of the database in line 3, the database schema in line 4, and the table in which the avatars will be stored in line 5. After we have accessed a specific table of a specific database depending on the first five lines, we need access to a specific column of that table (Avatar table) to store the specific name of the avatar (line 7). The data type of the name is specified in line 8 and the column in which the name should be stored is specified in line 9.

```
1  <class>
2      <name> Avatar </name>
3      <database> GameDatabase  </database>
4      <schema> gamedatabase </schema>
5      <table> Avatar </table>
6      <attribute>
7          <name> AVname </name>
8          <elementtype> std : : string </elementtype>
9          <columnname> name <columnname>
10     </attribute>
11 </class>
```

Listing 1 – Part of an XML mapping file that describes an avatar class

Shadow information

Shadow information is an additional information added to a persistent object by the application developer, as shown in Fig. 6 within the *UserClass*. This information is required by EPM to manage the persistent objects and is not required by the actual application. An object within an application is uniquely identified by its address in the memory which is only valid as long as the object exists in the memory. Records of a table, in contrast, are uniquely identified by a primary-key that is valid as long as the database exists. Therefore, persistent objects require a unique ID to identify them in the application, as well as in the database. This ID is referred to as EPM PersistenceID, see Fig. 7; it consists of Universally Unique Identifier (UUID) and type information that indicates the type of the persistent object. Shadow information is an indicator of whether the object already exists as a record in the database or not. This information is used to generate an appropriate SQL command to store or update the object. The "insert" SQL command is used if the object has not been stored before, otherwise, the "update" SQL command is needed.

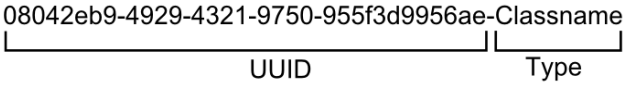


Fig. 7 – Example: presentation of PersistenceID as database key

d. Object-Relational Mapping

One of the most important issues in the design of EPM is *object-relational mapping* that converts complex data and objects of MMOG applications into simple data of primitive types for using a relational DBMS. The software components located on the right side of Fig. 6 are responsible for this. These components work across three stages of the persistence process:

(1) First, after reading the mapping information of persistent classes according to what the game developer specified in MappingFile and DatatypeMapping, the ClassMapCreator creates one ClassMap for each persistent class in the application; the ClassMap component works in the next stage.

(2) The MappingClasses are located within the essential part of EPM as shown in Fig. 6; they consist of two components: ClassMapManager and ClassMap. These components cooperate with the main EPM component (PersistenceManager) to store the persistent objects in an efficient manner. The ClassMapManager manages all the system's ClassMaps and ensures that they are initialized and made available to PersistenceManager. The ClassMap can access all data of an object at run time to generate the appropriate SQL commands, and thus the current state of an object becomes ready for storing in a relational database. These SQL commands are encapsulated within the essential part of EPM and used in the next stage.

(3) Finally, the persistent object is sent to permanent data storage (relational database). After obtaining the SQL statements from ClassMap, the DatabaseSchema defines the necessary tables, columns, relationships, data types, database links, and other elements which are necessary to store the persistent object.

With regard to the mainloop, our strategy with the mapping components focuses on separating the SQL generation from SQL execution. Thus, the ClassMap interrupts the mainloop to generate the SQL statements, and thereafter, the DatabaseSchema can execute the SQL statements concurrently with the mainloop (as a separate thread).

In sophisticated applications, most objects have one or more relationships with other objects. To avoid their separate storing, the ClassMap performs transitive persistence by storing the object with all of its associated objects to the database automatically. This recursive process of storing is called *cascading*. For example, the ClassMapAvatar of *avatar object* will cascade the storing operation to its associated object (*inventory object*). The (*save: Inventory*) task, included within ClassMapAvatar, holds mapping information that indicates a relationship between *avatar* and *inventory*. In EPM, any entity is automatically saved, loaded, and deleted to/from database, together with its associated objects.

e. Connection with databases

To connect with various RDBMS during the establishment of persistence, EPM provides a standard database interface, with different configuration possibilities for the game developer. The database interface accepts SQL statements and returns query results as result sets which are actually an object-oriented representation of relations. The RDBMS then stores the rows and columns that are represented in these result sets.

Listing 2 shows an excerpt from the EPM's database interface. The methods in line 3 and 4 initialize the database connection, i.e. opening and closing a connection with a specific database. The *getName()* method in line 5 identifies the specific name of the database with which the connection is established. The methods in lines 7 through 10 insert, retrieve, update, and delete data to/from database, correspondingly.

```
1  class database {
2  // methods to initialized the database connection
3  virtual void open ( ) = 0;
4  virtual void close ( ) = 0;
5  std : : string getName ( ) { return this -> name ; }
6  virtual Resultset * processSql (std : : string &) = 0 ;
7  virtual void processSql ( InsertSqlStatement &) = 0 ;
8  virtual ResultSet * processSql (SelectSqlStatement &) = 0 ;
9  virtual void processSql ( UpdateSqlStatement &) = 0 ;
10 virtual void processSql ( DeletetSqlStatement &) = 0 ; } ;
```

Listing 2 – Excerpt from the database interface

The DataTypeMapping and DatabaseConfig files are provided for the application developer as database configuration files as shown in Fig. 6. Since different databases use different data types to store data, DataTypeMapping file defines which C++ data type is mapped to which data types of the database. The DatabaseConfig file is used to configure a connection with a specific database. For example, this file can include: name of the database, database type, network address of the database server, and the authentication data for logging in; the file is configured during the initialization of the EPM. The database library used by EPM is also specified in this configuration file. Currently, EPM employs the SOCI library [10] which provides different backends supporting connection to various databases.

Fig. 8 shows the sequence of the database connection initialization. Here, the DatabaseConfig-Loader is the component located inside the essential part of EPM.

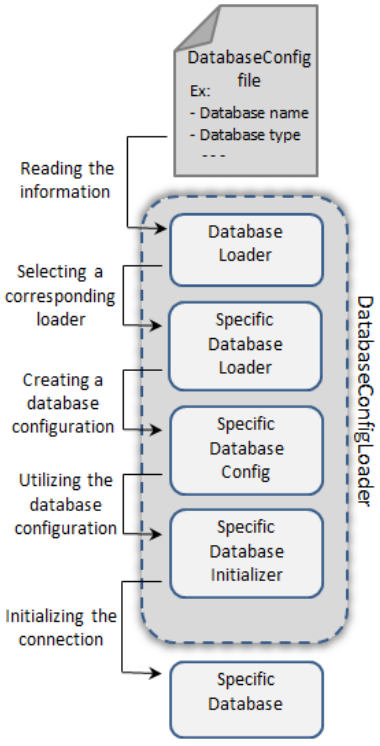


Fig. 8 – Initialization of database connection

f. Management of persistent objects

The components of EPM that support the management of persistent objects at runtime are: ClassMaps and SpecificDatabases. The main component of EPM (PersistenceManager) cooperates with these componentsand with shadow information to manage the persistent objects. The PersistenceManager provides the application developer with a programming interface to store, load and unload the persistent objects to/from database. We propose two methods within the PersistenceManager: to manage the active persistent objects in the main memory and to manage the registered persistent objects during the mainloop, as explained in the following.

Active Persistent Objects

To manipulate an object, it must be copied from the database into the main memory. The object that resides in the main memory is called *active object*. To manage the active state of the persistent objects, the EPM module checks if the persistent object is newly loaded to the main memory or already resides there. The PersistenceManager makes a list of IDs for persistent objects which currently reside in the memory. Here, the ID is the EPM persistenceID introduced at the beginning of this section. For instance, if EPM requests an active persistent object for a second time (Fig. 9 step 1), then the Persistence-Manager checks whether this object is already in main memory by checking the list of IDs (step 2), and after finding it, the PersistenceManager returns to the active object in the main memory (step 3). Therefore, the PersistenceManager can store the object copy which contains all changes without any data loss (step 4).

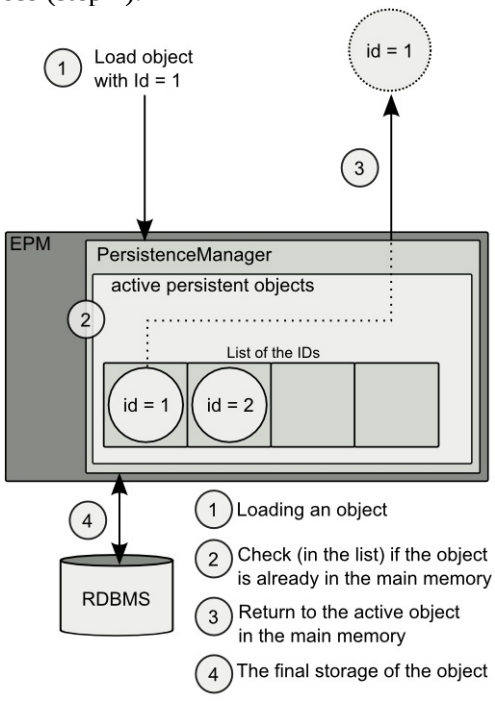


Fig. 9 – Loading an already active object

Automatic storage of objects

EPM allows for saving persistent entities during the mainloop. The API of the PersistenceManager offers a possibility for the application developer to register any persistent entity which needs to beautomatically and continuously stored.The complete storing of all persistent entities cannot be implemented efficiently at each iteration of the mainloop, rather the application developer should specify which attributes should be stored at what time: e.g., theattributes that do not change often can be stored at longer intervals than other attributes. Fig. 10 illustrates the process of automatic persistence for an object with two attributes. The first attribute (black) should be saved every 20 ticks, and the second attribute (gray) should be saved every 4 ticks. For this reason, the object is registered twice in the PersistenceManager for automatic storage (Fig. 10 step 1): once for storing the first attribute and once for storing the second attribute. The PersistenceManager holds a list that contains: the persistenceID of the registered objects, the attributes, and the tick-numbers. When the automatic persistence method is initiated (step 2), the PersistenceManager checks which object must be saved during the mainloop-tick. In the 4th tick, the second attribute is not stored, rather it is updated and remains in the main memory for a limited time. However, in the 20th tick, both attributes will be stored in the database because 20 is a multiple of 4. EPM combines all suspended updates of the persistent object and then stores them using only one access to the database (step 3).

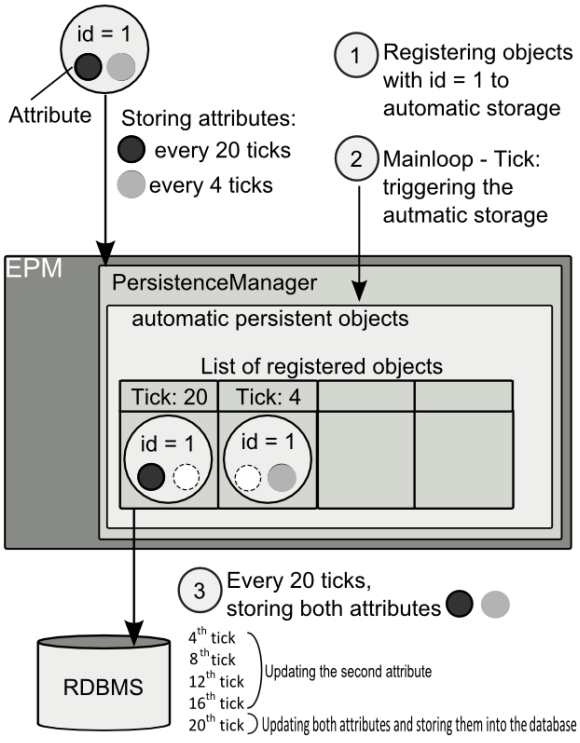


Fig. 10 – Automatic saving of an object

g. Persistence of entities

EPM aims at ensuring persistence for two kinds of objects: (1) states of game world, and (2) individual entities. These objects are mapped onto database tables for persistency. For this *Object-Relational mapping* [12], various kinds of information are required by EPM: the mapping information which specifies where and how the entities are stored, and the shadow information that is needed by EPM at runtime for managing persistent objects. The mapping strategy of EPM for dealing with the tables of the database is divided into three areas of mapping: (a) mapping of attributes, (b) mapping of hierarchies, and (c) mapping of relationships.

6. CONCLUSION AND RELATED WORK

Persistent data storage plays an important role in many distributed Real-time Online Interactive Applications (ROIA) such as modern Massively Multiplayer Online Games (MMOG). For the game developer, programming the connection between MMOG applications and RDBMS is not only time-consuming and error-prone, it is also poorly reusable. Therefore, a flexible and reusable solution is desirable.

In this paper, we analyze the problem of persistency for ROIA applications and present a preliminary design of the Entity Persistence Module (EPM) as a middle software layer to store the persistent data of MMOG applications. The game developer is provided by EPM with a comfortable API that relieves him from writing any additional code for both database access and object-relational mapping. The developer creates a configuration file to define which objects and attributes of the objects are persistent and in which database they should be stored. Depending on this information, the Mapping Classes and the required database schemas are automatically generated by EPM. The Mapping Classes then prepare the persistent data of the application and make it compatible with structures and data-types of RBBMS, as well as generate the required SQL commands to retrieve and store data from/to database.

The presented methods block the mainloop of ROIA as short as possible by generating the SQL commands to update the database, and executing them asynchronously, running in a separate thread. EPM provides a method for partial storage of objects because not always the whole object needs to be saved if only few attributes have changed. With this method, the time to build and run the SQL commands is shortened. After registering an object in the automatic storage method, the application

developer has the opportunity to store an object continuously in a database.

Although there are several sophisticated persistent data systems for Java such as Hibernate [14], or Java Data Objects [15], only few systems have been developed for C++. For example, *LiteSQL* [16] focuses on object-relational mapping by providing a layer that integrates C++ objects into a relational database; our persistence layer is specified to persist the state of real-time applications, and in addition to persist the C++ objects by our approach of object-relational mapping. *DataXtend CE* [17] has been used for applications with demanding real-time and object persistence requirements, particularly, in the fields of financial applications, flight booking, and courier delivery services. But it could not be applied in the field of MMOG applications, because the complexity of MMOG-architecture requires an efficient approach to manage the persistence of both objects and game state that are distributed across multiple game servers.

In comparison to existing approaches in the field of object persistence middleware for MMOG applications like Versant [4], EPM provides more generic middleware which allows to store the persistent data to major types of relational databases, while [4] depends upon a native persistence for objects. Hence, the core database engine of [4] requires a specific database technology while our approach overcomes this drawback.

As future work, we plan to integrate EPM with the Real-Time Framework (RTF) [18] that was developed at the University of Münster [19] within the *edutain@grid* project. After integrating the features of EPM (object- and game state-persistence) with the features of RTF (high-level game design), we will obtain a comprehensive middleware for developing and running online games.

ACKNOWLEDGMENT

Mohammed Nsaif is supported by the cooperative program (BaghDAAD) for German-Iraqi academic exchange.

7. REFERENCES

[1] World of Warcraft – Homepage, Blizzard Entertainment, [Online]. http://www.wow-europe.com/de/index.xml

[2] The database technology of Guild Wars, [Online], http://www.dbms2.com/2007/06/09/the-database-technology-of-guild-wars

[3] Mitch Wagner, Inside Second Life's Data Centers. In: Information-Week. [Online]. http://www.informationweek.com/news/showArticle.jhtml?articleID=197800179

[4] Robert Green, Advanced Mata Management for MMOG – The Versant Object Database in MMOG Applications, Versant, White Paper Version 2008.

[5] Quake 3 Arena Homepage. [Online]. http://www.idsoftware.com

[6] Database Classifications and the Marketplace. [Online]. http://seqcc.icarnegie.com/content/SSD/SSD7/1.5.2/normal/pg-trends/pg-nonrdb/pg-dbclassifications/pg-dbclassifications.html

[7] Lisbeth Bergholt and Jacob Steen Due,.: The Centre of Object Technology (COT), 1998.

[8] MySQL++ Homepage. [Online]. http://www.tangentsoft.net/myaql++/

[9] Lipqxx Homepage. [Online]. http://pqxx.org/development/libpqxx/

[10] SOCI Homepage. [Online]. http://www.soci.sourceforge.net/

[11] Sun Microsystems – Core J2EE Patterns – Data Access Object, [Online]. http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

[12] S.W. Ambler. (1998, Amby-Soft Inc. Version: May) Mapping Objects to Relational Databases: O/R Mapping In Detail. [Online]. http://www.agiledata.org/essays/mappingObjects.htm

[13] S.W. Ambler, The Design of a Robust Persistence Framework for Relational Databases / Amby-Soft Inc. [Online]. http://www.ambysoft.com/downloads/persistenceLayer.pdf

[14] Hibernate Homepage. [Online]. http://www.hibernate.org

[15] Java Data Objects Homepage. [Online]. http://java.sun.com/jdo/

[16] LiteSQL Homepage. [Online]. http://sourceforge.net/projects/litesql/

[17] DataXtend CE – Progress Software. [Online]. http://www.progress.com

[18] Frank Glinka, Alexander Ploss, Sergei Gorlatch, and Jens Müller-Iden, High-Level Development of Multiserver Online Games, International Journal of Computer Games Technology, no. Article ID 327387, pp. 16 pages doi: 10.1155/2008/327387, vol 2008.

[19] The Real-Time Framework (RTF). [Online]. http://www.real-time-framework.com/

Max Knemeyer *received MSc degree in 2009 in Computer Science from the University of Muenster (Germany). He worked in the group of parallel and distributed systems at the University of Muenster. His research area focuses on relational databases, object oriented application frameworks, Real-time Online Interactive Applications (ROIA), and Massively Multiplayer Online Games (MMOG).*

Mohammed Nsaif *received MSc degree in 2005 in Computer Science from Iraqi Commission for Computers and Informatics (ICCI) in Iraq. He is a PhD student with the Department of Computer Science (in the group of Prof. Sergei Gorlatch), University of Muenster in Germany. His research interests are relational database management systems, distributed systems, Real-time Online Interactive Applications (ROIA), and Massively Multiplayer Online Games (MMOG).*

Frank Glinka *received his computer science degree from the University of Muenster in 2006 and is now a research associate at the department of computer science in the group of Prof. Sergei Gorlatch. He has worked as a work package leader for the European research project edutain@grid covering the topic of real-time application services and is currently completing his PhD thesis titled "Developing Grid Middleware for a High-Level Programming of Real-Time Online Interactive Applications".*

Alexander Ploss *received his degree in mathematics from the University of Muenster in 2007 and was a research associate in the department of computer science in the group for parallel and distributed systems at Muenster until July 2011. During this time, he worked in the international research project edutain@grid and received his doctoral degree in 2011 for his dissertation on "Efficient Dynamic Communication for Real-Time Online Interactive Applications in Heterogeneous Environments". He has published about 20 papers in peer-reviewed conferences and journals as well as book chapters on the scalability of interactive applications, e.g., massively multiplayer online games, with the focus on communication aspects.*

Sergei Gorlatch *has been Full Professor of Computer Science at the University of Muenster (Germany) since 2003. Earlier he was Associate Professor at the Technical University of Berlin, Assistant Professor at the University of Passau, and Humboldt Research Fellow at the Technical University of Munich, all in Germany. Prof. Gorlatch published more than 150 peer reviewed papers and books. He has been principal investigator in various international research and development projects in the field of parallel, distributed, Grid and Cloud computing. Sergei Gorlatch holds MSc degree from the State University of Kiev, PhD degree from the Institute of Cybernetics of Ukraine, and the Habilitation degree from the University of Passau (Germany).*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

КОНЦЕПЦИЯ ГИБРИДНОЙ АДАПТИВНОЙ ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ

Игорь Котенко, Филипп Нестерук, Андрей Шоров

Лаборатория проблем компьютерной безопасности СПИИРАН
14 линия, 39, Санкт-Петербург, 199178, Россия
ivkote@iias.spb.su, 08p@mail.ru, ashorov@comsec.spb.ru
http://comsec.spb.ru

Резюме: *в работе предлагается концепция гибридной адаптивной защиты информационно-телекоммуникационных систем на основе биометафоры нервных и нейронных сетей. Верхний уровень системы защиты, основанный на подходе «нервная система сети», базируется на распределенном механизме сбора и обработки информации, который координирует действия основных устройств компьютерной сети, идентифицирует атаки и принимает контрмеры. Реализацию информационных процессов на нижнем уровне предлагается выполнять с привлечением «информационно-полевого» программирования, которое позволяет описывать распределенные информационные поля в виде пакетных нейросетевых программ.*

Ключевые слова: *компьютерная безопасность, гибридная защита, нервная система, нейронные сети.*

CONCEPTION OF A HYBRID ADAPTIVE PROTECTION OF INFORMATION SYSTEMS

Igor V. Kotenko, Filipp G. Nesteruk, Andrey V. Shorov

Laboratory of computer security problems SPIIRAS
39, 14th Liniya, St. Petersburg, 199178, Russia
ivkote@iias.spb.su, 08p@mail.ru, ashorov@comsec.spb.ru
http://comsec.spb.ru/en

Abstract: *The paper suggests the conception of a hybrid adaptive protection of information and telecommunication systems which is based on a biometaphor of nervous and neural networks. A top level of a protection system, based on an approach of "nervous system network" is a distributed mechanism for collecting and processing information. We suggest to implement the information processes on the low level with the assistance of an "information field" programming. It allows specifying the distributed information fields in the form of neural network software packages.*

Keywords: *Data mining, malware, detection.*

1. ВВЕДЕНИЕ

Актуальность проблемы защиты информационных систем, продиктована сложностью программного и аппаратного обеспечения, обуславливающей наличие уязвимостей, прогрессирующей динамикой их развития, распределенной и разнородной структурой и многими другими факторами. Очевидна аналогия между эволюцией и естественным отбором в природе и информационно-телекоммуникационных системах. Живые организмы существуют и эволюционируют, в том числе благодаря совершенной защите от различных угроз, выработанной веками, используя информацию, циркулирующую в их распределенной структуре на основе реализации различных механизмов защиты.

Поэтому представляется, что необходимо наделять системы защиты информации (СЗИ) информационно-телекоммуникационных сетей эволюционными свойствами, присущими биосистемам, такими как возможность развития

(самосовершенствования), адаптивность (приспособление к текущим условиям обстановки), репродукция и наследование. Этот тезис подтверждается текущими тенденциями в индустрии программных систем – известные производители программного обеспечения заявляют, например, о необходимости применения технологий активной адаптивной защиты, основанной на оценке поведения программных компонентов с точки зрения их потенциальной опасности.

В свете современных представлений постановка задачи разработки моделей, методик и алгоритмов создания адаптивных СЗИ носит комплексный характер и может основываться на биосистемной аналогии. Эволюция средств обработки информации осуществляется в направлении создания систем с элементами самоорганизации, в которых присутствуют процессы зарождения (запуска) необходимых функций, сервисов и процессов, их приспособления и развития. На названных процессах основаны биологические системы, для которых характерны высокая защищенность, накопление опыта эволюции и селективный отбор.

Заимствование архитектурных принципов биосистем привело к разработке теорий нейронных сетей (НС), нейро-нечетких систем (ННС), иммунокомпьютинга и эволюционных методик, лежащих в основе искусственных интеллектуальных систем, базирующихся на распределенной нейросетевой обработке информации и использовании принципов иммунной защиты биосистем.

В настоящей работе предлагается концепция гибридной адаптивной защиты информационных систем на основе гибридных механизмов, сочетающих биометафоры нервных и нейронных сетей. Статья организована следующим образом. В разделе 2 дан краткий анализ исследований, основанных на биосистемной аналогии и гибридных подходах. В разделе 3 представлено видение подхода «нервная система сети», как верхнего уровня системы защиты. В разделе 4 описывается нижний, нейро-нечеткий уровень системы, и приводится пример его схемотехнической реализации. В разделе 5 верхний и нижний уровни рассматриваются в совокупности с целью описания концепции гибридного адаптивного подхода к защите информационных систем на основе нервных и нейронных сетей. В заключении, разделе 6, обобщены выводы по предложенной концепции.

2. АНАЛИЗ ИССЛЕДОВАНИЙ

Биосистемы обладают многоуровневой иерархической системой жизнеобеспечения, реализованной с использованием комплекса механизмов информационной избыточности, защиты и иммунитета. Механизмы защиты информации по возможностям далеки от биологических прототипов, поэтому разработка технологии создания адаптивных систем с встроенными функциями жизнеобеспечения и защиты, основанных на биосистемной аналогии, представляется актуальной [1–5]. Особенно эта задача актуальна для информационных систем критических инфраструктур, которые должны выполнять свое назначение в условиях воздействий угроз всевозможных категорий.

Одним из основных направлений развития информационных систем можно считать создание гибридных адаптивных СЗИ, реализующих механизмы жизнеобеспечения и защиты биологических систем, базирующихся на технической реализации с привлечением современных технологий в виде сверхбольших интегральных схем (СБИС).

Особую роль в эволюции биосистем играет нервная система как адаптивный инструмент взаимодействия со средой. Нервная система необходима для формирования рефлексов в ответ на воздействия. Рефлексия – продукт верхних уровней информационных систем, а информация о механизмах реализации рефлексов хранится на нижних уровнях (в генетической памяти) и наследуется. Поведенческие реакции в биосистеме – результат функционирования нервной системы, свидетельствующий о развитии связи между воздействиями и реакцией организма. Отмечают разделение информации между носителями различной природы: ДНК и нервными клетками – нейронами. Поведенческая информация формируется на основе механизмов, передаваемых через ДНК, и фиксируется в информационном поле нервной системы. Биосистемам свойственно накопление жизненного опыта и передача его потомкам через обучение [6, 7]. Целенаправленность поведения биосистемы развивает форму памяти в виде адаптивного информационного поля нейронной сети нервной системы.

Анализ источников научно-технической информации показал, что исследованию средств, основанных на распределенной нейросетевой обработке информации и принципах иммунной защиты биосистем, гибридным подходам, уделяется большое внимание.

Компания HP пропагандирует технологию ProCurve, в основе которой лежит попытка

интеллектуализации таких сетевых устройств как коммутаторы, маршрутизаторы, точки доступа к беспроводной сети. В частности, делается попытка наделить эти устройства функциями, отвечающими за безопасность сети, например, такими как проверка и фильтрация пакетов, защита от вирусов, шифрование данных.

Компания Cisco применяет концепцию самозащищающейся сети (Cisco's Self-Defending Network). Для защиты передаваемых по сети данных используются защищенные протоколы и технология VPN. Для защиты от внешних угроз задействуется интегрированная система, состоящая из различных компонентов защиты, таких как межсетевые экраны, системы предотвращения вторжений, системы защиты от DDoS-атак и др. Для защиты клиента используются специальные программные агенты, которые служат для конфигурирования клиента в соответствии с заданной политикой безопасности, используемой в компьютерной сети. Также обеспечивается базовая аутентификация пользователей и проверка на соответствие клиента заданной в сети политике безопасности. На основе полученных данных пользователь может получить доступ в сеть или ему может быть отказано в доступе. Имеется возможность создания зон карантина, куда перенаправляются пользователи, не удовлетворяющие условиям, которые требуются для получения доступа в сеть.

Перспективной считается концепция самозащищающейся сети, которая может распознавать все объекты по принципу "свой-чужой", а также защита на основе проверки сетевых объектов на соответствие применяемым политикам безопасности. В случае несоответствия требуемому уровню защищенности, проверяемый объект (компьютер, программа, файл) может быть отправлен на карантин, где, если это возможно, путем установки патчей, обновления антивируса и других операций, уровень его защищенности будет повышен, или же объекту будет предоставлен ограниченный доступ либо отказано в доступе.

Гибридный принцип, или принцип гибридности, зачастую заключается в сочетании, казалось бы, несочетаемого, вследствие чего традиционные системы, приобретают совершенно новые свойства и становятся уникальными в своём роде и области. Так, например, лидер японского автопрома Toyota, выпускает с заката прошлого века гибридные модели Prius [8], позволяющие существенно снизить потребление топлива при соизмеримых мощности, скорости, массе и характеристиках аналогичных моделей машин. Неоспоримы успехи исследователей в области биологии по отбору, селекции, скрещиванию и получению новых полезных гибридов [9]. Продукт «Лаборатории Касперского» Kaspersky Internet Security 2012, в котором реализован принцип гибридности, основан на сочетании классических антивирусных и новейших поведенческих и облачных технологий защиты, что позволяет не только минимизировать время реакции на угрозы, но и снизить нагрузку на компьютеры [10].

В настоящее время сложился определенный задел в области нейросетевой обработки и иммунокомпьютинга, имеется ряд результатов, реализующих нейросетевые средства интеллектуального анализа данных, применимых для защиты информации. Отметим здесь работы [11–20], важные для исследуемой области.

Рассмотрим ниже сначала отдельно два различных биоинспирированных подхода, предлагаемых авторами работы, а затем сформулируем предлагаемую концепцию гибридной адаптивной защиты, которая базируется на использовании этих подходов на различных уровнях представления системы защиты.

3. МЕХАНИЗМЫ РЕАЛИЗАЦИИ ВЕРХНЕГО УРОВНЯ СЗИ

Нервная система человека была взята как основа подхода к защите компьютерных сетей, называемого "нервная система сети", предложенного, наприме р, в работе Ю.Чена и Х.Чена [21].

На основе биоаналогии, подход "нервная система сети" использует распределенный механизм сбора и обработки информации для обнаружения атак и противодействия им. Подобно биологической нервной системе, множество компонентов защиты связаны между собой, что позволяет оперативно обмениваться информацией, координировать действия узлов входящих в "нервную систему", детектировать атаки и принимать меры для их нейтрализации.

Структура данной системы повторяет структуру нервной системы человека (рис. 1). Механизм работы нервной системы сети – распределенный, т.е. предполагается, что нет единого центра, который координирует действия всей сети.

Предполагается, что сетевые домены Интернет-провайдеров (ISP) или автономные системы (AS) соединены между собой как физически связанные нейроны. В каждом домене есть специальный сервер (или кластер серверов).

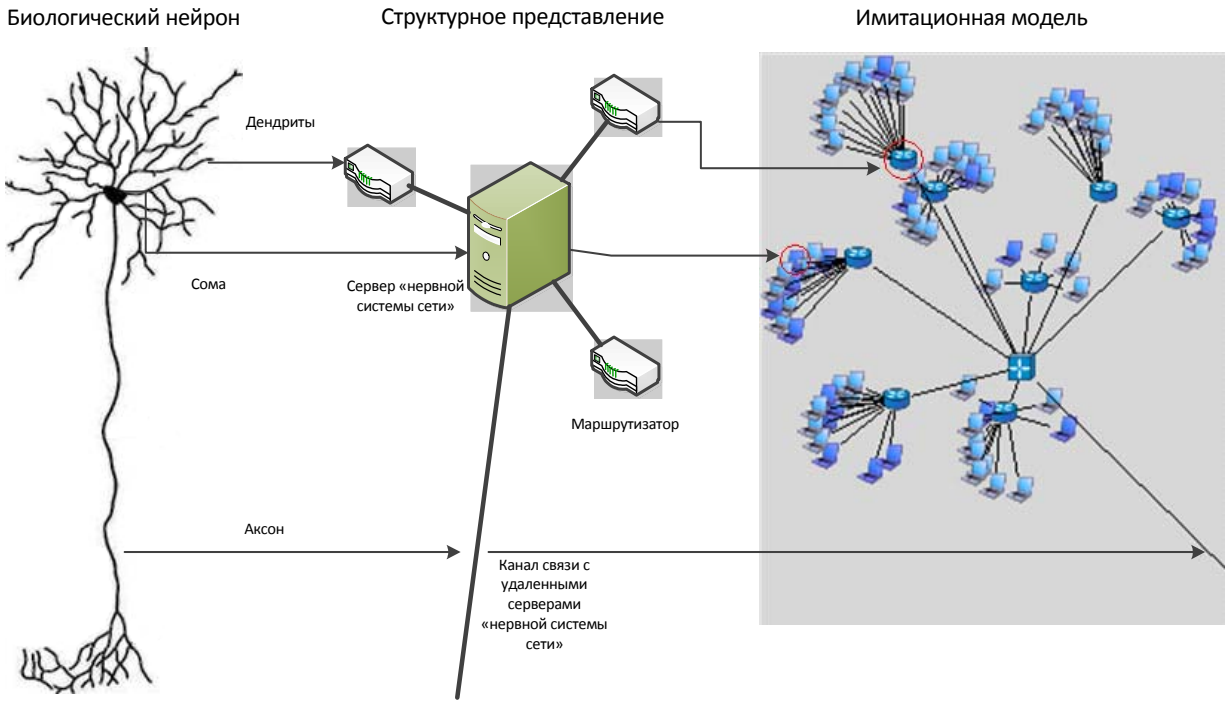


Рис. 1 – Представление биологического нейрона в модели компьютерной сети

Этот сервер исполняет роль сомы в нейроне. Сома является центральной частью нейрона, она реализует большую часть процессов обработки и анализа информации.

Другие сетевые устройства (маршрутизаторы) функционируют как дендриты нейрона, которые передают большую часть информации нейрону. Виртуальная частная сеть (VPN), к которой подключены все серверы, соответствует аксону, передающему сигналы от сомы к другим нейронам (доменам), а также получает информацию от этих нейронов (доменов).

Для обеспечения безопасности системы в [21] предлагается протокол IFSec (InFrastructure Security protocol). Этот протокол работает на сетевом уровне (уровень 3) и определяет формат и механизм шифрования, которые поддерживают безопасный обмен информацией между доменами (нейронами), а также между маршрутизаторами (дендритами) и сервером (сома) в домене. IFSec строится как надстройка IP и работает прозрачно, чтобы транспортировать протоколы более высокого уровня.

Протокол IFSec предоставляет три уровня коммуникации.

Самый низкий уровень дает возможность маршрутизаторам в одном домене обмениваться информацией для контроля состояния сети.

Второй уровень — коммуникация между маршрутизаторами и сервером, расположенными в одном домене.

На самом высоком уровне сервер обменивается информацией с другими серверами, расположенными в других доменах.

Таким образом, протокол IFSec работает в трех различных слоях. Слой 1 служит для коммуникации между одиночными узлами. Слой 2 реализует взаимодействие между узлами и их сервером. Слой 3 объединяет серверы в разных доменах.

Архитектура системы, основанной на данном подходе, представляется следующим образом. Домены сети, которые подключены к нервной системе сети, формируют оверлейную сеть и взаимодействуют между собой с помощью протокола IFSec. Маршрутизаторы, расположенные в разных точках сети, взаимодействуют не только друг с другом, но и со специализированным сервером безопасности в своей подсети [22 – 26].

Функциональные возможности данной архитектуры могут быть представлены на двух уровнях: локальная обработка поступившей информации на отдельных устройствах и обработка информации в масштабе распределенной кооперации провайдеров.

Конкретный процесс по обеспечению защиты осуществляется локально, т.е. в каждом отдельном узле. Крупномасштабная кооперация выполняется для реализации защищенного обмена информацией как внутри домена (от маршрутизатора к маршрутизатору, от маршрутизатора к серверу), так и между доменами (от сервера к серверу). В этом случае

информация автоматически распределяется по различным узлам сети. Своевременное получение информации позволяет более эффективно реагировать на различные внешние угрозы.

Каждый узел состоит из функциональных блоков со стандартным интерфейсом передачи данных, что обеспечивает большую гибкость при динамическом обновлении и обслуживании узлов.

Для начала представим общую архитектуру «нервной системы сети» (рис. 2). В подсети 1 компьютерной сети имеется сервер «нервной системы сети». Он связан с серверами «нервной системы» в других подсетях.

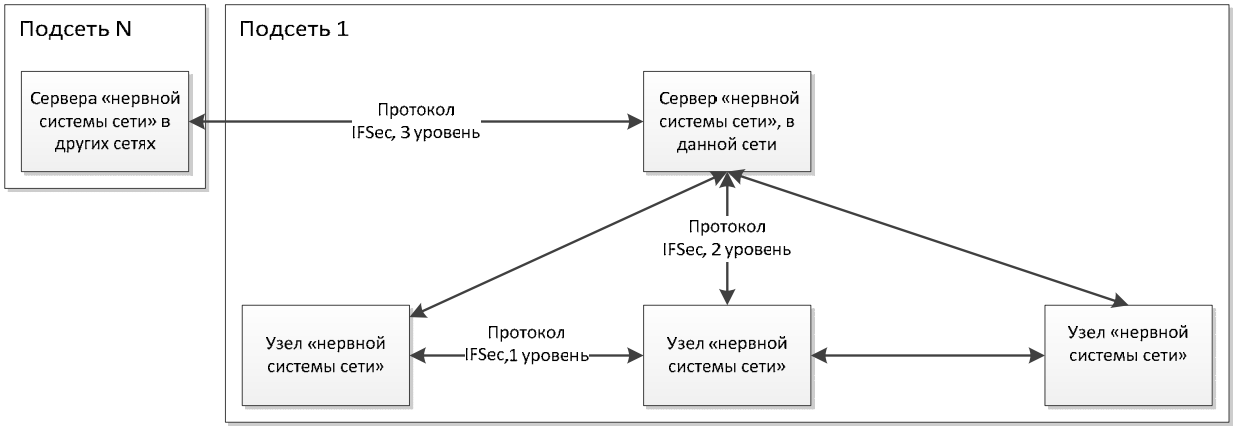


Рис. 2 – Структурное представление «нервной системы сети»

К каждому серверу подключены узлы «нервной системы сети», находящиеся в одной подсети с главным сервером. Кроме того, каждый из узлов имеет связи с другими узлами «нервной системы» в данной подсети. На основе предложенной архитектуры отобразим компонентную структуру «нервной системы сети». В частности раскроем компоненты сервера и узла «нервной системы сети».

Сервер «нервной системы сети» имеет модули обмена данными с подчиненными ему узлами, а также с серверами, находящимися в других подсетях. Модули обмена данными соединены с компонентом, отвечающим за анализ данных и принятие решений.

С помощью него, они получают команды и данные для отправки на узлы и другие сервера «нервной системы» и доставляют ему информацию о событиях, происходящих в сети.

К модулю анализа и принятия решений подключена база данных, которая служит хранилищем данных, полученных из внешних источников, и поставляет ранее сохраненную информацию.

В модуле анализа и принятия решений данные, полученные от модулей обмена информацией с узлами и серверами «нервной системы сети», попадают в модуль приоритезации, где в соответствии с установленными политиками классифицируются события, и определяется, насколько важна та или иная информация, на основе чего принимается решение об очередности выполнения действий в следующих блоках.

Затем данные попадают в модуль корреляции, который, в соответствии с приоритетом, выбирает события и запрашивает похожие события из базы данных (БД) с помощью компонента «обмен данными с БД».

После чего происходит сопоставление набора событий, и определяется уровень угрозы. Для обнаружения источника атаки используется алгоритм на основе подхода «множество изменяемых деревьев» [4].

После этого в блоке «решение о блокировке» на основе политик и порогов определяется реакция на текущую ситуацию в сети.

В данном модуле проверяется, превысил ли подозрительный IP-адрес пороговое значение. Если превысил, то принимается решение о блокировке адреса атакующего, которое отправляется всем подчиненным узлам, а также удаленным серверам «нервной системы сети».

Узел «нервной системы сети» является модулем для первичной обработки данных, поступающих с сенсоров, и управления режимами их работы. В качестве сенсоров могут выступать как простые мониторы трафика, так и более сложные механизмы защиты.

На первом этапе обработки узел с помощью блока перенаправления потоков распределяет потоки трафика, исходя из IP-адреса отправителя. Далее, используя блок классификации пакетов, он определяет типы пакетов, отправляемых источником. После этого производится анализ трафика, полученного после обработки. К модулю анализа и

противодействия подключена база данных, из которой он получает информацию, на основе чего производится анализ трафика.

Если узел обнаружил, что трафик – вредоносный, он передает информацию об этом вместе с данными о вредоносном трафике модулю сдерживания атак. Легитимный трафик возвращается в сеть. Модуль сдерживания атак, с помощью компонентов обмена данными, пересылает эту информацию серверу и узлам, а также получает информацию от них и обновляет базу данных правил и сигнатур.

В модуле анализа и противодействия пакеты из модуля классификация попадают в компонент «анализ на основе правил», где на базе правил фильтрации принимается решение о блокировке трафика.

Далее трафик проходит модули поиска аномалий и сигнатурного анализа. Если какой-либо модуль принял решение о блокировке трафика, пакет, не проходя через последующие фильтры, попадает на модуль блокировки, который в случае положительного решения удаляет пакет и передает информацию о нем в модуль сдерживания атак. В случае если пакет легитимный, он возвращается в сеть. Если узел обнаруживает вредоносные потоки трафика, он отправляет сообщение об этом серверу «нервной сети», с которым он связан.

Серверы «нервной системы сети» постоянно анализируют информацию, поступающую от подключенных к ним узлов и других серверов, вследствие чего принимаются решения об ограничении работы тех или иных пользователей вычислительной сети.

4. МЕХАНИЗМЫ РЕАЛИЗАЦИИ НИЖНЕГО УРОВНЯ СЗИ

В качестве базы для построения адаптивных СЗИ может быть использован технический аналог биосистемы в виде взаимосвязанных интерфейсом нейросетевых командных пулов, управляемых потоком данных [27].

В соответствии с принципами монолитности исполнения и многофункциональности, обработку данных целесообразно организовывать в командных пулах путем выполнения операций чтения, модификации и записи.

Формальная модель процессов, протекающих в информационных полях нейросетевых средств защиты информации в основных режимах работы, необходима для адекватного задания методов проектирования и верификации адаптивных СЗИ, специфицируемых с помощью пакетных нейросетевых программ.

Программирование информационных полей нейронных сетей (и нейро-нечетких систем) в СЗИ можно свести к описанию структуры информационных полей с помощью пакетных нейросетевых программ [27], что позволяет детализировать и исследовать процессы, происходящие в нейронных сетях различных уровней систем адаптивной защиты путем моделирования взаимодействия оперативных данных с распределенными избыточными информационными полями нейронных сетей.

Адаптивность СЗИ предлагается обеспечить использованием функционально устойчивой элементной базы – обычных и логарифмических формальных нейронов, способных к обучению. Адаптивные средства СЗИ согласно принципу биоаналогии следует представлять в виде описания информационных полей нейронных сетей иммунного и рецепторного уровней защиты. Нейронная сеть представляется в виде совокупности взаимосвязанных командных пакетов, которая размещается в командных пулах. При описании нейронных сетей пакетными нейросетевыми программами возможна различная степень детализации: командный пакет может соответствовать одной из функций нейросетевого логического базиса, функции формального нейрона, слоя из формальных нейронов или нейронной сети в целом.

Важным принципом биосистемной аналогии является представление жизненно важных функций и информации в форме топологии, например, генома биологического вида [27]. Известен подход представления топологии информационной системы в виде совокупности командных пакетов, каждый из которых соответствует отдельному фрагменту топологии и определяет реализуемую фрагментом функцию, а также местоположение источников исходных данных и приемников результатов [28]. Пакеты данных предназначены для передачи результатов обработки информации от одних командных пакетов (источников) другим командным пакетам (приемникам). Данный подход соответствует потоковым вычислениям, а подобные информационной системы называют машинами, управляемыми потоком данных.

Программирование в биосистемах носит избыточный распределенный характер, что обеспечивает высокую функциональную устойчивость информационных процессов. Отдельные искажения информации, с одной стороны, компенсируются избыточностью информационных полей, а, с другой, – создают предпосылки для реализации механизма мутаций и эволюционных процессов развития и отбора.

Для исследования информационных процессов в адаптивных СЗИ можно использовать пакетные нейросетевые программы, которые позволяют описывать топологию избыточных распределенных информационных полей НС [29].

Командные пулы организуются в виде многофункциональной регулярной вычислительной структуры, в которой размещены пакетные нейросетевые программы. В качестве средства формализации выбран язык графического описания объектов, а в качестве механизма управления вычислениями – машины, управляемые потоком данных, которые обеспечивают безопасность хранимой информации: операция записи данных производится не по конкретному адресу памяти, а по содержанию; отсутствует операция выборки данных из памяти и, следовательно, непосредственный доступ к информации. Готовые к обработке данные, представленные в виде пакетов, извлекаются из памяти автоматически (без управления извне).

Объединение функций хранения и обработки информации в многофункциональных пулах упрощает их структуру за счет исключения части коммуникационных цепей, предназначенной для передачи готовых к обработке командных пакетов от локальных пулов команд к процессорным узлам, и снижает загрузку интерфейса. Минимизация потоков данных между командными пулами позволяет использовать простейшие виды интерфейсов для передачи пакетов данных. По мере повышения функциональной мощности командных пакетов наблюдается снижение объема передачи пакетов и функциональная специализация командных пулов. И наоборот, снижение функциональной мощности командных пакетов приводит к универсальности командных пулов, интенсификации трафика передачи сообщений, что предъявляет повышенные требования к скоростным возможностям интерфейса.

Наличие современной технологической базы делает целесообразным использование командных пакетов, соответствующих уровню детализации командный пакет – слой формальных нейронов. Для реализации командных пулов на базе СБИС с программируемой структурой следует ограничиться уровнем командного пакета – формальных нейронов, а минимизацию информационного обмена обеспечивать путем размещения пакетных нейросетевых программ в пределах базового блока (ряда базовых блоков) для замыкания информационных потоков между слоями или формальными нейронами НС в рамках отдельных СБИС.

Средства адаптивной защиты могут быть распределенными по базовым блокам, либо локализованными в отдельном базовом блоке. Предложен схемотехнический вариант реализации адаптивной нейросетевой вычислительной среды, отличающийся организацией интерфейса между памятью командных пакетов и операционными блоками [27]. Отмечена взаимосвязь структуры базовых блоков с уровнем детализации описания процессов в НС, формой представления и порядком поступления пакетов данных.

Реализацию информационных процессов на нижнем уровне предлагается выполнять с привлечением так называемого “информационно-полевого” программирования, которое позволяет описывать избыточные распределенные информационные поля в виде пакетных нейросетевых программ [27, 30]. Адаптивные процессы в информационных полях позволяют СЗИ развиваться и накапливать опыт при расширении множества угроз, а наследование опыта сводится к передаче информационных полей в аналогичные по назначению системы.

В качестве базы для создания адаптивной системы защиты информации (технический аналог живого организма) можно использовать нейросетевую среду – взаимосвязанные интерфейсом командные пулы, используемые для размещения пакетных нейросетевых программ и выполнения распределенной обработки за счет взаимодействия оперативных данных с адаптивным избыточным информационным полем НС.

Для обеспечения целостности информации в нейросетевых СЗИ можно использовать аппаратные способы защиты информации, например, на основе организации командных пулов в виде накопителей, не имеющих внешних шин записи/чтения, в которых доступны только входная и выходная очереди, что затрудняет осуществление несанкционированных действий, нарушение целостности и конфиденциальности информации, в сочетании с комбинированием различных механизмов защиты, например, с комбинированием обнаружения сканирования в компьютерных сетях [31].

5. ГИБРИДНЫЙ ПОДХОД К ЗАЩИТЕ ИНФОРМАЦИИ

Предполагается, что концептуальные и архитектурные решения по построению адаптивных СЗИ, должны быть основаны на

принципах биоаналогии [11]. Предлагаемая концепция гибридной адаптивной защиты информационных систем формируется путем объединения представленных выше подходов.

В качестве базы для построения адаптивных СЗИ предлагается использовать технический аналог структуры биосистемы в виде взаимосвязанных нейросетевых командных пулов (центров), управляемых на основе поступающих данных о состоянии системы.

Архитектурной особенностью биосистем является внутрисистемный характер механизмов защиты. Поэтому в процессе проектирования СЗИ предполагается, что функции защиты информации должны быть внутренними функциями проектируемой системы.

Иерархия адаптивной СЗИ отражает разделение функций защиты на иммунные, проверяющие форму представления информации, и рецепторные, реализующие взаимодействие со средой и накопление опыта.

Выделим, в качестве базовых, пять следующих принципов построения гибридных адаптивных СЗИ, основанных на биологической метафоре.

1. Интеллектуальная обработка информации, интеллектуальный анализ информации:

- обеспечение иерархии элементов обработки информации необходимыми ресурсами;
- на нижних уровнях иерархии осуществляется хранение и анализ генетической информации, реализация механизмов мутации и распределенного преобразования информации, разделение сообщений в соответствии с анализом по критерию “свой/чужой”, накопление опыта по идентификации патогена в иммунологической памяти;
- на верхних уровнях иерархии реализуется связь системы со средой через “органы чувств” (сенсоры) и накопление опыта в распределенных информационных полях нервной системы;
- изменение генетической информации связывается с изменением не формы представления, а содержания информации;
- защита информации обеспечивается, в том числе, за счет реализации свойства адаптивности – приобретения жизненного опыта, позволяющего успешно оперировать ситуациями, в частности, распознавать своих и чужих, выбирать поведение в сложной изменяющейся обстановке.

2. Биосистемная аналогия:

- информация в элементах обработки информации хранится в виде структурированных информационных полей: внизу иерархии – поля идентифицирующего угрозы, вверху иерархии – поля опыта, ставящего в соответствие полю известных угроз механизмы защиты информации;
- нижние (иммунные) уровни средств защиты осуществляют проверку соответствия формы передаваемых в системе сообщений по критерию “свой/чужой”;
- идентифицирующая информация – своя для каждой системы и связана с формой, но не содержанием информации; (как, например, паспорта различных государств, содержащих идентифицирующую информацию, такую как фотографический образ, имя, личную подпись, дату и место рождения, идентификационный номер присвоенный органом государственной регистрации, адрес проживания, и т.п., по принятыми, законами конкретных стран формам, но не несущих полную информацию о владельце паспорта)
- верхние (рецепторные) уровни защиты необходимы для связи с внешней средой и накопления опыта;
- перенос и наследование информации – передача иерархии информационных полей, сформированных в процессе жизненного цикла адаптивной информационной системы, в последующие реализации системы.

3. Поддержание свойств, необходимых для реализации функций интеллектуального анализа информации:

- возможность наследования ранее накопленного опыта адаптивной информационной системы в виде иерархии информационных полей;
- возможность решения задач классификации и кластеризации с оперативной адаптацией информационных полей;
- коррекция жизненного опыта информационной системы на основе коррекции и расширения системы нечетких правил, адаптация информационных полей иерархии уровней системы;
- возможность анализа, коррекции и переноса (наследования) информации в другие информационной системы.

4. Нейронные и нейро-нечеткие сети представляют собой нижний уровень СЗИ, предназначенный для обмена информацией с внешней средой и передачи ее на верхний уровень СЗИ, приема информации от верхнего уровня, а также формирования ответных реакций на воздействия.

5. Верхний уровень адаптивных СЗИ представлен “нервной системой сети” и предназначен для управления процессами системы и взаимодействия с элементами и блоками нижнего уровня. Верхний и нижний уровни работают как одно целое, в постоянном информационном взаимодействии и согласовании решений в режиме реального времени.

6. ЗАКЛЮЧЕНИЕ

В статье предложена общая концепция гибридной адаптивной защиты информационных систем, сочетающая биометафоры нервных и нейронных сетей.

На основе метафор нервной и нейронной сети в работе предлагается гибридная адаптивная сетевая инфраструктура, обеспечивающая получение, передачу, хранение и защиту информации, принятие решений, исходя из сложившейся ситуации, в соответствии с аналогией работы нервной и нейронной систем живых существ.

Кооперация распределенных компонентов происходит подобно реакции человеческой нервной системы. Одиночные компоненты работают не только как исполнители, но также и как сенсоры. Помимо общей защиты, которая осуществляется ими самостоятельно, они также предоставляют результаты анализа данных другим компонентам системы.

Планируется в результате исследований разработать технологию создания сетевых компонентов со встроенными функциями защиты, отличающуюся представлением структуры компонента в виде иерархии компонентов, выполненных с различной степенью детализации, описанием информационной структуры с помощью графического языка, функциональным блокам которой соответствуют командные пакеты, информационным потокам – пакеты данных.

Достоинствами такого подхода являются применение подхода управления потоком данных для организации распределенных вычислений, а также средств интеллектуального анализа данных в составе адаптивной системы защиты информации для обеспечения оперативной реакции на изменение множества угроз и условий эксплуатации.

Будущая работа связана с моделированием компонентов представленного концептуального подхода к построению гибридных адаптивных систем защиты.

Работа выполняется при финансовой поддержке Министерства образования и науки РФ (государственный контракт 11.519.11.4008), РФФИ (проект №13-01-00843-а), программы фундаментальных исследований ОНИТ РАН (проект №2.2), проектов Евросоюза SecFutur и MASSIF.

7. СПИСОК ЛИТЕРАТУРЫ

[1] D. Dasgupta, H. Bersini, et al., *Artificial Immune Systems and Their Usage*, D. Dasgupta (Eds.), translated from English under edit. of A.A. Romanyiukha, M.: FIZMATLIT, 2006, 344 p. (in Russian)

[2] R.M. Khaitov, *Physiology of Immune System, Moscow,* VINITI RAN, 2001, 223 p. (in Russian)

[3] N.K. Jerne, *Towards a network theory of the immune system*, *Ann. Immunol. (Inst. Pasteur),* (125) (1974), pp. 435-441.

[4] G. Miller, P. Todd, S. Hedge, Designing neural networks using genetic algorithms, *Proc. 3rd Int. Conf. on Genetic Algorithms,* (1989), pp. 379-384.

[5] I.V. Kotenko, F.G. Nesteruk, A.V. Shorov, Methods of computer networks defense on the base of bio-inspired approaches, *Voprosi zaschiti informacii,* (2) (2012), pp.35-46. (in Russian)

[6] M.E. Lobashev, *Genetics*, Leningrad, LGU Publishers, 1969, 357 p. (in Russian)

[7] I.V. Melik-Gaynazya, *Information Processes and Reality*, Moscow, Nauka, 1998, 137 p. (in Russian)

[8] Electronic resource. Access mode – URL: http://www.toyotacenter.ru/ (in Russian)

[9] Electronic resource. Access mode – URL: http://rudocs.exdat.com/docs/index-227356.html (in Russian)

[10] Electronic resource. Access mode – URL: http://www.kaspersky.ru/news?id=207733674 (in Russian)

[11] L.B. Booker, D.E. Goldberg, I.E. Holland, Classifier systems and genetic algorithms, *Artificial Intelligence,* Elsevier, (40) (1989), pp. 235-282.

[12] G. Deffuant, *Reseaux Connectionistes Auto-construits*, These D'Etat, 1992, 141 p.

[13] M. Dorigo, H. Bersini A comparative analysis of Q-learning and classifier systems, *Proc. SAB'94, MIT Press, (*1994), pp. 248-255.

[14] S. Fahlman, C. Lebiere, The cascade-correlation learning architecture, *Advances in Neural Information Processing System*, Morgan Kaufman, (2) (1990), pp. 524-532.

[15] M. Fombellida, Methodes heuristiques et methodes d'optimalisation non contraintes pour l'apprentissage des perceptrons multicouches, *Proc. 5th Int. Conf. on Neural Networks and their Application: Neuro-Nimes, (*1992), pp. 349-366.

[16] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989, 432 p.

[17] Y. Hirose, K. Yamashita, S. Hijiya, Back-propagation algorithm which varies the number of units, *Neural Networks,* (4) (1991), pp. 61-66.

[18] J.H. Holland, K.J. Holyoak, R.E. Nisbett, P.R. Thagard, *Induction: Processes of Inference, Learning and Discovery*, Cambridge: MIT Press, 1986, 386 p.

[19] T. Salom, H. Bersini, An algorithm for self-structuring neural net classifiers, *Proc. 2nd IEEE Conf. on Neural Network* (ICNN'94, 1994), pp. 1307-1312.

[20] R.S. Sutton, Reinforcement learning architectures for animats, *Proc. 1st SAB Conference* (Eds. J.-A. Meyer and S.W. Wilson). MIT Press, (1990), pp. 288-296.

[21] Y. Chen, H. Chen, NeuroNet: An Adaptive Infrastructure for Network Security, *International Journal of Information, Intelligence and Knowledge,* (1) 2 (2009), pp.143-168.

[22] I.V. Kotenko, A.M. Konovalov, A.V. Shorov, Modeling of botnets and tools of defense against them, *Sistemi visokoi dostupnosti*, (2) (2011), pp.107-111. (in Russian)

[23] I.V. Kotenko, A.M. Konovalov, A.V. Shorov, Researchers modeling of botnets and defense against them, *Prilojenie k jurnalu "Informacionnie technologii*", (1) (2012), pp. 32. (in Russian)

[24] I.V. Kotenko, A.V. Shorov, F.G. Nesteruk, Analysis of bio-inspired approaches for defense of computer systems and networks, *Trudy SPIIRAN,* (3) 18 (2011), pp. 19-73. (in Russian)

[25] I. Kotenko, A. Konovalov, A. Shorov, Agent-based Modeling and Simulation of Botnets and Botnet Defense, *Conference on Cyber Conflict. Proceedings 2010.* CCD COE Publications. Tallinn, Estonia, (June 15-18, 2010), pp. 21-44.

[26] I. Kotenko, A. Konovalov, A. Shorov, Agent-based simulation of cooperative defense against botnets, *Concurrency and Computation: Practice and Experience,* (24) 6 (2012), pp. 573-588.

[27] F.G. Nesteruk, A.V. Suhanov, L.G. Nesteruk, G.F. Nesteruk, *Adaptive Means of Information Systems Safety Supplying,* Monograph. SPb.: Polytechnic University Publishing, 2008, 626 p. (in Russian)

[28] J.B. Dennis, J.B. Fossin, J.P. Linderman, *Scheme of data flow*, Teoriya programmirivaniya, Novosibirsk: VC SO AN SSSR, (1972), Part. 2. pp. 7-43. (in Russian)

[29] G.F. Nesteruk, M.S. Kupriyanov, F.G. Nesteruk, About developing of language means for neural networks structure programming, *Proceedings of V International Conference SCM'2002.* SPb, (2002), Vol. 2. pp. 52-55. (in Russian)

[30] F.G. Nesteruk, L.G. Nesteruk, G.F. Nesteruk, Application of the Formal Model for Describing Processes of Adaptive Information Security in Computer-aided Systems, *Automation and Remote Control,* (70) 3 (2009), pp. 491-501.

[31] A.A. Chechulin, I.V. Kotenko, Combining of defense tools against scanning in computer networks, *Informacionno-upravliauschie sistemi,* (12) *(*2010), pp. 21-27. (in Russian)

Игорь Витальевич Котенко, *Заведующий лабораторией проблем компьютерной безопасности СПИИРАН. Закончил с отличием ВИКИ им. А.Ф. Можайского (1983 г.) и Военную академию связи (1987 г.). В 1990 г. защитил кандидатскую диссертацию, а в 1999 г. – докторскую. В 2001 г. присвоено ученое звание профессор по кафедре "Телекоммуникационные системы". Автор более 450 научных работ. Область научных интересов – информационная безопасность, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, ложные информационные системы, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму; искусственный интеллект, в том числе многоагентные системы, мягкие и эволюционные вычисления, машинное обучение.*

Филипп Геннадьевич Нестерук, *Старший научный сотрудник лаборатории проблем компьютерной безопасности СПИИРАН. Закончил ОМГТУ (2000 г.) Поступил в аспирантуру СПбГУЭиФ, в 2005 г. защитил кандидатскую диссертацию по теме «Разработка модели адаптивной системы защиты информации на базе нейро-нечетких сетей», специальность 05.13.19, в СПбГУ ИТМО.*

Андрей Владимирович Шоров, *Научный сотрудник лаборатории проблем компьютерной безопасности СПИИРАН. Окончил Санкт-Петербургский Государственный Инженерно-Экономический Университет (2008 г.), в 2012 г. защитил кандидатскую диссертацию по теме «Имитационное моделирование механизмов защиты компьютерных сетей от инфраструктурных атак на основе подхода “Нервная система сети”» (Специальность: 05.13.19 — Методы и системы защиты информации, информационная безопасность), в СПИИРАН.*

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

CONCEPTION OF A HYBRID ADAPTIVE PROTECTION OF INFORMATION SYSTEMS

Igor V. Kotenko, Filipp G. Nesteruk, Andrey V. Shorov

Laboratory of computer security problems SPIIRAS
39, 14th Liniya, St. Petersburg, 199178, Russia
ivkote@iias.spb.su, 08p@mail.ru, ashorov@comsec.spb.ru
http://comsec.spb.ru/en

Abstract: *The paper suggests the conception of a hybrid adaptive protection of information and telecommunication systems which is based on a biometaphor of nervous and neural networks. A top level of a protection system, based on an approach of "nervous system network" is a distributed mechanism for collecting and processing information. We suggest to implement the information processes on the low level with the assistance of an "information field" programming. It allows specifying the distributed information fields in the form of neural network software packages.*

Keywords: *Data mining, malware, detection.*

1. INTRODUCTION

There is an analogy in evolution and natural selection in nature and technical systems. Living organisms exist and evolve through improved protection against a variety of threats by using information circulating in their distributed structure and implementing various security mechanisms.

Therefore, it seems that it is necessary to endow the information security systems by evolutionary properties inherent biological systems. These properties are the possibility of progress (self-improvement), adaptability (accommodation to the current conditions of the situation), reproduction and inheritance.

This thesis is confirmed by the latest trends in the industry of software systems. For example, known software vendors claim that they need active adaptive security technologies, based on an assessment of the behavior of software components in terms of their potential danger.

In the paper we propose the conception of adaptive protection of information systems based on hybrid mechanisms that combine bio-metaphors of nervous and neural networks.

First we outline a brief analysis of investigations based on the bio-metaphors and different hybrid approaches. Then we define an approach "network nervous system" as the top-level protection system, determine the lower protection level as neuro-fuzzy system, and present an example of their realization.

2. TOP LEVEL PROTECTION MECHANISMS

One of approaches for protection of computer networks is a bio-inspired approach "nervous network system" [1-3].

The protection system is based on a distributed mechanism for collecting and processing information, which coordinates the activities of the main devices of the network, detected of attack and take countermeasures.

The structure of the "nervous network" follows the structure of the human nervous system (Fig.1). The mechanism of the "nervous network" is distributed, i.e. no single center which coordinates the activities of all network.

The protection system consists of two main components – the server of the "nervous network system" and the node of the "nervous network system". Servers are installed in different subnets and implement most functions of information processing and analysis, as well as the coordination of nearby network devices.

Nodes are used for data collection, initial processing and transmission of network status information to servers. Nodes can be installed on routers. Servers are located in different subnets and exchange information on the status of their subnets.

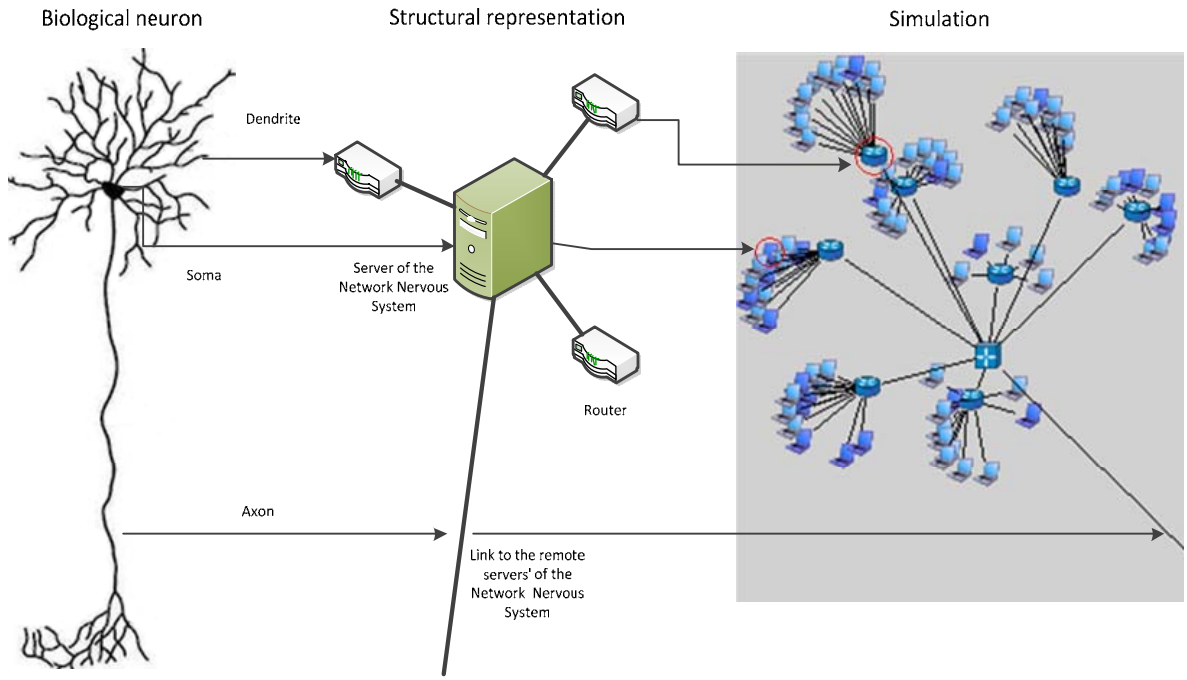


Fig.1. – Metaphor of the biological neuron in the computer network model

Thus, based on the metaphor of the "nervous network system", the paper proposes an adaptive network infrastructure which provides information collection and its transfer to the special server and making decisions based on the current situation.

3. LOWER LEVEL PROTECTION MECHANISMS

We suggest implementing the information processes on the lower level with the assistance of an "information field" programming [4, 5].

It allows specifying the distributed information fields in the form of neural network software packages.

Adaptive processes in the information fields allow developing the security systems which can evolve and gain experience when expanding the set of threats. In this case the inheritance of the experience is reduced to transferring of information fields.

4. HYBRID APPROACH FOR PROTECTION MECHANISMS

It is assumed that the conceptual and architectural solutions for building adaptive hybrid information protection systems should be based on the principles of bio-analogy. The proposed concept of a hybrid adaptive protection of information systems is formed by combining two above approaches.

This research is being supported by the state contract 11.519.11.4008 of Ministry of education and science of Russia, grant of Russian Foundation of Basic Research (Project No. №13-01-00843-a), program of fundamental research of the Department for Informational Technologies and Computation Systems of the Russian Academy of Sciences (contract No 2.2), Russian Science Support Foundation and partly funded by the EU as part of the MASSIF project and SecFutur project.

5. REFERENCES

[1] Y. Chen, H. Chen, NeuroNet: An Adaptive Infrastructure for Network Security, *International Journal of Information, Intelligence and Knowledge,* (1) 2 (2009), pp. 143-168.

[2] I. Kotenko, A. Konovalov, A. Shorov Agent-based Modeling and Simulation of Botnets and Botnet Defense, *Conference on Cyber Conflict. Proceedings 2010.* CCD COE Publications. Tallinn, Estonia, (June 15-18, 2010), pp. 21-44.

[3] I. Kotenko, A. Konovalov, A. Shorov Agent-based simulation of cooperative defence against botnets, *Concurrency and Computation: Practice and Experience,* Vol. 24, Issue 6, (25 April 2012), pp. 573-588.

[4] F.G. Nesteruk, A.V. Suhanov, L.G. Nesteruk, G.F. Nesteruk, *Adaptive Means of Information Systems Safety Supplying,* Monograph. SPb.: Polytechnic University Publishers, 2008, 626 p. (in Russian)

[5] F.G. Nesteruk, L.G. Nesteruk, G.F. Nesteruk, Application of the Formal Model for Describing Processes of Adaptive Information Security in Computer-aided Systems, *Automation and Remote Control,* (70) 3 (2009), pp. 491-501.

Naksit Anantalapochai, Axel Sikora
INTEGRATION OF BACNET OPC UA-DEVICES USING A JAVA OPC UA SDK SERVER WITH BACNET OPEN SOURCE LIBRARY IMPLEMENTATION

The variety of technologies used in modern Building Automation Systems (BAS) calls for methods to support interoperability of the devices from different technologies and vendors. OLE for Process Control Unified Architecture (OPC UA) provides the possibility to enable secure interoperability of devices with platform independence and efficient information model features. However, OPC has not found broad space in the world of building automation, yet.

In this paper, results and experiences from a project are presented, where BACnet devices were implemented with OPC UA standard models. The values and controls are presented by the OPC UA server running on an embedded device. In the method, we map the BACnet information models into the corresponding OPC UA information models. The information model (in OPC UA form) of the BACnet devices can be accessed by connecting the OPC UA Clients to the OPC UA Server. This objective should be pursued by using as many available open-source projects as possible.

Andrew J. Kornecki, Slawomit T. Wierzchon, Janusz Zalewski
REASONING UNDER UNCERTAINTY WITH BAYESIAN BELIEF NETWORKS ENHANCED WITH ROUGH SETS

The objective of this paper is to present a new approach to reasoning under uncertainty, based on the use of Bayesian belief networks (BBN's) enhanced with rough sets. The role of rough sets is to provide additional reasoning to assist a BBN in the inference process, in cases of missing data or difficulties with assessing the values of related probabilities. The basic concepts of both theories, BBN's and rough sets, are briefly introduced, with examples showing how they have been traditionally used to reason under uncertainty. Two case studies from the authors' own research are discussed: one based on the evaluation of software tool quality for use in real-time safety-critical applications, and another based on assisting the decision maker in taking the right course of action, in real time, in the naval military exercise. The use of corresponding public domain software packages based on BBN's and rough sets is outlined, and their application for real-time reasoning in processes under uncertainty is presented.

Vitaly Deibuk, Ion Grytsku
OPTIMAL SYNTHESIS OF REVERSIBLE QUANTUM SUMMATORS USING GENETIC ALGORITHM

The paper suggests a new way of chromosome coding in a genetic algorithm for simulation of reversible one-bit full summators with propagate function in Fredkin basis. The circuits obtained with the use of such an approach demonstrate better delay parameters and better number of inputs/outputs compared with the known analogs. It confirms the effectiveness and applicability of the proposed approach.

Vladimir G. Red'ko
INTERACTION BETWEEN LEARNING AND EVOLUTION IN POPULATIONS OF AUTONOMOUS AGENTS

The model of interaction between learning and evolution for the evolving population of modeled organisms is designed and investigated. The mechanism of genetic assimilation of the acquired features during the numerous generations of Darwinian evolution is studied. The mechanism of influence of the learning load is analyzed. It is showed that the learning load leads to a significant acceleration of an evolution. The hiding effect is also studied. This effect means that a strong learning inhibits the evolutionary search in some situations.

Victor Chernega
PERFORMANCE OF A TRANSPORT LEVEL OF WLANS IEEE 802.11g FUNCTIONING IN INFRASTRUCTURAL MODE

The procedure of frames exchange between client computers and a wireless access point based on 802.11g standard is analyzed in details. We have obtained the expressions which allow calculate the potential bandwidth of such wireless network.

Vladimir Haasz, David Slepicka, Petr Suchanek
POST-CORRECTION OF ADC NON-LINEARITY USING INTEGRAL NON-LINEATITY CURVE

The accuracy of AD conversion can be improved using the post-correction of digitizer non-linearity. In principle two methods could be applied – look-up table or an analytical inverse function of integral non-linearity curve (INL(n)). Look-up table can be easily implemented but it demands huge memory space particularly for high resolution ADCs. Inverse function offers flexible solution for parameterization (e.g. frequency dependence) but it also requires fast DSP for real-time correction. The data or coefficients for both methods are frequently determined from a histogram of acquired pure sinusoidal signal. Non-linearity curve can also be gained by another procedure demanding significantly less samples – approximation from a frequency spectrum. The correction of ADC nonlinearity by means of inverse function of INL(n) curve is analyzed in this paper and the results are presented.

Dmitry V. Komashinskiy, Igor V. Kotenko
INTELLIGENT DATA ANALYSIS FOR MALWARE DETECTION

The paper considers a state-of-the-art survey of systems for malware detection and identification based on intelligent data analysis. The SADT methodology was adopted to formalize this process in order to generalize common procedural aspects described in the analyzed papers within the area. The set of basic abstract items specifying the essence of each concrete approach to detect malware is emphasized.

M. Knemeyer, M. Nsaif, F. Glinka, A. Ploss, S. Gorlatch
TOWARDS DATA PERSISTENCY IN REAL-TIME ONLINE INTERACTIVE APPLICATIONS

The class of distributed Real-time Online Interactive Applications (ROIA) includes such important applications as Massively Multiplayer Online Games (MMOGs), as well as interactive e-Learning and simulation systems. These applications usually work in a persistent environment (also called world) which continues to exist and evolve also while the user is offline and away from the application. The challenge is how to efficiently make the world and the player characters persistent in the system over time. In this paper, we deal with storing persistent data of real-time interactive applications in modern relational databases. We analyze the major requirements to a system for persistency and we describe a preliminary design of the Entity Persistence Module (EPM) middleware which liberates the application developer from writing and maintaining complex and error-prone code for persistent data management. EPM automatically performs the mapping operations to store/retrieve the complex data to/from different types of relational databases, supports the management of persistent data in memory, and integrates it into the main loop of the ROIA client-server architecture.

Igor V. Kotenko, Philipp G. Nesteruk, Andrey V. Shorov
CONCEPTION OF A HYBRID ADAPTIVE PROTECTION OF INFORMATION SYSTEMS

The paper suggests the conception of a hybrid adaptive protection of information and telecommunication systems which is based on a biometaphor of nervous and neural networks. A top level of a protection system, based on an approach of “nervous system network” is a distributed mechanism for collecting and processing information. We suggest to implementthe information processes on the low level with the assistance of an “information field” programming. It allows specifying the distributed information fields in the form of neural network software packages.

Naksit Anantalapochai, Axel Sikora
ІНТЕГРАЦІЯ ПРИСТРОЇВ BACNET OPC UA ЗА ДОПОМОГОЮ СЕРВЕРА JAVA OPC UA SDK З РЕАЛІЗАЦІЄЮ ВІДКРИТОЇ БІБЛІОТЕКИ BACNET

Різноманітність технологій, що використовуються в сучасних системах автоматизованої побудови (САП) вимагає методів для підтримки сумісності пристроїв різних технологій і постачальників. OLE для управління процесами уніфікованої архітектури (OPC UA) надає можливості для забезпечення безпечної взаємодії з пристроями незалежно від платформи і особливостей ефективної інформаційної моделі. Тим не менш, OPC поки що не знайшла широкого застосування в побудові автоматизованих систем.

У даній статті, представлені результати та досвід проекту, де пристрої BACnet були реалізовані із стандартними моделями OPC UA. Змінні та управління представленого сервера OPC UA працюють на вбудованому пристрої. У цьому способі ми відобразили інформаційні моделі BACnet у відповідних інформаційних моделях OPC UA. Інформаційну модель (у формі OPC UA) з пристрою BACnet можна отримати, підключивши клієнтів OPC UA до сервера OPC UA. Це завдання має вирішуватися шляхом використання якомога більшої кількості проектів з відкритим кодом.

Andrew J. Kornecki, Slawomit T. Wierzchon, Janusz Zalewski
АРГУМЕНТУВАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ З БАЙЄСІВСЬКИМИ ДОВІРЧИМИ МЕРЕЖАМИ, ЩО ПОКРАЩЕНІ НАБЛИЖЕНИМИ МНОЖИНАМИ

Метою даної роботи є представлення нового підходу до аргументування в умовах невизначеності, що базується на використанні Байєсівської довірчої мережі (БДМ) покращеної наближеними множинами. Роль наближеної множини полягає в наданні додаткових аргументів, щоб допомогти БДМ в процесі генерації висновку, у разі відсутності даних або при труднощах з оцінкою значень відповідних ймовірностей. Коротко представлені основні поняття обох теорій, БДМ та наближених множин, з прикладами, що показують, як вони традиційно використовувались для аргументування в умовах невизначеності. Розглядаються два тематичних випадки на основі власних досліджень авторів: один базується на оцінці якості програмного продукту, що використовується в режимі реального часу для безпеко-критичних додатків, а інший, для надання допомоги особі, що приймає рішення в реальному часі про правильний курс дій у військово-морських навчаннях. Описано використання відповідних пакетів програмного забезпечення на основі БДМ і наближених множин, а також представлено їх застосування в реальному часі для аргументування в умовах невизначеності.

Віталій Дейбук, Іон Грицку
ОПТИМАЛЬНИЙ СИНТЕЗ ЗВОРОТНИХ КВАНТОВИХ СУМАТОРІВ З ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

У роботі запропоновано новий спосіб кодування хромосом у генетичному алгоритмі для моделювання схем зворотних повних однорозрядних суматорів з функцією транзиту у базисі елементів Фредкіна. Отримані з допомогою такого підходу схеми мають кращі параметри затримки та кількості зайвих виходів(входів) порівняно з відомими аналогами, що демонструє ефективність та застосовність такого підходу.

Володимир Г. Редько
ВЗАЄМОДІЯ МІЖ НАВЧАННЯМ І РОЗВИТКОМ В ПОПУЛЯЦІЯХ АВТОНОМНИХ АГЕНТІВ

Розроблено та досліджено модель взаємодії між навчанням та еволюцією для виділення популяції модельованих організмів. Вивчено механізм генетичної асиміляції набутих рис під час численних поколінь дарвінівської еволюції. Проаналізовано механізм впливу навчального навантаження. Показано, що навчальне навантаження призводить до значного прискорення еволюції. Також вивчено прихований ефект; цей ефект означає, що посилене навчання стримує еволюційний пошук в деяких ситуаціях.

Віктор Чернега
ПРОПУСКНА СПРОМОЖНІСТЬ ТРАНСПОРТНОГО РІВНЯ БЕЗПРОВІДНИХ ЛОКАЛЬНИХ МЕРЕЖ IEEE 802.11g, ЩО ФУНКЦІОНУЮТЬ В ІНФРАСТРУКТУРНОМУ РЕЖИМІ

Детально проаналізована процедура обміну кадрами між клієнтськими комп'ютерами і точкою доступу безпровідної локальної комп'ютерної мережі стандарту 802.11g і отримані вирази, що дозволяють розрахувати потенційну пропускну спроможність такої мережі.

Vladimir Haasz, David Slepicka, Petr Suchanek
ПОСТ-КОРЕКЦІЯ НЕЛІНІЙНОСТІ АЦП ЗА ДОПОМОГОЮ ІНТЕГРАЛЬНОЇ НЕЛІНІЙНОЇ КРИВОЇ

Точність аналого-цифрового перетворення може бути покращена за допомогою пост-корекції нелінійності цифрового перетворювача. В принципі можна застосовувати два методи – довідникову таблицю або аналітичну зворотну функцію інтегральної нелінійної кривої (INL(n)). Довідкова таблиця може бути легко реалізована, але це вимагає величезного обсягу пам'яті, особливо при великій роздільній здатності АЦП. Зворотна функція пропонує гнучке рішення для параметризації (наприклад, частотна залежність), але це також вимагає швидкого процесора цифрової обробки сигналів для корекції в реальному часі. Дані або коефіцієнти для обох методів часто визначаються за гістограмою отриманого чистого синусоїдального сигналу. Нелінійну криву можна також отримати в рамках іншої процедури, що вимагає значно менше зразків – шляхом апроксимації частотного спектру. В даній статті аналізується корекція нелінійності АЦП за допомогою зворотної функції кривої INL(n). Представлено відповідні результати досліджень.

Дмитро Комашинський, Ігор Котенко
ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ ДЛЯ ВИЯВЛЕННЯ ШКІДЛИВИХ ПРОГРАМ

У статті проводиться огляд найбільш значущих робіт у галузі створення систем виявлення та ідентифікації шкідливих програм на основі методів інтелектуального аналізу даних. Для формалізації цього процесу використовуються елементи методології SADT, що узагальнюють основні процедурні аспекти існуючих робіт, присвячених даній предметній області. Виділяються основні групи сутностей, що використовуються для формування типових методик виявлення шкідливих програм на основі даної групи методів.

M. Knemeyer, M. Nsaif, F. Glinka, A. Ploss, S. Gorlatch
ДО СТІЙКОСТІ ДАНИХ В ОН-ЛАЙН ІНТЕРАКТИВНИХ ДОДАТКАХ РЕАЛЬНОГО ЧАСУ

Клас розподілених он-лайн інтерактивних додатків реального часу включає в себе такі важливі додатки, як он-лайн ігри великого масиву гравців, а також інтерактивні системи електронного навчання та моделювання. Ці програми зазвичай працюють у постійному оточенні (що зветься світом), який продовжує існувати і розвиватися також, коли користувач не в мережі і не використовує додаток. Завдання полягає в тому, як ефективно змусити світ і персонажів залишатися в системі з плином часу. У даній роботі ми маємо справу зі зберіганням постійних даних в інтерактивних додатках он-лайн в сучасних реляційних базах даних. Ми аналізуємо основні вимоги до системи стійкості та описуємо ескізний проект проміжного модуля тривалої стійкості (МТС), який звільняє розробника додатків від написання і підтримки складного коду з можливими помилками для постійного управління даними. МТС автоматично виконує операції зіставлення, для збереження/відновлення комплексу даних в/з різних типів реляційних баз даних, підтримує управління постійними даними в пам'яті, та інтегрує їх в основний цикл клієнт-серверної архітектури.

Ігор Котенко, Пилип Нестерук, Андрій Шоров
КОНЦЕПЦІЯ ГІБРИДНОГО АДАПТИВНОГО ЗАХИСТУ ІНФОРМАЦІЙНИХ СИСТЕМ

У роботі пропонується концепція гібридного адаптивного захисту інформаційно-телекомунікаційних систем на основі біометафори нервових і нейронних мереж. Верхній рівень системи захисту, що оснований на підході «нервова система мережі», базується на розподіленому механізмі збору та обробки інформації, який координує дії основних пристроїв комп'ютерної мережі, ідентифікує атаки і приймає контрзаходи. Реалізацію інформаційних процесів на нижньому рівні пропонується виконувати із залученням «інформаційно-польового» програмування, що дає змогу описувати розподілені інформаційні поля у вигляді пакетних нейромережевих програм.

Naksit Anantalapochai, Axel Sikora
ИНТЕГРАЦИЯ УСТРОЙСТВ BACNET OPC UA С ПОМОЩЬЮ СЕРВЕРА JAVA OPC UA SDK С РЕАЛИЗАЦИЕЙ ОТКРЫТОЙ БИБЛИОТЕКИ BACNET

Разнообразие технологий, используемых в современных системах автоматизированного построения (САП) требует методов для поддержки совместимости устройств различных технологий и поставщиков. OLE для управления процессами унифицированной архитектуры (OPC UA) предоставляет возможности для обеспечения безопасного взаимодействия с устройствами независимо от платформы и особенностей эффективной информационной модели. Тем не менее, OPC пока не нашла широкого применения в построении автоматизированных систем.

В данной статье, представлены результаты и опыт проекта, где устройства BACnet были реализованы со стандартными моделями OPC UA. Переменные и управление представленного сервера OPC UA работают на встроенном устройстве. В этом способе мы отразили информационные модели BACnet в соответствующих информационных моделях OPC UA. Информационную модель (в форме OPC UA) из устройства BACnet можно получить, подключив клиентов OPC UA к серверу OPC UA. Эта задача должна решаться путем использования как можно большего количества проектов с открытым кодом.

Andrew J. Kornecki, Slawomit T. Wierzchon, Janusz Zalewski
АРГУМЕНТИРОВАНИЕ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ С БАЙЕСОВСКИМИ ДОВЕРИТЕЛЬНЫМИ СЕТЯМИ, УЛУЧШЕННЫМИ ПРИБЛИЖЕННЫМИ МНОЖЕСТВАМИ

Целью данной работы является представление нового подхода к аргументированию в условиях неопределенности, основанному на использовании Байесовской доверительной сети (БДС) улучшенной приближенными множествами. Роль приближенного множества заключается в предоставлении дополнительных аргументов, чтобы помочь БДС в процессе генерации вывода, в случае отсутствия данных или при трудностях с оценкой значений соответствующих вероятностей. Кратко представлены основные понятия обеих теорий, БДС и приближенных множеств, с примерами, показывающими, как они традиционно использовались для аргументирования в условиях неопределенности. Рассматриваются два тематических случая на основе собственных исследований авторов: первый базируется на оценке качества программного продукта, используемого в режиме реального времени для безопасно-критических приложений, а другой, для оказания помощи лицу, принимающему решения в реальном времени о правильном курсе действий в военно-морских учениях. Описано использование соответствующих пакетов программного обеспечения на основе БДС и приближенных множеств, а также представлено их применения в реальном времени для аргументации в условиях неопределенности.

Виталий Дейбук, Ион Грицку
ОПТИМАЛЬНЫЙ СИНТЕЗ ОБРАТНЫХ КВАНТОВЫХ СУММАТОРОВ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

В работе предложен новый способ кодирования хромосом в генетическом алгоритме для моделирования схем обратных полных одноразрядных сумматоров с функцией транзита в базисе элементов Фредкина. Полученные с помощью такого подхода схемы имеют лучшие параметры задержки и количества лишних входов (выходов) сравнительно с известными аналогами, что демонстрирует эффективность и применимость такого подхода.

Владимир Г. Редько
ВЗАИМОДЕЙСТВИЕ МЕЖДУ ОБУЧЕНИЕМ И РАЗВИТИЕМ В ПОПУЛЯЦИЯХ АВТОНОМНЫХ АГЕНТОВ

Разработана и исследована модель взаимодействия между обучением и эволюцией для выделения популяции моделируемых организмов. Изучен механизм генетической ассимиляции приобретенных черт во время многочисленных поколений дарвиновской эволюции. Проанализирован механизм влияния учебной нагрузки. Показано, что учебная нагрузка приводит к значительному ускорению эволюции. Также изучен скрытый эффект; этот эффект означает, что усиленное обучение сдерживает эволюционный поиск в некоторых ситуациях.

Виктор Чернега
ПРОПУСКНАЯ СПОСОБНОСТЬ ТРАНСПОРТНОГО УРОВНЯ БЕСПРОВОДНЫХ ЛОКАЛЬНЫХ СЕТЕЙ IEEE 802.11g, ФУНКЦИОНИРУЮЩИХ В ИНФРАСТРУКТУРНОМ РЕЖИМЕ

Детально проанализирована процедура обмена кадрами между клиентскими компьютерами и точкой доступа беспроводной локальной компьютерной сети стандарта 802.11g и получены выражения, позволяющие рассчитывать потенциальную пропускную способность такой сети.

Vladimir Haasz, David Slepicka, Petr Suchanek
ПОСТ-КОРРЕКЦИЯ НЕЛИНЕЙНОСТИ АЦП С ПОМОЩЬЮ ИНТЕГРАЛЬНОЙ НЕЛИНЕЙНОЙ КРИВОЙ

Точность аналого-цифрового преобразования может быть улучшена с помощью пост-коррекции нелинейности цифрового преобразователя. В принципе можно использовать два метода – справочную таблицу или аналитическую обратную функцию интегральной нелинейной кривой (INL(n)). Справочная таблица может быть легко реализована, но это требует огромного объема памяти, особенно при большом разрешении АЦП. Обратная функция предлагает гибкое решение для параметризации (например, частотная зависимость), но это также требует быстрого процессора цифровой обработки сигналов для коррекции в реальном времени. Данные или коэффициенты для обоих методов часто определяются по гистограмме полученного чистого синусоидального сигнала. Нелинейную кривую также можно получить в рамках другой процедуры, что требует значительно меньше образцов – путем аппроксимации частотного спектра. В данной статье анализируется коррекция нелинейности АЦП с помощью обратной функции кривой INL(n). Представлены соответствующие результаты исследований.

Дмитрий Комашинский, Игорь Котенко
ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ ДЛЯ ВЫЯВЛЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

В статье проводится обзор наиболее значимых работ в области создания систем обнаружения и идентификации вредоносных программ на основе методов интеллектуального анализа данных. Для формализации этого процесса используются элементы методологии SADT, обобщающие основные процедурные аспекты существующих работ, посвященных данной предметной области. Выделяются основные группы сущностей, используемых для формирования типовых методик обнаружения вредоносных программ на основе данной группы методов.

M. Knemeyer, M. Nsaif, F. Glinka, A. Ploss, S. Gorlatch
К УСТОЙЧИВОСТИ ДАННЫХ В ОН-ЛАЙН ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЯХ РЕАЛЬНОГО ВРЕМЕНИ

Класс распределенных он-лайн интерактивных приложений реального времени включает в себя такие важные приложения, как он-лайн игры большого массива игроков, а также интерактивные системы электронного обучения и моделирования. Эти программы обычно работают в постоянном окружении (называемом миром), который продолжает существовать и развиваться также, когда пользователь не в сети и не использует приложение. Задача состоит в том, как эффективно заставить мир и персонажей оставаться в системе с течением времени. В данной работе мы имеем дело с хранением постоянных данных в интерактивных он-лайн приложениях в современных реляционных базах данных. Мы анализируем основные требования к системе устойчивости и описываем эскизный проект промежуточного модуля длительной устойчивости (МДУ), который освобождает разработчика приложений от написания и поддержки сложного кода с возможными ошибками для постоянного управления данными. МТС автоматически выполняет операции сравнения, для сохранения/восстановления комплекса данных в/из различных типов реляционных баз данных, поддерживает управление постоянными данными в памяти, и интегрирует их в основной цикл клиент-серверной архитектуры.

Игорь Котенко, Филипп Нестерук, Андрей Шоров
КОНЦЕПЦИЯ ГИБРИДНОЙ АДАПТИВНОЙ ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ

В работе предлагается концепция гибридной адаптивной защиты информационно-телекоммуникационных систем на основе биометафоры нервных и нейронных сетей. Верхний уровень системы защиты, основанный на подходе «нервная система сети», базируется на распределенном механизме сбора и обработки информации, который координирует действия основных устройств компьютерной сети, идентифицирует атаки и принимает контрмеры. Реализацию информационных процессов на нижнем уровне предлагается выполнять с привлечением «информационно-полевого» программирования, которое позволяет описывать распределенные информационные поля в виде пакетных нейросетевых программ.

Prepare your paper according to the following requirements:

Unconditional requirement - paper should not be published earlier.

(i) Suggested composition (frame) of paper:
- Issue formulation stressing its urgent solving; evaluation of recent publications in the explored issue
- short formulation of paper's purpose
- description of proposed method (algorithm)
- implementation and testing (verification)
- conclusion.

(ii) Use A4 (210 x 297 mm) paper. Size of paper has to be extended up to 6-8 pages.

(iii) Please use main text two column formatting;

(iv) A paper must have an abstract and some keywords;

(v) Place a full list of references at the end of the paper. Please place the references according to their order of appearance in the text.

(vi) An affiliation of each author is wanted.

(vii) The text should be single-spaced. Use Times New Roman (11 points, regular) typeface throughout the paper.

(viii) Equations should be placed in separate lines and numbered. The numbers should be within brackets and right aligned.

(ix) The figures and tables must be numbered, have a self-contained caption. Figure captions should be below the figures; table captions should be above the tables. Also, avoid placing figures and tables before their first mention in the text.

(x) As soon as you have the complete materials, the final versions should come electronically in MS Word'97 of MS Word 2000 format to the address computing@computingonline.net.

(xi) A hardcopy of your article is needed to be sent by regular mail for our publishing house.

(xii) Please send short CVs (up to 20 lines) and photos of every author.

(xiii) There is no other formatting required. The publishing department makes all rest formatting according to the publisher's rules.

Journal Topics:
- Algorithms and Data Structure
- Bio-Informatics
- Cluster and Parallel Computing, Software Tools and Environments
- Computational Intelligence
- Computer Modeling and Simulation
- Cyber and Homeland Security
- Data Communications and Networking
- Data Mining, Knowledge Bases and Ontology
- Digital Signal Processing
- Distributed Systems and Remote Control
- Education in Computing
- Embedded Systems
- High Performance Computing and GRIDS
- Image Processing and Pattern Recognition
- Intelligent Robotics Systems
- Internet of Things
- Standardization of Computer Systems
- Wireless Systems

Основні вимоги до подання і оформлення публікацій наукового журналу "Комп'ютинг":

Безумовною вимогою є те, щоб стаття не була опублікована раніше!

(i) Наукові статті повинні мати такі необхідні елементи:
- постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями;
- аналіз останніх досліджень і публікацій, в яких започатковано розв'язання даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, котрим присвячується означена стаття;
- формулювання цілей статті (постановка завдання);
- виклад основного матеріалу дослідження з повним обгрунтуванням отриманих наукових результатів;
- висновки з даного дослідження і перспективи подальших розвідок у даному напрямку.

(ii) Використовуйте A4 (210 x 297 mm) формат сторінки. Загальний розмір статті має містити 6-8 сторінок.

(iii) Використовуйте двоколонкове форматування основного тексту;

(iv) Стаття повинна обов'язково містити основний текст українською мовою, анотацію (написану на Англійській і Українській мовах) і список ключових слів;

(v) В кінці статті розмістіть список літератури. Розміщуйте список літератури в порядку її цитування.

(vi) Необхідною є інформація про наукові звання, титули та посади авторів.

(vii) Текст повинен бути набраним одинарним інтервалом із використанням шрифту Times New Roman (11 points, regular).

(viii) Формули повинні відділятись від основного тексту пустими стрічками а також пронумеровані у круглих дужках та відцентровані по правому краю.

(ix) Таблиці і рисунки повинні бути пронумерованими. Заголовки рисунків розміщують під рисунком по центру. Заголовки таблиць розміщують по центру зверху таблиці.

(x) Завершені версії статей повинні бути надісланими в електронному MS Word'97 або MS Word 2000 форматі за адресою computing@computingonline.net.

(xi) Просимо надсилати поштою роздруковані копії статей.

(xii) В кінці кожної статті потрібно подати її назву, резюме (абстракт) і ключові слова англійською мовою.

(xiii) Просимо надсилати нам короткі біографічні дані (до 20 рядків) і скановані фотографії кожного із авторів.

(xiv) Видавництво здійснює остаточне форматування тексту згідно із вимогами друку.

(xv) У закордонних читачів можуть виникнути проблеми при ознайомленні з працями на російській та українській мовах. В зв'язку з цим редакційна колегія просить авторів додатково прислати розширений реферат (резюме), щоб б містило дві сторінки тексту англійською мовою, і супроводжувалось заголовком, прізвищами та адресами авторів. Авторам рекомендується використовувати у рисунках статті позначення переважно англійською мовою, або давати переклад у дужках. Тоді у розширеному резюме можна буде посилатися на рисунки у основному тексті.

Тематика журналу:

- Алгоритми та структури даних
- Біо-інформатика
- Кластерні та паралельні обчислення, програмні засоби та середовище
- Обчислювальний інтелект
- Комп'ютерне та імітаційне моделювання
- Кібернетична безпека та захист від тероризму
- Обмін даними та організація мереж
- Видобування даних, бази знань та онтології
- Цифрова обробка сигналів
- Розподілені системи та дистанційне управління
- Освіта в комп'ютингу
- Вбудовувані системи
- Високопродуктивні обчислення та ГРІД
- Обробка зображень та розпізнавання шаблонів
- Інтелектуальні робототехнічні системи
- Інтернет речей
- Стандартизація комп'ютерних систем
- Безпровідні системи

Основные требования к подаче и оформлению публикаций научного журнала “Компьютинг”:

Безусловное требование – чтобы статья не была опубликована ранее!

(i) Научные статьи должны иметь такие необходимые элементы:
- постановка проблемы в общем виде и ее связь с важными научными или практическими задачами;
- анализ последних исследований и публикаций, в которых начаты решения данной проблемы и на которые опирается автор, выделение нерешенных прежде частей общей проблемы, которым посвящается обозначенная статья;
- формулирование целей статьи (постановка задача);
- изложение основного материала исследования с полным обоснованием полученных научных результатов;
- выводы из данного исследования и перспективы дальнейших изысканий в данном направлении.

(ii) Используйте A4 (210 x 297 mm) формат страницы. Общий размер статьи 6-8 страниц.

(iii) Используйте двухколоночное форматирование основного текста;

(iv) Статья должна обязательно содержать основной текст на Русском языке, аннотацию (написанную на Английском и Русском языках) и список ключевых слов;

(v) В конце статьи разместите список литературы. Размещайте список литературы в порядке ее цитирования.

(vi) Необходима информация о научных званиях, титулах и должностях авторов.

(vii) Текст должен быть набранным одинарным интервалом с использованием шрифта Times New Roman (11 points, regular).

(viii) Формулы должны отделяться от основного текста пустыми строками, а также пронумерованные в круглых скобках и отцентрованные по правому краю.

(ix) Таблицы и рисунки должны быть пронумерованными. Заголовки рисунков размещают под рисунком по центру. Заголовки таблиц размещают по центру сверху таблицы.

(x) Завершенные версии статей должны быть присланы в электронном MS Word'97 или MS Word 2000 формате по адресу computing@computingonline.net.

(xi) Просим присылать распечатанные копии статей по почте.

(xii) В конце каждой статьи необходимо предоставить ее название, резюме (абстракт) и ключевые слова на английском языке.

(xiii) Просим присылать нам короткие биографические данные (до 20 строчек) и сканированные фотографии каждого из авторов.

(xiv) Издательство осуществляет окончательное форматирование текста в соответствии с требованиями печати.

(xv) У зарубежных читателей могут возникнуть проблемы при ознакомлении с трудами на русском и украинском языках. В связи с этим редакционная коллегия просит авторов дополнительно прислать расширенный реферат (резюме), который содержал бы две страницы текста на английском языке, и сопровождался заголовком, фамилиями и адресами авторов. Авторам рекомендуется использовать в рисунках статьи обозначения преимущественно на английском языке, или давать перевод в скобках. Тогда в расширенном резюме можно будет посылаться на рисунки в основном тексте.

Тематика журнала:

- Алгоритмы и структуры данных
- Био-информатика
- Кластерные и параллельные вычисления, программные средства и среды
- Вычислительный интеллект
- Компьютерное и имитационное моделирование
- Кибернетическая безопасность и защита от терроризма
- Обмен данными и организация сетей
- Добыча данных, базы знаний и онтологии
- Цифровая обработка сигналов
- Распределенные системы и дистанционное управление
- Образование в компьютинге
- Встраиваемые системы
- Высокопроизводительные вычисления и ГРИД
- Обработка изображений и распознавание шаблонов
- Интеллектуальные робототехнические системы
- Интернет вещей
- Стандартизация компьютерных систем
- Беспроводные системы

CALL FOR PAPERS

IDAACS'2013

The 7th IEEE International Conference on
Intelligent Data Acquisition and Advanced Computing Systems:
Technology and Applications

September 11-14, 2013
BERLIN, GERMANY

Organized by
Research Institute of Intelligent Computer Systems,
Ternopil National Economic University and V.M. Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine
in cooperation with
University of Applied Sciences,
Hochschule für Technik und Wirtschaft (HTW)
Berlin, Germany
www.idaacs.net

Conference Co-Chairmen
Anatoly Sachenko, Ukraine
Jürgen Sieck, Germany
International Programme Committee Co-Chairmen
Vladimir Haasz, Czech Republic
Kurosh Madani, France
IDAACS International Advisory Board
Dominique Dallet, France
Richard Duro, Spain
Domenico Grimaldi, Italy
Vladimir Haasz, Czech Republic
Robert Hiromoto, USA
Theodore Laopoulos, Greece
George Markowsky, USA, Chair
Vladimir Oleshchuk, Norway
Fernando Lopez Pena, Spain
Peter Reusch, Germany
Anatoly Sachenko, Ukraine
Wieslaw Winiecki, Poland

Important Dates
Abstract submission: 1 March 2013
Notification of Acceptance: 15 April 2013
Camera ready paper: 1 June 2013
Early registration: 15 April – 30 June 2013

CORRESPONDENCE
The correspondence should be directed to
IDAACS Organizing Committee:

IDAACS Organizing Committee
Research Institute of Intelligent Computer Systems
Ternopil National Economic University
3 Peremoga Square
Ternopil 46020 Ukraine
Phone: +380352-475050 ext.: 12234
Fax: +380352-475053 (24 hours)
E-mail: orgcom@idaacs.net

Topics

The conference scope includes, but it's not limited to:

1. Special Stream in Wireless Systems
2. Special Stream in Project Management
3. Special Stream in Cyber Security
4. Special Stream in High Performance Computing
5. Special Stream in eLearning Management
6. Special Stream in Advanced Information Technologies in Ecology
7. Advanced Instrumentation and Data Acquisition Systems
8. Intelligent Distributed Systems and Remote Control
9. Virtual Instrumentation Systems
10. Cluster and Parallel Computing, Software Tools and Environments
11. Embedded Systems
12. Artificial Intelligence and Neural Networks for Advanced Data Acquisition and Computing Systems
13. Advanced Mathematical Methods for Data Acquisition and High Performance Computing
14. Pattern Recognition and Digital Image and Signal Processing
15. Data Analysis and Modeling
16. Intelligent Information Systems, Data Mining and Ontology
17. Robotics and Autonomous Systems
18. Information Computing Systems for Education and Commercial Applications
19. Bio-Informatics
20. Safety, Security and Reliability of Software
21. Intelligent Testing and Diagnostics of Computing Systems
22. Internet of Things
23. Special Stream in Intelligent Robotics and Components

It's our pleasure to invite the interested scientists to organize own Streams.